

# Wireless Channel Modeling using MIMObit

Georgios Tsitsikas



School of Electrical and Computer Engineering  
Technical University of Crete

Greece

August 2017



*To my grandfather.*



# Acknowledgements

First of all, I am particularly grateful to Dr. Karystinos for entrusting me with this project and for providing much needed advice during the whole process of our work. I would also like to thank him for always readily providing me with the necessary equipment when required.

I would also like to express my great appreciation to Dr. Buris for his elaborate assistance on comprehending various parts of MIMObit, since it proved of vital importance for the completion of our project.

Furthermore, I wish to thank Dr. Liavas and Dr. Bletsas for being part of the thesis committee, and for their patience and eagerness during my thesis presentation.

Last but not least, I would like to acknowledge the support provided by my lab mates, family and friends while working on this project.

# Abstract

With the ever-increasing need for high speed and high quality wireless communication services that dominates our society nowadays, there has been an inevitable requirement for advanced and innovative antenna systems. The advent of Multiple Input Multiple Output (MIMO) antenna systems has been a game changer in recent years, bringing to life capabilities that were thought to not be possible before. However, as revolutionary as these systems are, they require exponentially more sophisticated and extensive research, which can render the conventional antenna analysis tools ineffective and insufficient.

Nebens' MIMObit offers an integrated platform that aspires to relieve telecommunications engineers of the burden of creating or interlinking suitable tools for their projects, thereby enabling them to focus on what really matters; that is the antenna design itself. Additionally, MIMObit aims to provide features that enhance the current methods and concepts of antenna design, and thus enable the user to study their ideas on a whole new level.

In this thesis, we attempt to break down various parts of the software and explain how they can be related and analyzed in terms of the abstract signals & systems and information theory terminology. Finally, we make some channel capacity comparisons between MIMObit and our own implementations of algorithms in the simple case of Single Input Single Output (SISO) antenna systems, in an attempt to study the extent to which the software adheres to valid theoretical communications models.

# Contents

<b>1</b>	<b>Telecommunication Systems</b>	<b>8</b>
1.1	Wireless Communication . . . . .	8
1.1.1	A Simple Channel Model . . . . .	8
1.1.2	Baseband Equivalent . . . . .	9
1.2	Antennas . . . . .	15
1.2.1	Phasor Representation . . . . .	15
1.2.2	Dipole Antennas . . . . .	17
<b>2</b>	<b>MIMObit</b>	<b>21</b>
2.1	Basics Overview . . . . .	21
2.2	Comprehensive Guide . . . . .	27
2.2.1	Structure Concepts . . . . .	27
2.2.2	Transmitters . . . . .	35
2.2.3	Receivers . . . . .	41
2.2.4	Apps & Simulation Files . . . . .	42
2.2.5	Custom Antennas . . . . .	47
<b>3</b>	<b>Channel Capacity</b>	<b>52</b>
3.1	AWGN Channel . . . . .	52
3.1.1	Baseband . . . . .	52
3.1.2	Passband . . . . .	58
3.2	Flat Channel . . . . .	58
3.3	Frequency-selective Channel . . . . .	60
3.3.1	Water-filling . . . . .	64
3.4	Real-World Channels . . . . .	69
3.4.1	TucStats Software . . . . .	69
3.4.2	Calculations on MIMObit Data . . . . .	74
<b>4</b>	<b>Conclusion And Future Work</b>	<b>91</b>
	<b>Bibliography</b>	<b>92</b>

# Chapter 1

## Telecommunication Systems

### 1.1 Wireless Communication

Every time that electromagnetic waves propagate, we know that depending on the characteristics of the environment, the waves might either be reflected, refracted or diffracted, which causes multipath propagation. In addition, the energy of the waves might be absorbed by physical obstacles or be severely attenuated due to various natural phenomena; in this case, we say that the waves experience shadowing. Due to these perturbations, we know that at every point in space we will observe a variation of the original wave, which is the result of constructive or destructive interference of the multiple reflected, refracted or diffracted versions of the wave.

In wireless communications, we will say that the transmitted signal,  $x(t)$ , experiences *fading*, and that the propagation environment corresponds to a *fading channel*. In practice, the various reflected, refracted or diffracted versions of the transmitted signal can be grouped into a finite number,  $N$ , of groups which have approximately the same properties; that is in general the same amplitude attenuation,  $a$ , and delay,  $\tau$ . Further, in the case that the transmitter operates with alternating currents and voltages, then all these properties will also be dependent on the frequency content of the transmitted signal, and thus the number of the groups along with their properties might change as a function of frequency as well. Finally, it is obvious that due to the vast complexity of most propagation environments in real life, we will not always be able to control its characteristics, and thus it is very likely that its properties will be altered as a function of time. For this reason, common propagation environments are many times approximated using complex probabilistic models, which can then be analyzed employing advanced tools from various areas of mathematics.

#### 1.1.1 A Simple Channel Model

Consider now that we decide to install a receiver at a specific point in the propagation environment in order to capture this signal and establish communication with the transmitter. From the discussion above, we can be confident that the received signal,  $y(t)$ , can be nicely and conveniently approximated as follows

$$y(t, f) = \sum_{i=0}^{N-1} a_i(t, f)x(t - \tau_i(t, f)) \quad (1.1)$$

In order to simplify our model, we are going to focus on the special case where

the transmitter, the receiver and the propagation environment are stationary; hence, our propagation model becomes time-invariant. Second, we further simplify our formulation by assuming that the effects of frequency changes on the signal reception are negligible. As a result, we obtain the following model for the received signal

$$y(t) = \sum_{i=0}^{N-1} a_i x(t - \tau_i) \quad (1.2)$$

Equation (1.2) is really important since it informs us that the whole wireless communication process can in fact be approximated and described fairly well by a Linear Time-Invariant (LTI) channel with an impulse response

$$h(t) = \sum_{i=0}^{N-1} a_i \delta(t - \tau_i) \quad (1.3)$$

We can also calculate its frequency response using the Continuous-Time Fourier Transform as follows

$$\mathcal{F}\{h(t)\} = \mathcal{F}\left\{\sum_{i=0}^{N-1} a_i \delta(t - \tau_i)\right\} \Leftrightarrow \quad (1.4)$$

$$H(F) = \sum_{i=0}^{N-1} a_i \mathcal{F}\{\delta(t - \tau_i)\} \Leftrightarrow \quad (1.5)$$

$$H(F) = \sum_{i=0}^{N-1} a_i e^{-j2\pi F \tau_i} \quad (1.6)$$

### 1.1.2 Baseband Equivalent

When we need to implement a wireless communication system in practice, we usually choose to transmit our signal in a passband  $[F_c - W/2, F_c + W/2]$  around a center frequency  $F_c$ . The transmitter constructs the desired signal for transmission, then upconverts it to a frequency  $F_c$ , it transmits, and finally the signal reaches the receiver, which downconverts the signal to obtain the original signal that the transmitter intended to sent.

Although most of the processing at the transmitter and the receiver happens on the downconverted signal, upconverted signals can prove useful since they can be transmitted at different passbands simultaneously in the same space without one interfering with the other. Another advantage is that as the center frequency of the passband gets higher, then the respective transmission and reception antennas get smaller and can for example be conveniently integrated into portable electronic devices.

As we mentioned, however, most of the processing happens on the downconverted signal, and thus it would be useful if we could transform the wireless propagation formulation in (1.2), and infer an equivalent formulation which is independent of the center frequency.

**Definition 1** (Baseband Equivalent). The baseband equivalent,  $s^b(t)$ , of a real continuous signal  $s(t)$  modulated at frequency  $F_c$  has the following Fourier Transform

$$S^b(F) = \begin{cases} 2S(F + F_c) & F + F_c \geq 0 \\ 0 & F + F_c < 0 \end{cases} \quad (1.7)$$

We will also say that  $S^b(F)$  is the baseband equivalent of  $S(F)$ .

**Corollary 1.1.** To obtain a formula for  $s^b(t)$ , we have the following

$$\begin{aligned} S(F) &= \frac{S^b(F - F_c)}{2} + \frac{S^{b*}(-F - F_c)}{2} \Leftrightarrow \\ \mathcal{F}^{-1}\{S(F)\} &= \mathcal{F}^{-1}\left\{\frac{S^b(F - F_c)}{2}\right\} + \mathcal{F}^{-1}\left\{\frac{S^{b*}(-F - F_c)}{2}\right\} \Leftrightarrow \\ s(t) &= \frac{1}{2} \left[ s^b(t)e^{j2\pi F_c t} + s^{b*}(t)e^{-j2\pi F_c t} \right] \Leftrightarrow \\ s(t) &= \frac{1}{2} \left[ s^b(t)e^{j2\pi F_c t} + (s^b(t)e^{j2\pi F_c t})^* \right] \Leftrightarrow \\ s(t) &= \Re \{ s^b(t) \cdot e^{j2\pi F_c t} \} \end{aligned} \quad (1.8)$$

The factor of 2 in Definition 1 can actually be defined to be any arbitrary number, and it is defined this way in our case for clarity and algebraic reasons in subsection 2.2.1. A different approach would be to define it as  $\sqrt{2}$  so that both the passband signal and its baseband equivalent can have the same energy.

### Continuous-Time

We can now derive the baseband equivalent of our wireless channel in (1.2) as

$$y(t) = \sum_{i=0}^{N-1} a_i x(t - \tau_i) \Leftrightarrow \quad (1.9)$$

$$\Re \{ y^b(t) e^{j2\pi F_c t} \} = \sum_{i=0}^{N-1} a_i \Re \{ x^b(t - \tau_i) e^{j2\pi F_c (t - \tau_i)} \} \Leftrightarrow \quad (1.10)$$

$$\Re \{ y^b(t) e^{j2\pi F_c t} \} = \Re \left\{ \sum_{i=0}^{N-1} a_i x^b(t - \tau_i) e^{j2\pi F_c (t - \tau_i)} \right\} \Leftrightarrow \quad (1.11)$$

$$\Re \{ y^b(t) e^{j2\pi F_c t} \} = \Re \left\{ \sum_{i=0}^{N-1} [a_i e^{-j2\pi F_c \tau_i}] x^b(t - \tau_i) e^{j2\pi F_c t} \right\} \Leftrightarrow \quad (1.12)$$

$$\Re \{ y^b(t) e^{j2\pi F_c t} \} = \Re \left\{ \left( \sum_{i=0}^{N-1} a_i^b x^b(t - \tau_i) \right) e^{j2\pi F_c t} \right\} \Leftrightarrow \quad (1.13)$$

$$y^b(t) = \sum_{i=0}^{N-1} a_i^b x^b(t - \tau_i) \quad (1.14)$$

where

$$a_i^b = a_i e^{-j2\pi F_c \tau_i} \quad (1.15)$$

Similarly to the passband model in (1.2), we observe that the baseband equivalent model of the wireless transmission can also be described by an LTI channel with impulse response

$$h^b(t) = \sum_{i=0}^{N-1} a_i^b \delta(t - \tau_i) \quad (1.16)$$

Also, as in the passband case, the frequency response of this channel is

$$H^b(F) = \sum_{i=0}^{N-1} a_i^b e^{-j2\pi F \tau_i} \quad (1.17)$$

from which we can infer that

$$H^b(F) = \sum_{i=0}^{N-1} a_i e^{-j2\pi F_c \tau_i} e^{-j2\pi F \tau_i} \Leftrightarrow \quad (1.18)$$

$$H^b(F) = \sum_{i=0}^{N-1} a_i e^{-j2\pi(F+F_c)\tau_i} \Leftrightarrow \quad (1.19)$$

$$H^b(F) = H(F + F_c) \quad (1.20)$$

We can see from (1.20) that the baseband equivalent of the channel frequency response is similar to the definitions of the baseband equivalents of the transmitted and the received signals. However, there are two important differences, with the first being that the baseband equivalent of the channel is actually independent of the coefficient that relates the baseband equivalents of our signal; this is the reason that the exact value of this factor is irrelevant. The second discrepancy stems from the fact that the baseband equivalent of the channel does not have the spectrum at the negative frequencies discarded, since it is just a shifted version of the passband frequency response,  $H(F)$ . Although this might seem absurd at first, we note that all the mathematical operations on the baseband model take place only inside the baseband bandwidth, while the rest of the frequencies are completely ignored. Specifically, we will see in our derivation of the respective discrete model, that the spectrum content outside the baseband  $[-W/2, W/2]$  is implicitly discarded.

## Discrete-Time

The most fundamental tool for the derivation of a discrete-time baseband equivalent channel is the Nyquist–Shannon sampling theorem, which states that any signal  $s(t)$  bandlimited to  $[-W/2, W/2]$ , can be described completely by its samples  $s(n/W)$  at rate  $W$ . In particular, by Shannon’s interpolation formula, in our case we have that

$$x^b(t) = \sum_{n=-\infty}^{\infty} x[n] \operatorname{sinc}(Wt - n) \quad (1.21)$$

where  $x[n] := x^b\left(\frac{n}{W}\right)$  and  $\operatorname{sinc}(t) := \frac{\sin(\pi t)}{\pi t}$

By plugging (1.21) into (1.14) we have that

$$y^b(t) = \sum_{i=0}^{N-1} a_i^b \sum_{n=-\infty}^{\infty} x[n] \operatorname{sinc}(Wt - W\tau_i - n) \Leftrightarrow \quad (1.22)$$

$$y^b(t) = \sum_{n=-\infty}^{\infty} x[n] \sum_{i=0}^{N-1} a_i^b \operatorname{sinc}(Wt - W\tau_i - n) \quad (1.23)$$

At this point, we sample the received signal,  $y(t)$ , at a rate of  $W$  to obtain its discrete counterpart; as in the case of the transmitted signal, we define  $y[m] := y^b\left(\frac{m}{W}\right)$ . We note that we are able to use the same sampling rate at both the transmitted and the received signal, because we have assumed an LTI channel. In practice, the bandwidth of the received signal,  $y(t)$ , might be slightly greater due to the Doppler spread.

$$y[m] = \sum_{n=-\infty}^{\infty} x[n] \sum_{i=0}^{N-1} a_i^b \operatorname{sinc}(m - W\tau_i - n) \stackrel{l:=m-n}{\Leftrightarrow} \quad (1.24)$$

$$y[m] = \sum_{l=-\infty}^{\infty} x[m-l] \sum_{i=0}^{N-1} a_i^b \operatorname{sinc}(l - W\tau_i) \Leftrightarrow \quad (1.25)$$

$$\boxed{y[m] = \sum_{l=-\infty}^{\infty} x[m-l] h_l} \quad (1.26)$$

where

$$h_l := \sum_{i=0}^{N-1} a_i^b \operatorname{sinc}(l - W\tau_i) \quad (1.27)$$

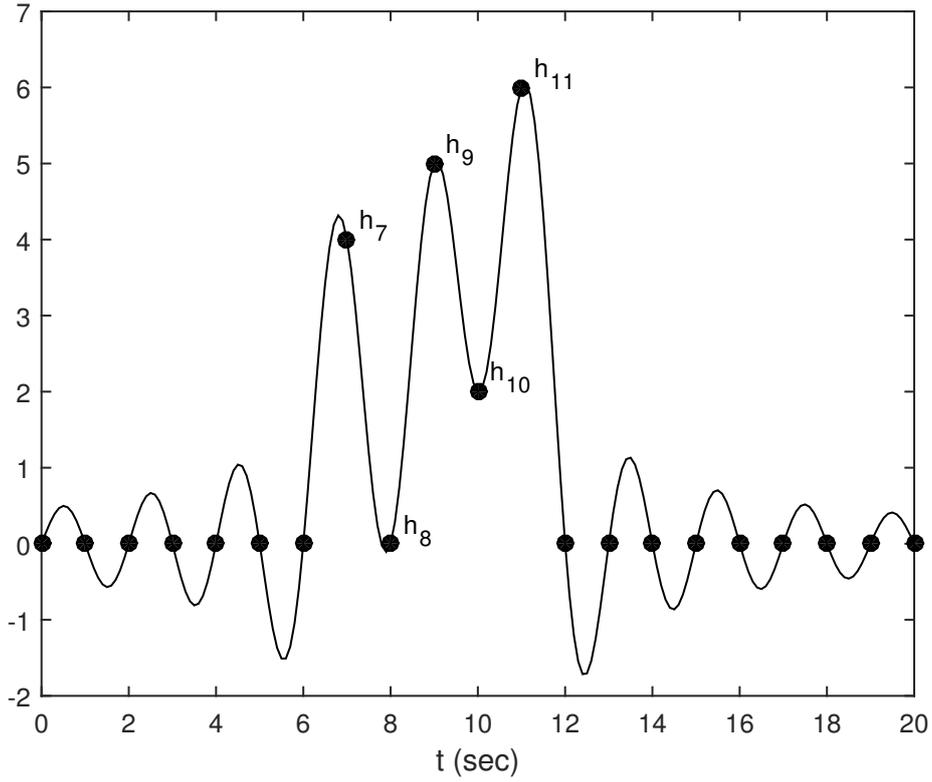
Looking at (1.27) we can understand that the channel taps,  $h_l$ , are samples of the sum of  $N$  scaled and time-shifted sinc pulses. We also notice that the shifts of the pulses depend only on the signal bandwidth and the delays of the various signal rays. Taking into account that a sinc pulse attenuates to zero as it tends to infinity at both sides, we can conclude that the sum of these sinc pulses will be a function that has non zero values only in a specific bounded time interval (see Figure 1.1). We can also manipulate (1.27) to get that

$$h_l = \sum_{i=0}^{N-1} a_i^b \operatorname{sinc} \left[ W \left( \frac{l}{W} - \tau_i \right) \right] \Leftrightarrow \quad (1.28)$$

$$h_l = \left[ \sum_{i=0}^{N-1} a_i^b \delta(t - \tau_i) \right] * \operatorname{sinc}(Wt) \Big|_{t=l/W} \Leftrightarrow \quad (1.29)$$

$$\boxed{h_l = h^b(t) * \operatorname{sinc}(Wt) \Big|_{t=l/W}} \quad (1.30)$$

Equation (1.30) can provide interesting insight as to how to interpret the channel taps,  $h_l$ , and how the discrete-time baseband equivalent channel relates to its continuous-time counterpart.



**Figure 1.1:** Plot of (1.27) for  $N = 4$ . We can see that only the  $h_7, h_9, h_{10}$  and  $h_{11}$  taps are non-zero.

Consider first the following Fourier transform

$$G(F) := \mathcal{F} \{h^b(t) * \text{sinc}(Wt)\} = H^b(F) \cdot \frac{1}{W} \Pi\left(\frac{F}{W}\right) \quad (1.31)$$

where

$$\Pi(F) = \begin{cases} 1 & , |F| \leq 0.5 \\ 0 & , \text{otherwise} \end{cases} \quad (1.32)$$

We see that we can simplify  $G(F)$  as

$$G(F) = \begin{cases} \frac{H^b(F)}{W} & , |F| \leq \frac{W}{2} \\ 0 & , \text{otherwise} \end{cases} \quad (1.33)$$

What (1.33) tells us is that the channel taps are essentially the samples at integer multiples of  $1/W$ , in the time domain, of the signal,  $g(t)$ , that corresponds to the low-pass filtered baseband channel,  $H^b(F)$ , divided by the bandwidth,  $W$ .

Furthermore, since  $g(t)$  is obviously sampled at the Nyquist frequency,  $W$ , we will be able to reconstruct it from its samples (the channel taps).

**Theorem 1.** Discrete-Time Fourier Transform

The Discrete-Time Fourier Transform (DTFT) transforms a sequence of  $N_c$  complex numbers  $s_0, s_1, \dots, s_{N_c-1}$  into a periodic function of frequency,  $X_d(f)$ ,

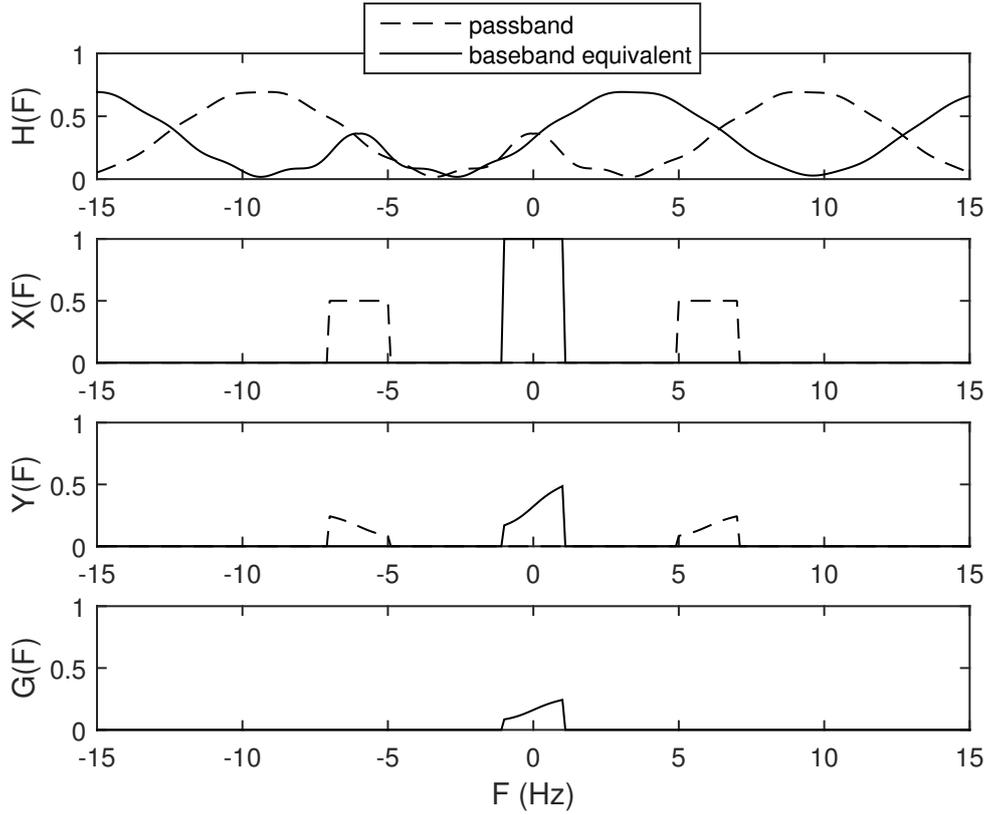


Figure 1.2

as follows

$$X_d(f) = \sum_{l=-\infty}^{+\infty} s_l e^{-j2\pi fl} \quad (1.34)$$

Hence, we have that

$$G_d(f) = \sum_{l=-\infty}^{+\infty} h_l e^{-j2\pi fl} \quad (1.35)$$

We also know that the CTFT,  $G(F)$ , and the DTFT,  $G_d(f)$ , of  $g(t)$  are related as follows

$$G_d\left(\frac{F}{W}\right) = W \cdot G(F) \quad , F \in \left[-\frac{W}{2}, \frac{W}{2}\right) \quad (1.36)$$

which, in combination with (1.33), gives that

$$G_d\left(\frac{F}{W}\right) = H^b(F) \quad , F \in \left[-\frac{W}{2}, \frac{W}{2}\right) \quad (1.37)$$

**Definition 2.** Discrete Fourier Transform

The Discrete Fourier Transform (DFT) transforms a sequence of  $N$  complex numbers  $s_0, s_1, \dots, s_{N-1}$  into another sequence of  $M$  complex numbers

$\tilde{s}_0, \tilde{s}_1, \dots, \tilde{s}_{M-1}$  as follows

$$\tilde{s}_m = \sum_{n=0}^{N-1} s_n e^{-\frac{j2\pi nm}{M}} \quad (1.38)$$

Note that this not the standard definition, and it implies zero-padding the sequence  $\{s_n\}$  when  $M > N$ .

Another interesting observation is that the DFT,  $\{\tilde{h}_l\}$ , of the channel taps sequence,  $\{h_l\}$ , consists in fact of samples of the DTFT of  $g(t)$  (we remind that  $h_l = g(l/W)$ ) which can be shown as follows

$$G_d(f) = \sum_{l=-\infty}^{+\infty} h_l e^{-j2\pi fl} \Rightarrow \quad (1.39)$$

$$G_d(f) = \sum_{l=0}^{L-1} h_l e^{-j2\pi fl} \Rightarrow \quad (1.40)$$

$$G_d\left(\frac{n}{L}\right) = \sum_{l=0}^{L-1} h_l e^{-\frac{j2\pi nl}{L}} \Rightarrow \quad (1.41)$$

$$G_d\left(\frac{n}{L}\right) = \tilde{h}_n \quad , n = 0, 1, 2, \dots, L-1 \quad (1.42)$$

and due to the periodicity of  $G_d$  we have that

$$\tilde{h}_n = \begin{cases} G_d\left(\frac{n}{L}\right) & , n = 0, 1, 2, \dots, \left[\frac{L}{2}\right] - 1 \\ G_d\left(\frac{n}{L} - 1\right) & , n = \left[\frac{L}{2}\right], \left[\frac{L}{2}\right] + 1, \left[\frac{L}{2}\right] + 2, \dots, L-1 \end{cases} \quad (1.43)$$

which along with (1.37) leads to the following

$$\tilde{h}_n = \begin{cases} H^b\left(\frac{nW}{L}\right) & , n = 0, 1, 2, \dots, \left[\frac{L}{2}\right] - 1 \\ H^b\left(\frac{nW}{L} - W\right) & , n = \left[\frac{L}{2}\right], \left[\frac{L}{2}\right] + 1, \left[\frac{L}{2}\right] + 2, \dots, L-1 \end{cases} \quad (1.44)$$

This result shows that the DFT,  $\{\tilde{h}_n\}$ , of the channel taps  $\{h_n\}$ , are just samples of the baseband equivalent of the channel frequency response,  $H^b(F)$ , in the baseband  $[-W/2, W/2)$ . Of course, since  $H^b(F)$  is just a frequency-shifted version of the passband channel frequency response,  $H(F)$ , we can directly calculate the sequence  $\{\tilde{h}_n\}$  just by sampling  $H(F)$  in the passband  $[F_c - W/2, F_c + W/2)$  at integer multiples of  $W/L$ .

Formula (1.44) will come in really useful in the derivation of the optimal power allocation algorithm in frequency-selective channels (see section 3.3).

## 1.2 Antennas

### 1.2.1 Phasor Representation

Consider the following RL circuit

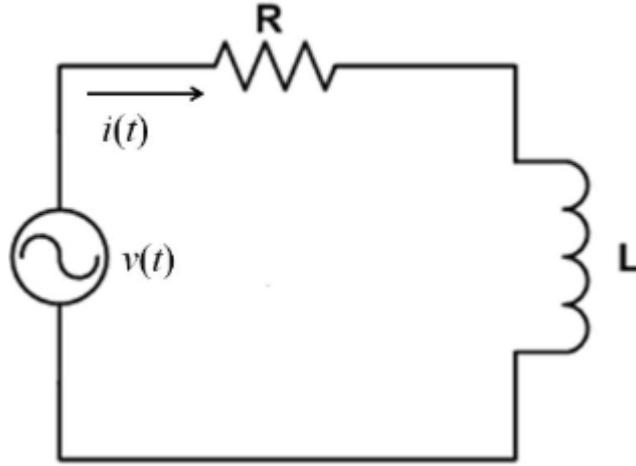


Figure 1.3

We know from basic circuit analysis that this circuit is described by the following equation

$$v(t) = Ri(t) + L \frac{di(t)}{dt} \quad (1.45)$$

Hence, we need to employ differential calculus in order to obtain a solution for the voltage,  $v(t)$ , and the current,  $i(t)$ . Even though in this case the equation may be simple to solve, it turns out that in more complex circuits the differential equations are sometimes very difficult or even impossible to solve.

For this reason, it is common practice to manipulate the circuit equations using smart techniques such as the Laplace and phasor transforms.

**Definition 3.** A sinusoidal signal,  $s(t) = A \cdot \cos(2\pi F_c t + \theta)$ , and its phasor,  $\mathbf{S}$ , are related by the following formula

$$s(t) = \Re\{\mathbf{S} \cdot e^{j2\pi F_c t}\} \quad (1.46)$$

**Corollary 3.1.** By Euler's formula we obtain

$$\begin{aligned} A \cos(2\pi F_c t + \theta) &= A \frac{e^{j(2\pi F_c t + \theta)} + e^{-j(2\pi F_c t + \theta)}}{2} \Leftrightarrow \\ s(t) &= A \frac{e^{j(2\pi F_c t + \theta)} + e^{-j(2\pi F_c t + \theta)}}{2} \Leftrightarrow \\ s(t) &= A \frac{1}{2} 2\Re\{e^{j(2\pi F_c t + \theta)}\} \Leftrightarrow \\ s(t) &= \Re\{Ae^{j\theta} \cdot e^{j2\pi F_c t}\} \Leftrightarrow \\ \mathbf{S} &= Ae^{j\theta} \end{aligned}$$

In fact, loosely speaking, the phasor transform is just the Laplace transform when the signals involved in the circuit are all sinusoidal of the same frequency. Keep in mind that it makes no sense to talk about the phasor of a non sinusoidal signal.

Also, note that by the way that the phasor was defined, its magnitude corresponds to the peak value of the signal and not the RMS value. We remind that the main reason that RMS values are employed in circuit analysis is the fact that we like the power formula VI to be defined similarly in the phasor domain. If peak values are used instead, then we need to add a coefficient of 0.5 in the phasor domain power formula.

We further know from circuit analysis that, when we deal with sinusoidal power sources at a specific frequency,  $F_c$ , then all the measured voltages and currents in the circuit will also be sinusoidal signals of the same frequency. Hence, in the case of (1.45) we get the following

$$V \cos(2\pi F_c t + \theta) = R \cdot I \cos(2\pi F_c t + \phi) + L \frac{d I \cos(2\pi F_c t + \phi)}{dt} \quad (1.47)$$

where  $v(t) = V \cos(2\pi F_c t + \theta)$  and  $i(t) = I \cos(2\pi F_c t + \phi)$ . Plugging now the phasor representation of  $v(t)$  and  $i(t)$  into (1.47), we get that

$$\begin{aligned} \Re\{\mathbf{V} \cdot e^{j2\pi F_c t}\} &= R \cdot \Re\{\mathbf{I} \cdot e^{j2\pi F_c t}\} + L \frac{d \Re\{\mathbf{I} \cdot e^{j2\pi F_c t}\}}{dt} \Leftrightarrow \\ \Re\{\mathbf{V} \cdot e^{j2\pi F_c t}\} &= \Re\{R \cdot \mathbf{I} \cdot e^{j2\pi F_c t}\} + \Re\left\{L \frac{d \mathbf{I} \cdot e^{j2\pi F_c t}}{dt}\right\} \Leftrightarrow \\ \Re\{\mathbf{V} \cdot e^{j2\pi F_c t}\} &= \Re\{R \cdot \mathbf{I} \cdot e^{j2\pi F_c t} + L \mathbf{I} \cdot e^{j2\pi F_c t} j2\pi F_c\} \Leftrightarrow \\ \Re\{\mathbf{V} \cdot e^{j2\pi F_c t}\} &= \Re\{(R\mathbf{I} + L\mathbf{I} \cdot j2\pi F_c) e^{j2\pi F_c t}\} \end{aligned}$$

It can be proved that we can work directly with the phasors as follows

$$\begin{aligned} \mathbf{V} &= R\mathbf{I} + L\mathbf{I} \cdot j2\pi F_c \Leftrightarrow \\ \mathbf{V} &= (R + Lj2\pi F_c)\mathbf{I} \Leftrightarrow \\ \mathbf{I} &= \frac{1}{R + Lj2\pi F_c} \mathbf{V} \end{aligned} \quad (1.48)$$

Interestingly, we observe that the differential equation is now transformed to a simple algebraic equation involving only summation and multiplication. This makes our life a lot easier! In fact, we can apply the inverse phasor transform in (1.48) and obtain an expression for  $i(t)$  in the time domain. Further, the coefficient  $1/(R + Lj2\pi F_c)$  is referred to as the current-to-voltage transfer function of the circuit. Finally, we will refer to the quantity  $Lj2\pi F_c$  as the impedance of the inductor, which is essentially an equivalent of the concept of resistance. This equivalence stems from the fact that the product of the inductor impedance with the phasor of the voltage across the inductor gives the phasor of the current that flows through the inductor, which is similar to what Ohm's law state. This concept can also be extended to capacitors, and hence in the phasor domain every component can be represented by an equivalent "resistance" which we will refer to as its (complex) impedance.

## 1.2.2 Dipole Antennas

### Dipole Equivalent Circuit

The simplest form of antenna is the so-called dipole. As we can see in Figure 1.4, loosely speaking, an antenna dipole is just the wires of a circuit that had been

bent appropriately so that they are collinear. This way the current flows back and forth on the axis that these wires form, and consequently electromagnetic waves are produced and propagate.

It would be reasonable to wonder now, how is it even possible for current to flow if we cut the wires and bend them? The answer lies to the fact that even though we have formed an open-circuit, there is still current that flows between the two parallel disconnected wires. In order to keep it short, we will not elaborate more on the subject and we urge the reader to study the area of transmission lines if more information is needed.

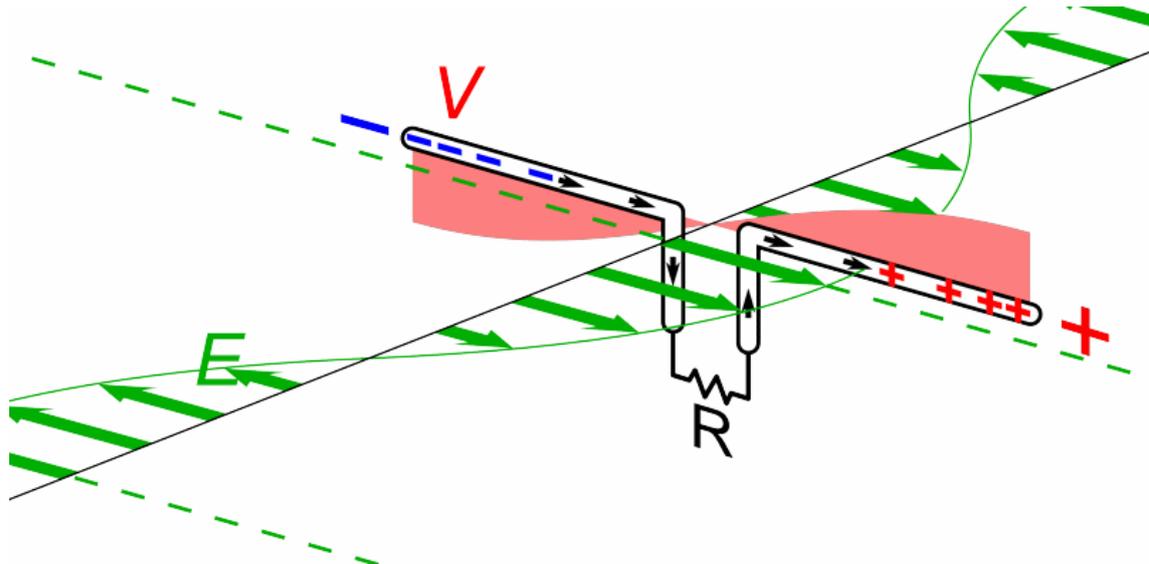


Figure 1.4: Source: [https://en.wikipedia.org/wiki/Dipole\\_antenna](https://en.wikipedia.org/wiki/Dipole_antenna)

Now that we have briefly seen how and why a simple antenna works, we continue to the construction of a simple wireless communication system with a transmitter and a receiver, each equipped with its own dipole antenna as shown in Figure 1.5.

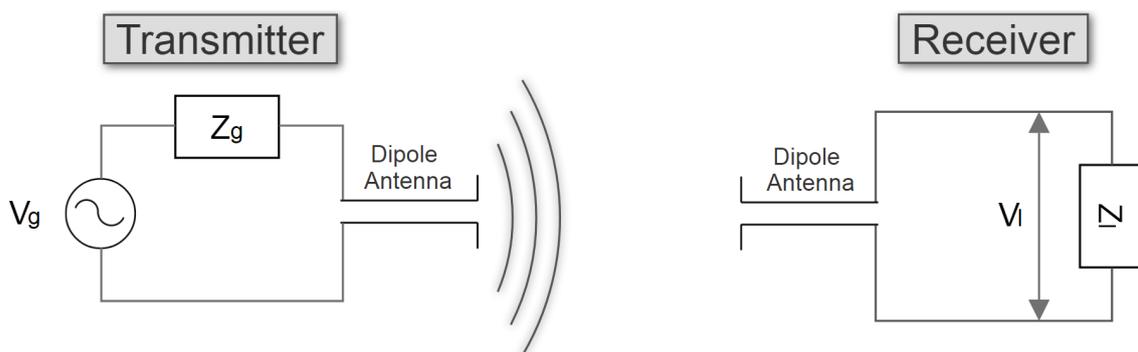


Figure 1.5

At the transmitter we have a voltage source,  $V_g$ , which carries the desired signal for transmission, a source impedance,  $Z_g$ , which is an inherent impedance that most of voltage sources have in antenna systems, and a dipole antenna which will convert our signal from its voltage form to propagating electromagnetic waves.

When these waves reach the receiver, its dipole antenna is going to convert the signal back to its voltage form and consequently we can attach to the antenna a

receiving system which is going to do further post-processing on the signal. We note that when this system is attached to the antenna, then from the antenna perspective it can be represented as a complex impedance which describes the overall influence of all the electronic circuitry of the system at its input. We will refer to this impedance as the load impedance,  $Z_l$ .

Similarly, we can obtain an equivalent impedance for each dipole, which has a real part that represents the power that is converted to electromagnetic waves plus the ohmic losses on the dipole, and an imaginary part which describes potential capacitive or inductive behavior of the dipole. We can further show that every kind of antenna can be represented by an equivalent impedance, which we will refer to as the antenna input impedance. We have at this point discovered an equivalent circuit for the whole communication system as shown in Figure 1.6.



Figure 1.6

### Maximum Power Transfer

As already mentioned, each voltage source,  $V_g$ , will be accompanied by the respective source impedance,  $Z_g$ , in which some power will be dissipated, and thus not all the power ends up in our antenna. It can be proved that we will have maximum power transfer to the antenna only when  $Z_g = Z_{antTx}^*$ , and this power can be calculated as

$$P_{avail} = \frac{|V_g|^2}{8\Re\{Z_g\}} \quad (1.49)$$

This selection of impedances is usually referred to as *impedance matching*.

We can now define the *mismatched radiation efficiency*,  $e_{rad}^{mis}$ , of the antenna as

$$e_{rad}^{mis} = \frac{P_{rad}}{P_{avail}} \quad (1.50)$$

where  $P_{rad}$  is the radiated power that corresponds to the transmitted electromagnetic waves.

We can also define the *matched radiation efficiency*,  $e_{rad}^{mat}$ , of the antenna as

$$e_{rad}^{mat} = \frac{P_{rad}}{P_{in}} \quad (1.51)$$

where  $P_{in}$  is the total power that is consumed by the antenna (radiated power plus ohmic losses)

### Antenna Noise

When the signal electromagnetic waves reach the receiver, the signal will be converted back to voltage with some noise added. This noise is usually modeled as additive white Gaussian noise (AWGN) and it is dependent on the *antenna temperature*.

Antenna temperature is a parameter that describes how much noise an antenna produces in a given environment. This temperature is not the physical temperature of the antenna. Moreover, an antenna does not have an intrinsic “antenna temperature” associated with it; rather the temperature depends on its gain pattern and the thermal environment that it is placed in. Antenna temperature is also sometimes referred to as Antenna Noise Temperature.

The power of the noise can be expressed as

$$P_{noise} = kTB \tag{1.52}$$

where  $T$  is the antenna temperature in Kelvin,  $B$  is the bandwidth over which the receiving antenna is operating, and  $k$  is the Boltzmann constant, which is approximately equal to  $1.38064852 \cdot 10^{-23}$  Joule/Kelvin.

# Chapter 2

## MIMObit

### 2.1 Basics Overview

The first window that appears after MIMObit is launched is shown in Figure 2.1. At this stage the user has the option to either create a new configuration file, or load an existing one. The configuration file is an important component of the software since it contains the settings of several fundamental parameters that can affect any kind of simulation results.

In this overview a new configuration file is created by clicking on the "New..." button.

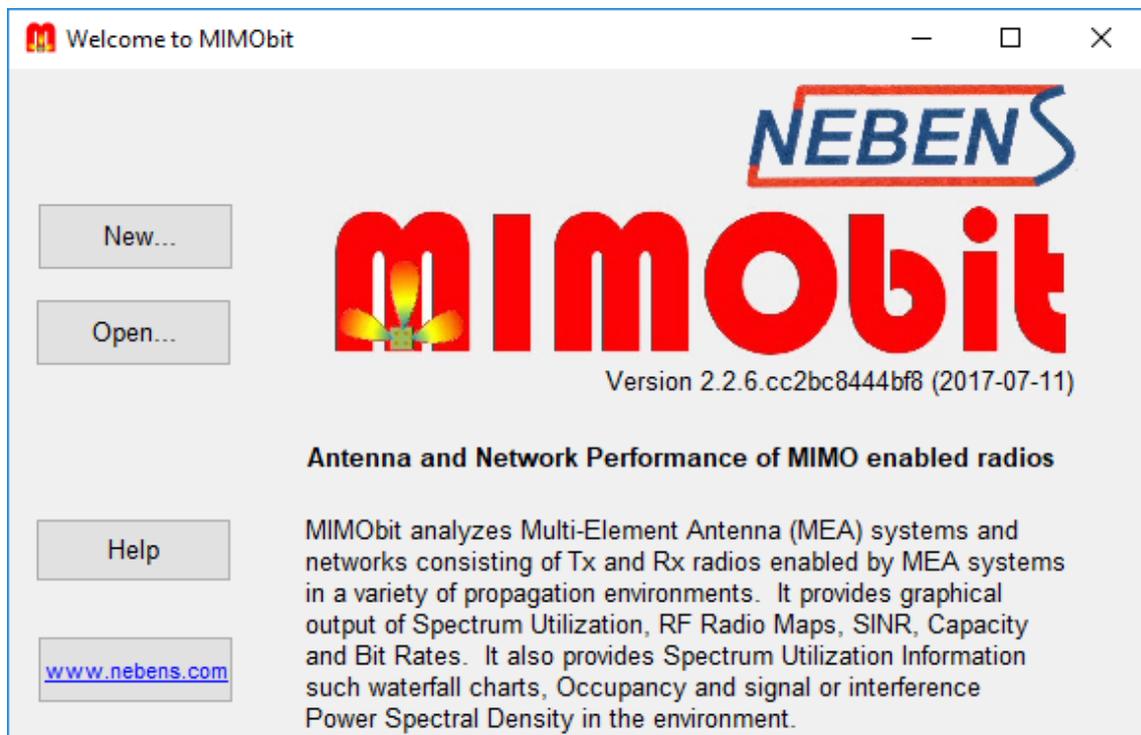


Figure 2.1

Next, a new window appears, where we are asked to specify a name and a path for the configuration file (Figure 2.2). *It is highly recommended that, from this point on, every new file is saved at the directory where the configuration file is located, in order to avoid unexpected malfunction of the software.*

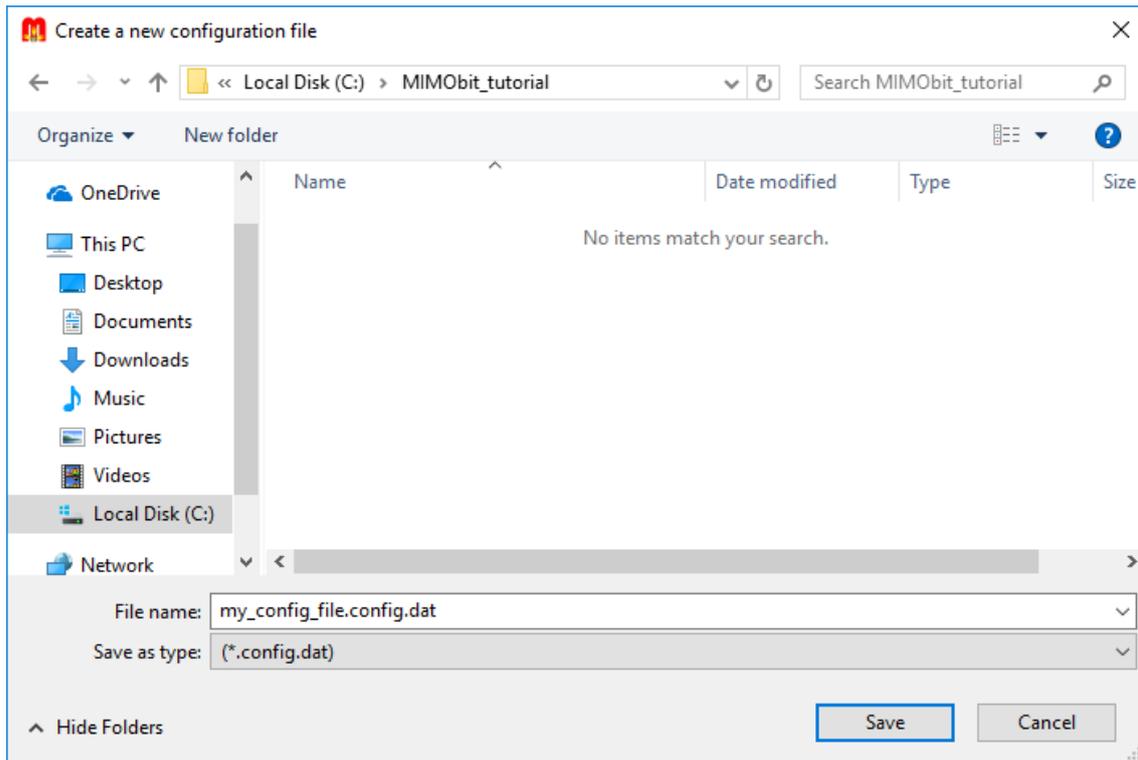


Figure 2.2

After the file is saved, the two main windows of the software show up (Figure 2.3 and Figure 2.4). At the window in Figure 2.3 we can access and inspect or alter almost every simulation parameter, and finally run the simulation. The window in Figure 2.4 is the status window which displays everything that happens before, during and after a simulation. In general, the top section provides real time information about any event, whereas the bottom section displays mainly warnings and errors that may occur at any time.

To inspect and set up desired simulations properties, the user can edit the configuration file by selecting the respective option as shown in Figure 2.5. Immediately afterwards, a new window shows up (see Figure 2.6), where important software parameters can be configured. In our example, we are going to change the value of “Sample size” parameter to 2.

*We should mention that not all configuration file parameters can be accessed via this graphical interface. There exist several hidden parameters that can only be edited by opening the configuration file manually in a text editor. Only the most common parameters appear in this graphical environment.*

MIMObit neatly breaks down the complete wireless communication process in five basic components which are independent of each other. An overview of these components is presented below, while more detailed information is provided in section 2.2.

- **Transmitters**

Transmitters pertain to the specification of the properties of the signal transmitters and the interferers. Some of these properties refer to the location of the transmitting antennas in the three dimensional space, the physical properties of these antennas including the voltages by which they are excited, and

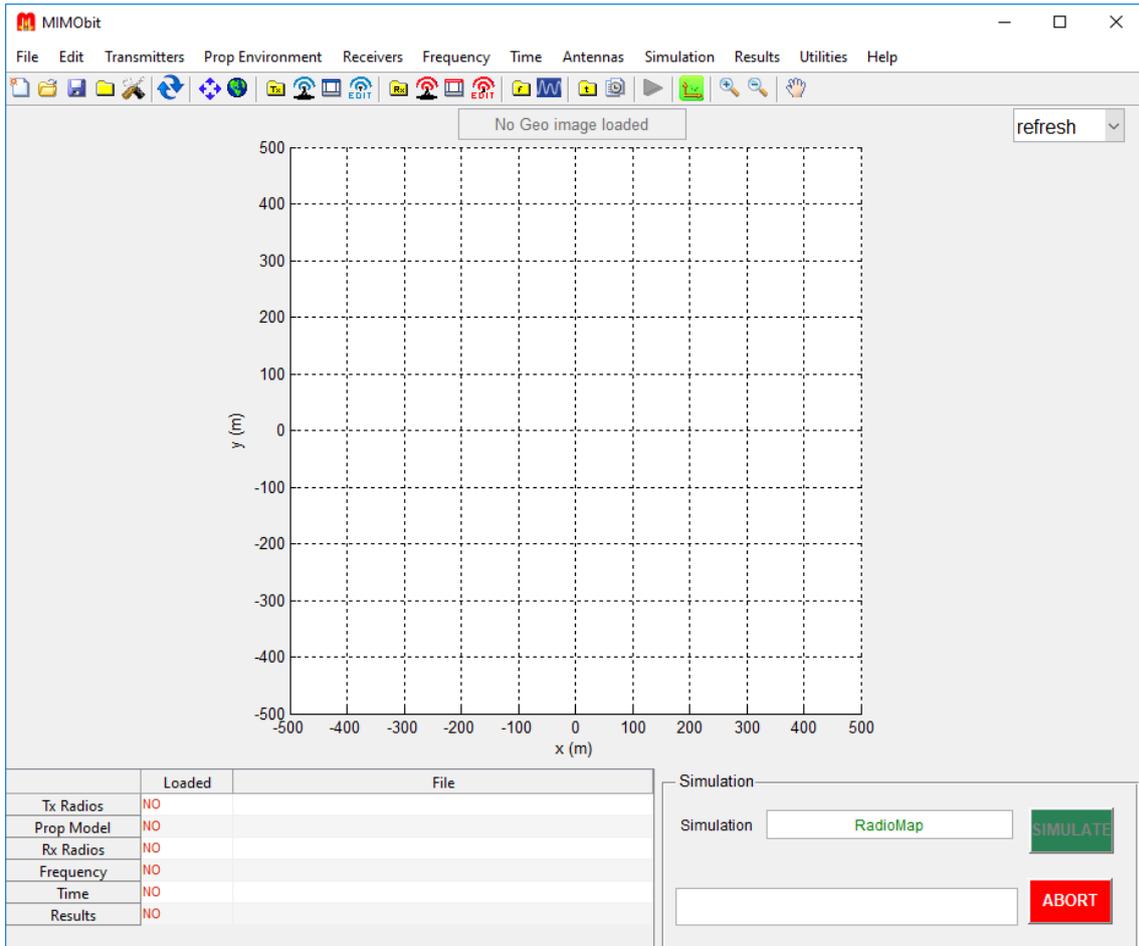


Figure 2.3

finally the bandwidth over which we transmit our signal as well as its power distribution over this bandwidth.

- **Propagation Environment**

The Propagation Environment refers to the way that the electromagnetic waves are transmitted through the physical environment until they reach the receiver. Depending on the environment the waves can be diffracted, refracted, reflected, absorbed, or of course reach the receiver directly from the transmitter if there is a line of sight. Hence, depending on the physical environment that the antennas are set to communicate in, there will be plenty of scenarios on the ways that each of these waves travel and reach the receiver, and thus a different behavior of the observed received signal is expected in each case. It is probably straightforward at this point that there are conceptual and computational complications in obtaining a complete and accurate description of every real life physical environment in which we plan to set up our antenna systems. To cope with this adversity we usually resort to idealized representations of the waves propagation, where mathematical symmetries and closed form formulas can be exploited. This way we become capable of performing heavy and complex mathematical calculations and transformations on these models, and therefore obtain several informative quantities about our communication system, such as Shannon's Capacity. Some of these models are deterministic

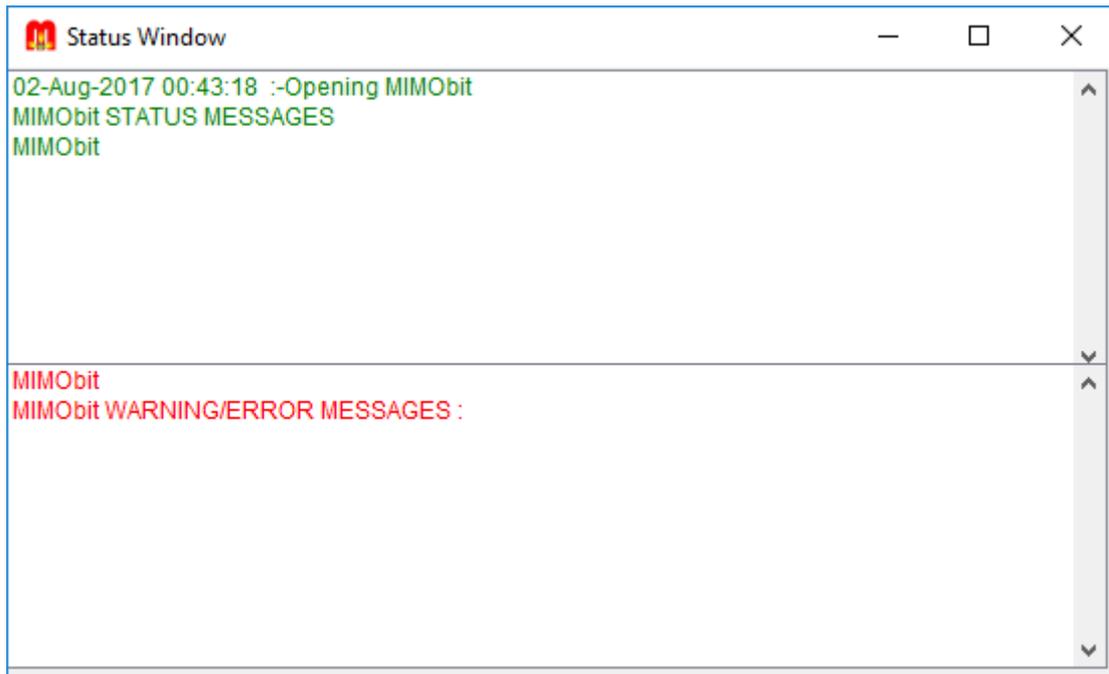


Figure 2.4

and simple, like the Line of Sight and the Flat Earth/Two Ray models, and some of them are complex probabilistic approximations. Some examples of the latter include models like the Rayleigh (which MIMObit attempts to approximate by its “iid” propagation environment), the 3GPP 3D and the New Radio propagation environments. It is noted that, till the time that this thesis was written, the communication channels in MIMObit are considered static, which means that the propagation environment is not altered as the time goes by.

- **Receivers**

Receivers pertain to the specification of the properties of the signal receivers. Some of these properties refer to the location of the receiving antennas in the three dimensional space, the physical properties of these antennas, and finally the resolution bandwidth of the receiving system.

- **Frequency**

The Frequency component refers to the specific frequencies at which the physical communication system is simulated. As explained in subsection 1.2.1, the whole physics and circuit analysis is performed assuming that only sinusoidal signals of a specific frequency exist in our system.

- **Time**

The Time component is incorporated so that the user can conduct certain studies where in essence the signal transmitted from each radio is simply scaled by a factor at each time instance. In order to avoid confusion, one can completely dismiss the concept of time, and think of the time instances as a group of different experiments which only differ in that the transmitted signal of each radio is multiplied by a certain factor. It is noted that the time instances are

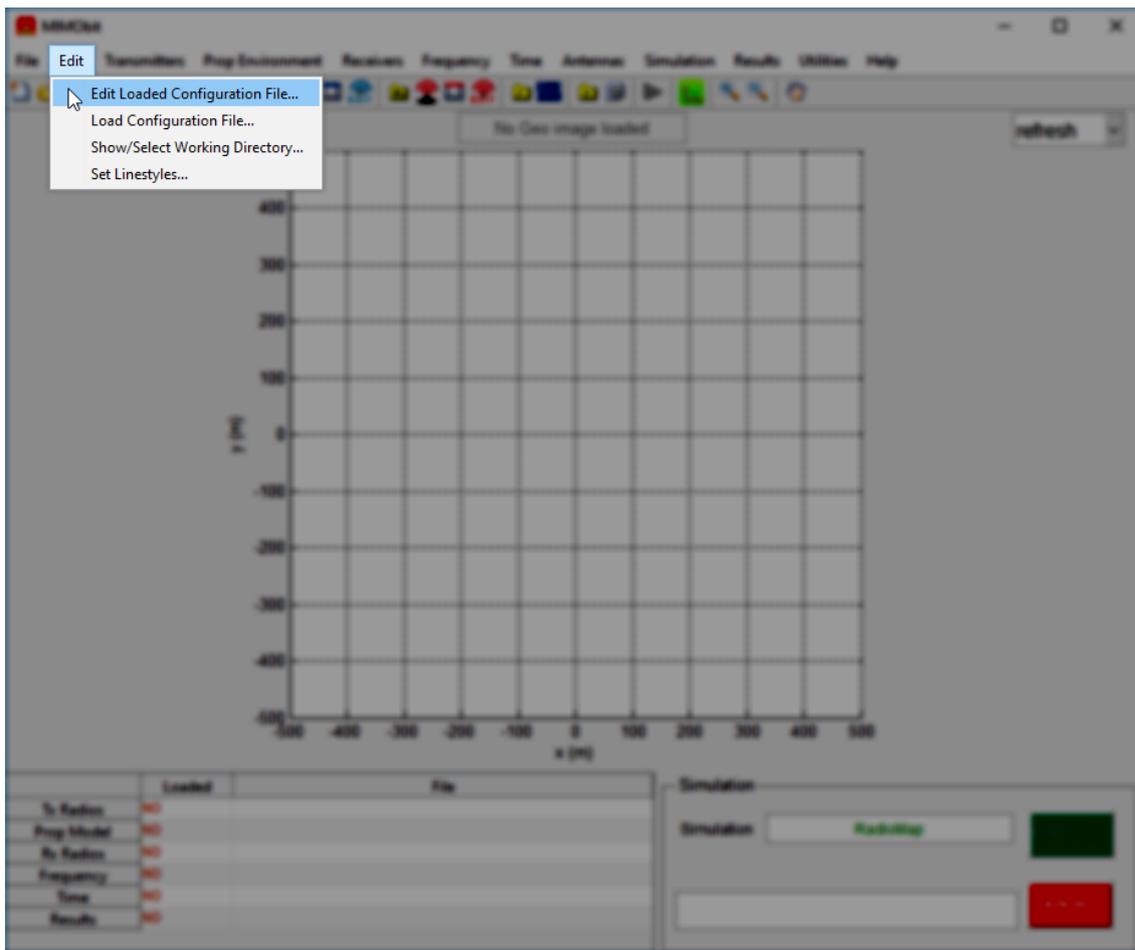


Figure 2.5

not related in any way to the propagation environment, and consequently they cannot affect the form of the communication channel.

These entities are in turn thoroughly represented by specific files which we will now need to create and load.

To create the **transmitters** files we first click on the "Individual Transmitters..." option as shown in Figure 2.7, and then a window with all the transmitters parameters appears (Figure 2.8). In this window, the first thing that we need to set is the "Number of Tx Radios" parameter, which in our case is one. Afterwards, we set up the rest of the parameters (in our example we proceed with the default values), and click the "Save Transmitter" button. After that, the "Tx Radio id" generally changes to the id of the next radio which we also configure and save. When all the transmitters (only one in our example) are saved, we click the "Save & Load" button, and a new window appears (Figure 2.9) asking for the name and the path where the transmitters files will be stored. It is noted that the transmitted signal for each transmitter is also implicitly specified by the "Power Mask" and the "BW" options. For more information on the Transmitters component see subsection 2.2.2.

In a completely similar fashion we can also set up and save the **receivers** files specifying only one receiver and altering the x coordinate to 100 meters. We leave the rest of the settings to their default state. For more information on the Receivers component see subsection 2.2.3.

Figure 2.6

The next important step is to select the desired **propagation environment**. In our example we opt for the New Radio UMi NLOS model as shown in Figure 2.10.

Hence, at this point we have completely specified the communication system properties and the form of the signal that will be transmitted.

Afterwards, we set up the **frequency** component by clicking on the "New Frequency File..." option as shown in Figure 2.11, and after the window in Figure 2.12 shows up, we define the desired simulation frequencies and click on the "Save & Load" button; in our case we leave the frequencies to their default values. Clicking on the "Test" button just displays every single frequency that has been specified, which might come in helpful when we use compact parametric notation. The concept of the simulation frequency may seem a bit vague at first, but hopeful it will get clear in subsection 2.2.1. In a completely similar fashion we can set up and save the **time** file using the default settings, and finally we choose the "PropStats" simulation type, as shown in Figure 2.13. The three **simulation types** provide us with different simulation results, depending on what quantities and properties of the communication system we need to study.

The main MIMObit window should now look like the Figure 2.14, where we can clearly see the location of the transmitter (blue circle) and the receiver (red circle), and that all the required files are loaded as shown at the bottom-left corner of the window. The "Results" file will be loaded right after the simulation is completed.

Before we proceed to the simulation, let us take a look at Figure 2.15 and Table 2.1, which show all the files that have been created in our project folder and

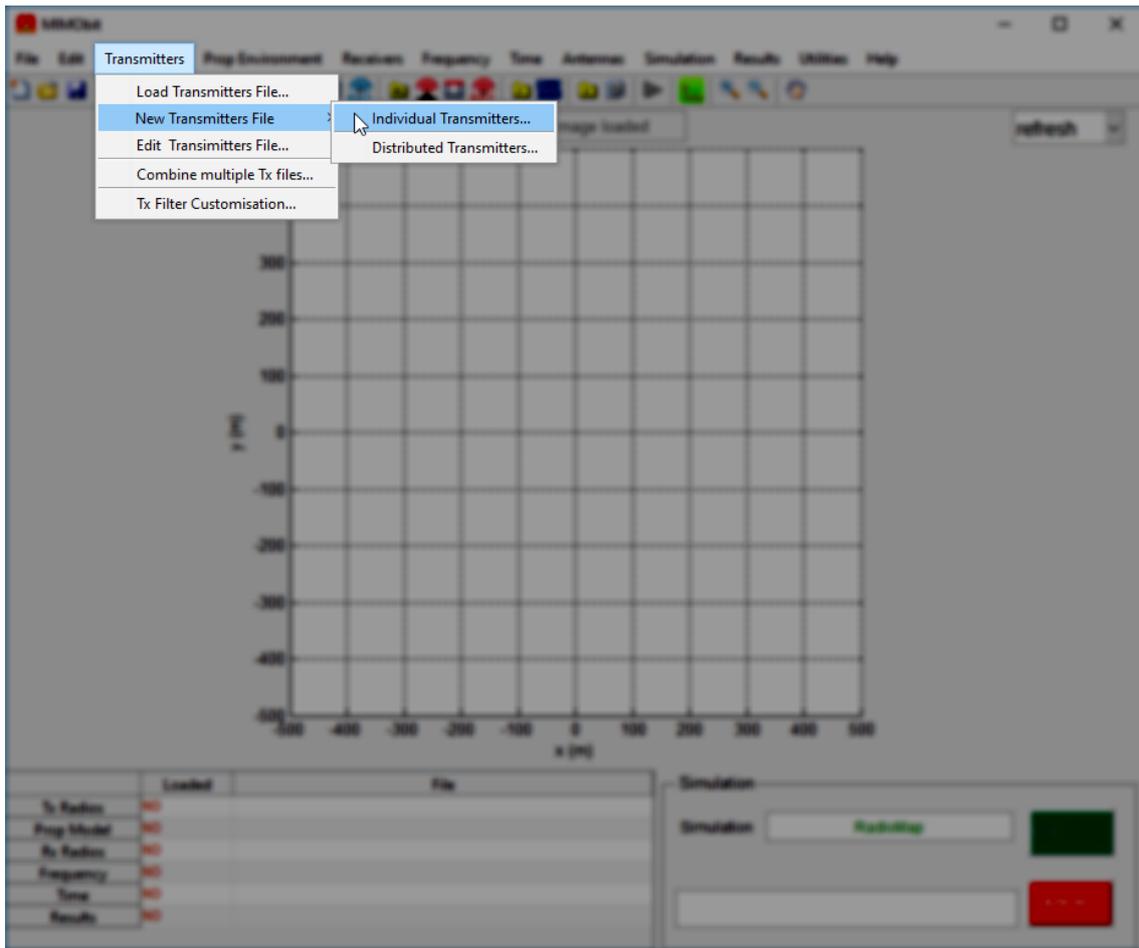


Figure 2.7

how they correspond to the components discussed so far, respectively. We are now ready to run our first simulation by clicking on the green simulation button.

When the simulation completes, a new window appears as shown in Figure 2.16 where we can obtain information about quantities such as capacity and throughput. There is also the option to alter some of the parameters that were set before the simulation, and then update the results quickly, without having to reproduce the whole simulation again. It is also important to mention that during the simulation three new files were created, as shown in Figure 2.17, which contain interesting information that characterize the channels involved in our communication system. For more information on the simulation results, please refer to section 2.2.4.

## 2.2 Comprehensive Guide

### 2.2.1 Structure Concepts

In this section we discuss some of the fundamental concepts that define the structure and the function of the software, and how these concepts are related to the abstract world of signals & systems and information theory.



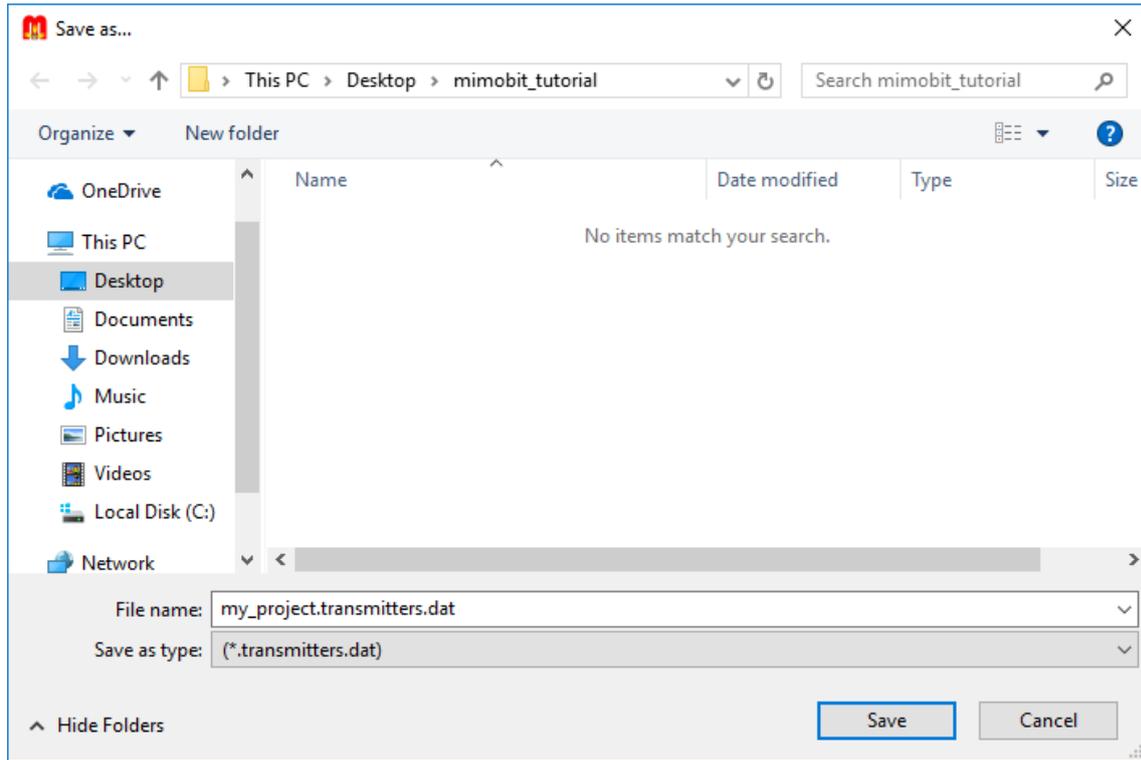


Figure 2.9

Interestingly, the coefficients  $H_{mn}$  are actually the source-to-load voltage transfer functions between the  $n$ -th port of the transmitter and the  $m$ -th port of the receiver, when only this specific port of the transmitter is excited. In this case, we have that

$$\mathbf{V}_{l_m} = H_{mn} \mathbf{V}_{g_n} \quad (2.2)$$

If we now account for the principle of superposition, which in our case states that the total voltage induced across the load of  $m$ -th port of the receiver will be the sum of the voltages on this load caused by each transmitter source voltage individually, then we can verify the validity of the result in (2.1).

Moreover, we can reformulate (2.1) as follows

$$\begin{bmatrix} \mathbf{V}_{l_1} \\ \mathbf{V}_{l_2} \\ \mathbf{V}_{l_3} \\ \vdots \\ \mathbf{V}_{l_M} \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^N H_{1n} \mathbf{V}_{g_n} \\ \sum_{n=1}^N H_{2n} \mathbf{V}_{g_n} \\ \sum_{n=1}^N H_{3n} \mathbf{V}_{g_n} \\ \vdots \\ \sum_{n=1}^N H_{Mn} \mathbf{V}_{g_n} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1N} \\ H_{21} & H_{22} & & \\ \vdots & & \ddots & \\ H_{M1} & & & H_{MN} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{g_1} \\ \mathbf{V}_{g_2} \\ \mathbf{V}_{g_3} \\ \vdots \\ \mathbf{V}_{g_N} \end{bmatrix}$$

or more compactly

$$\mathbf{V}_l = \mathbf{H} \cdot \mathbf{V}_g \quad (2.3)$$

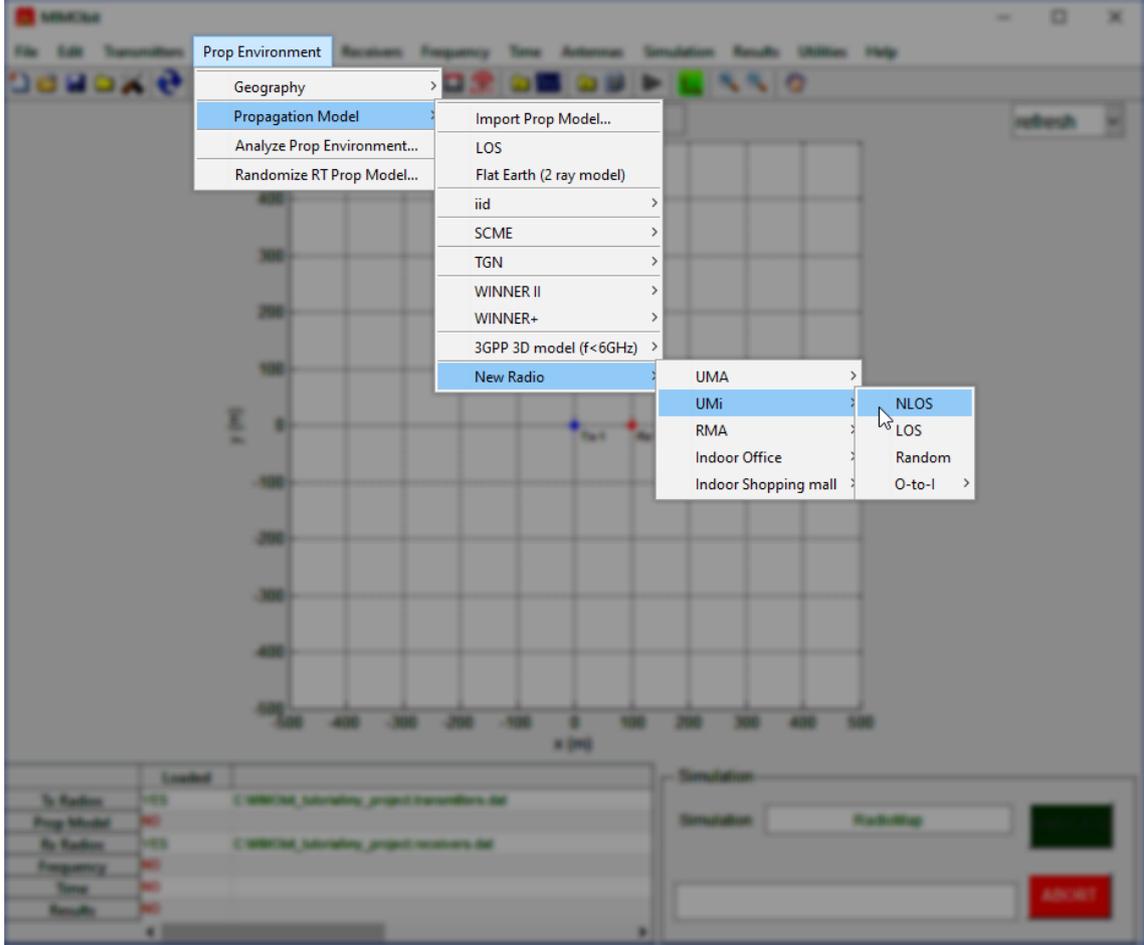


Figure 2.10

where

$$\mathbf{V}_l = \begin{bmatrix} \mathbf{V}_{l_1} \\ \mathbf{V}_{l_2} \\ \mathbf{V}_{l_3} \\ \vdots \\ \mathbf{V}_{l_M} \end{bmatrix}, \mathbf{V}_g = \begin{bmatrix} \mathbf{V}_{g_1} \\ \mathbf{V}_{g_2} \\ \mathbf{V}_{g_3} \\ \vdots \\ \mathbf{V}_{g_N} \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1N} \\ H_{21} & H_{22} & & \\ \vdots & & \ddots & \\ H_{M1} & & & H_{MN} \end{bmatrix}$$

Hence,  $\mathbf{H}$  is just the transfer function matrix of our Multiple-Input Multiple-Output (MIMO) antenna system, with elements the coefficients  $H_{mn}$ . (This is the reason why these coefficients are often referred to as “channels” in the software manual.)

During a simulation, MIMObit will calculate these coefficients for every pair of transmitter-receiver ports, and this process will be repeated for every frequency that has been specified in the frequency file (\*.frequencies.dat) (see RadMap App in section 2.2.4); this leads to all the combinations (frequencies x Tx ports x Rx ports). In the case of probabilistic propagation environments, there is also the option to produce different realizations of the environment, and then repeat the calculations above for each one of the realizations (see PropStats App in section 2.2.4); this time one coefficient will be produced for each combination (realizations x frequencies x Tx ports x Rx ports). There is also a third option where extra combinations are considered, for various rotations of the receiver (see OrientStats App in section 2.2.4),

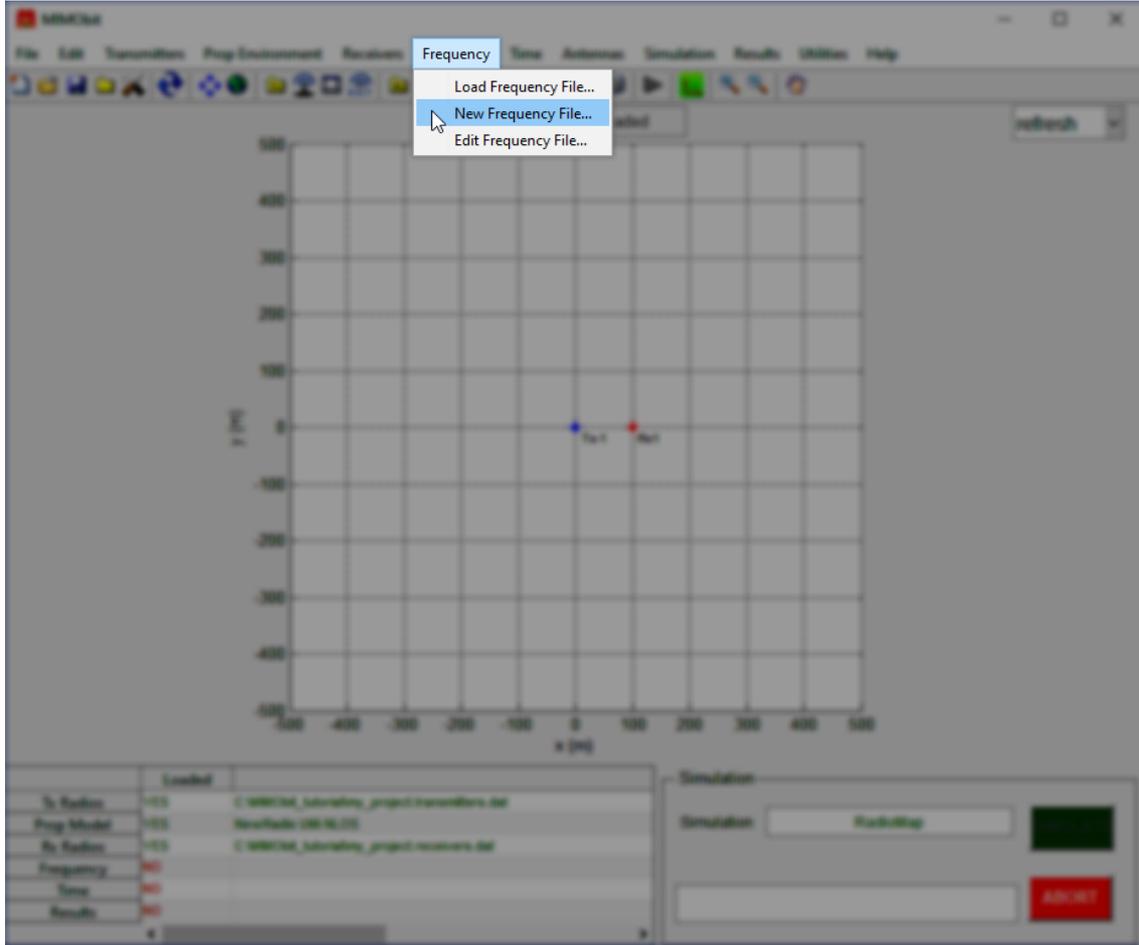


Figure 2.11

and thus a coefficient is produced for each pair (Rx rotations x realizations x frequencies x Tx ports x Rx ports). MIMObit stores all of this information along with the transfer functions coefficients in specific files (\*.Htotalresults.dat), which are discussed in section 2.2.4 in more detail.

Finally, MIMObit utilizes the transfer function matrix,  $\mathbf{H}$ , for post-processing, along with the rest of the parameters that we have set prior to the simulation, in order to provide us with important and meaningful results such as the antenna system capacity, throughput and RF radio map.

### Single-Input Single-Output Antenna Systems (SISO)

Now that we have discussed how MIMObit operates in a nutshell, we are going to focus mostly on communication systems with a single 1-port transmitter and a single 1-port receiver. Hence, our system will be completely described as follows

$$\mathbf{V}_l = G(F_c) \cdot \mathbf{V}_g \quad (2.4)$$

where  $G(F)$  is the transfer function,  $\mathbf{H} = H_{11}$ , of the system, calculated at a frequency  $F_c$ .

The first thing that we should notice, is that voltages in MIMObit are formulated using phasors, which is generally a standard procedure in circuit analysis as

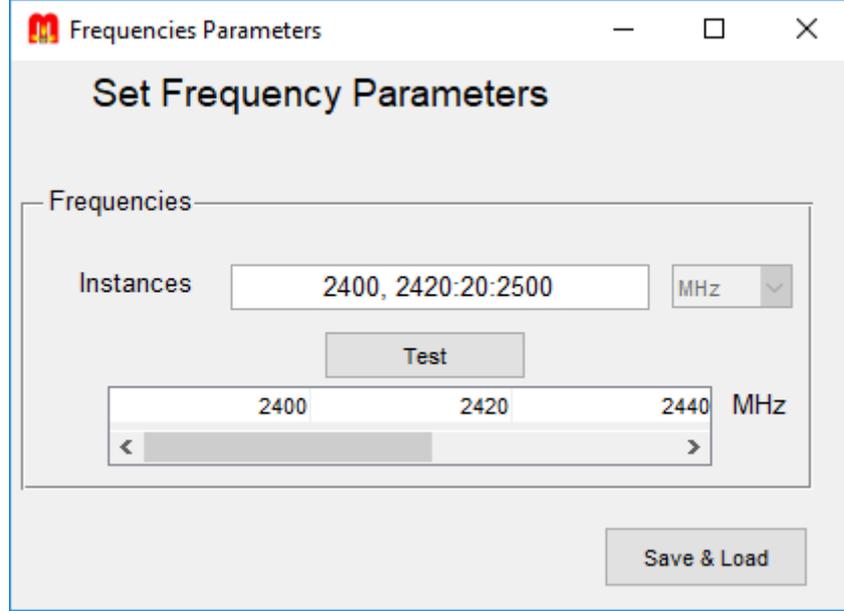


Figure 2.12

we explain in subsection 1.2.1. Of course, this immediately informs us that the transmitted signal in each simulation is by definition considered to be sinusoidal.

To completely define this sinusoidal signal, we need to specify its amplitude, its phase shift and its frequency. In MIMObit this frequency corresponds to one of the entries in the frequency file, while the amplitude and the phase shift of the signal can be derived by the real and imaginary part of the source voltage phasor.

However, considering that the voltage transfer function will depend only on the frequency of the transmitted signal, it turns out that the amplitude and phase shift of the transmitted signal are irrelevant information. Finally, it should be straightforward that, since we are dealing with only one frequency at a time, the voltage transfer function will be calculated only at that specific frequency, which results in a simple fixed complex number.

At this point, we assume that the source voltage is the input to an LTI system with impulse response  $h(t)$ , and that the load voltage is its output. Next, we define the source voltage as  $v_g(t) = A \cdot \cos(2\pi F_c t + \theta)$ , since we have assumed that it is a sinusoidal signal, which gives the following

$$\begin{aligned}
 v_l(t) &= h(t) * v_g(t) \Leftrightarrow \\
 \mathcal{F}\{v_l(t)\} &= \mathcal{F}\{h(t)\} \cdot \mathcal{F}\{A \cdot \cos(2\pi F_c t + \theta)\} \Leftrightarrow \\
 V_l(F) &= H(F) \cdot A \frac{\delta(F + F_c) + \delta(F - F_c)}{2} e^{j2\pi F \frac{\theta}{2\pi F_c}} \Leftrightarrow \\
 V_l(F) &= \frac{A}{2} \left[ H(-F_c) e^{j2\pi(-F_c) \frac{\theta}{2\pi F_c}} \delta(F + F_c) + H(F_c) e^{j2\pi F_c \frac{\theta}{2\pi F_c}} \delta(F - F_c) \right] \overset{v_l(t) \text{ is real}}{\Leftrightarrow} \\
 V_l(F) &= \frac{A}{2} \left[ H^*(F_c) e^{-j\theta} \delta(F + F_c) + H(F_c) e^{j\theta} \delta(F - F_c) \right] \quad (2.5)
 \end{aligned}$$

For the baseband equivalent of  $V_l(F)$  (see Definition 1) we get that

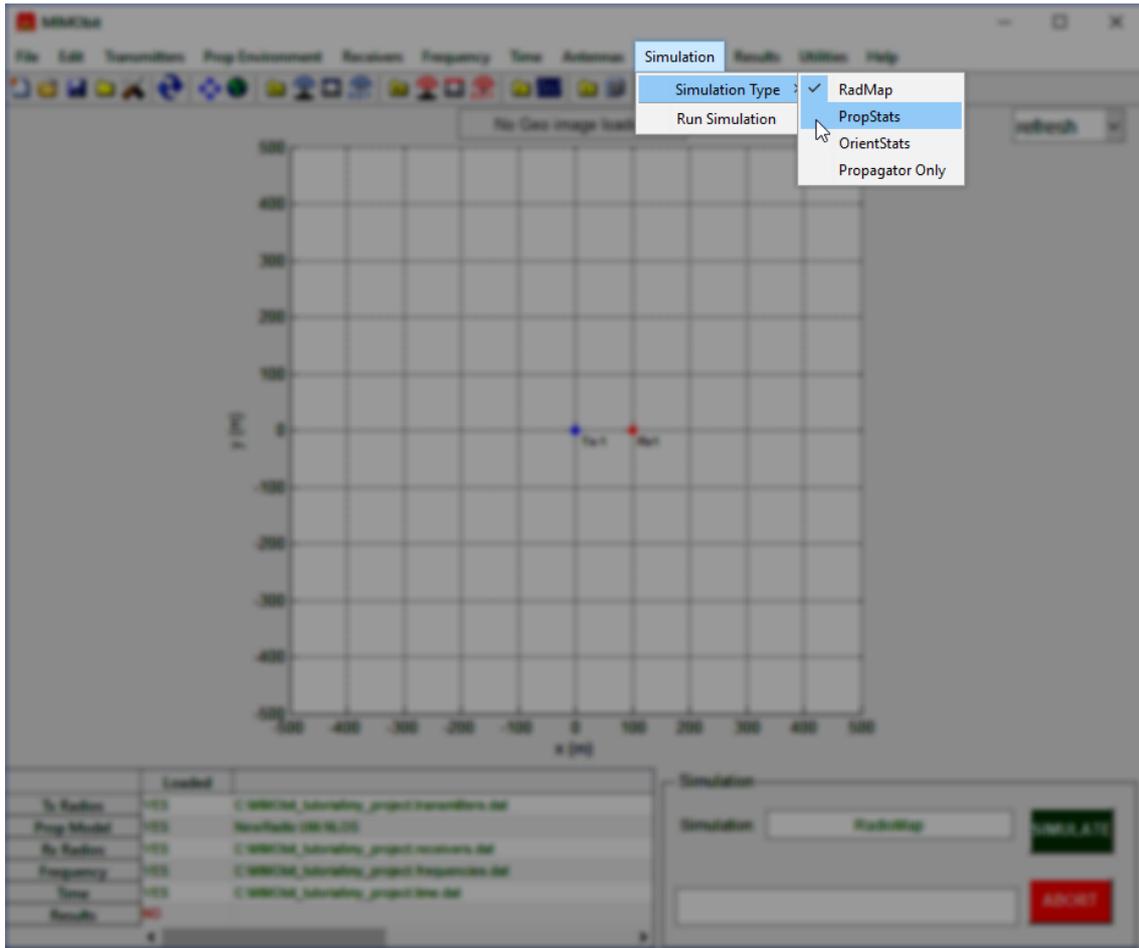


Figure 2.13

$$\begin{aligned}
 V_l^b(F) &= \begin{cases} 2V_l(F + F_c) & F + F_c \geq 0 \\ 0 & F + F_c < 0 \end{cases} \Leftrightarrow \\
 V_l^b(F) &= \begin{cases} A [H^*(F_c)e^{-j\theta}\delta(F + 2F_c) + H(F_c)e^{j\theta}\delta(F)] & F + F_c \geq 0 \\ 0 & F + F_c < 0 \end{cases} \Leftrightarrow \\
 V_l^b(F) &= \begin{cases} AH(F_c)e^{j\theta}\delta(F) & F + F_c \geq 0 \\ 0 & F + F_c < 0 \end{cases} \Leftrightarrow \\
 V_l^b(F) &= AH(F_c)e^{j\theta}\delta(F) \tag{2.6}
 \end{aligned}$$

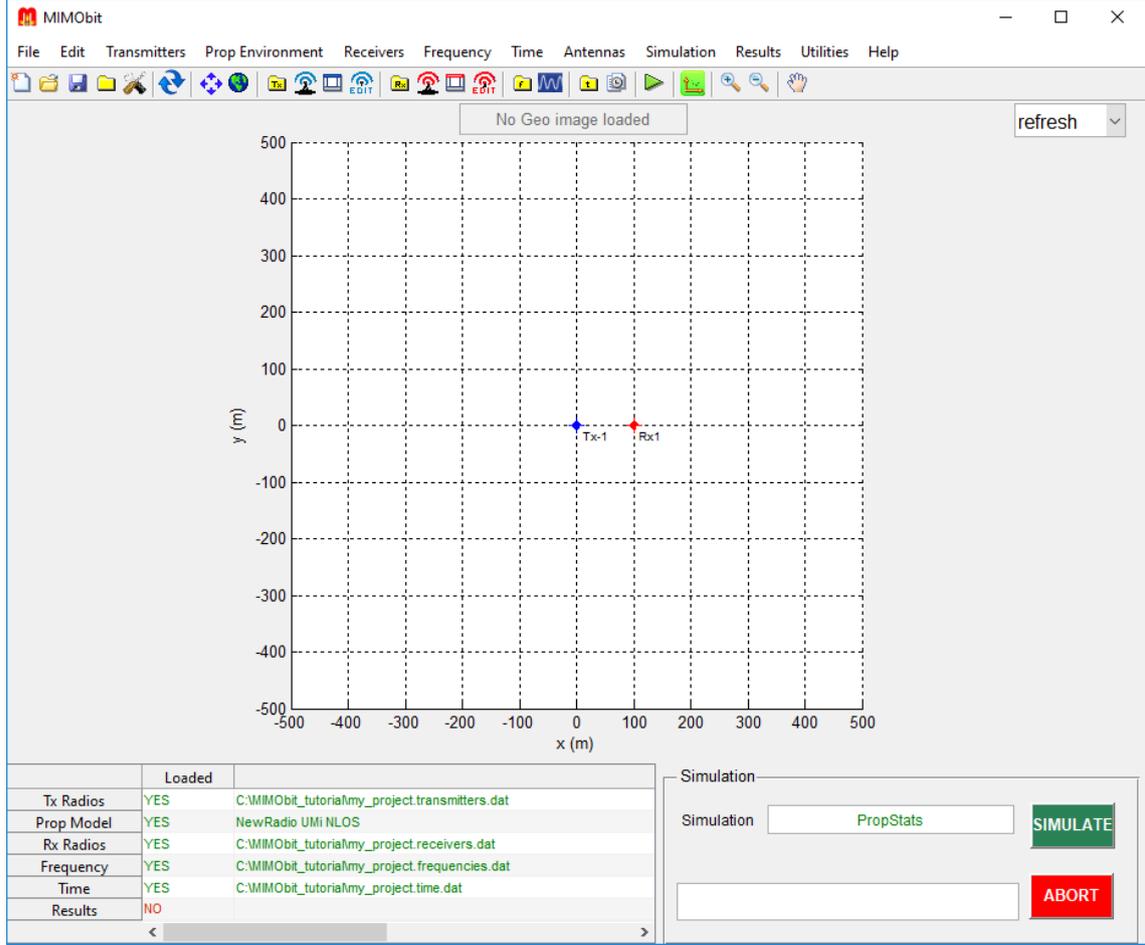


Figure 2.14

Also, for the baseband equivalent of  $V_g(F)$  we have

$$\begin{aligned}
 V_g^b(F) &= \begin{cases} 2V_g(F + F_c) & F + F_c \geq 0 \\ 0 & F + F_c < 0 \end{cases} \Leftrightarrow \\
 V_g^b(F) &= \begin{cases} A [e^{-j\theta} \delta(F + 2F_c) + e^{j\theta} \delta(F)] & F + F_c \geq 0 \\ 0 & F + F_c < 0 \end{cases} \Leftrightarrow \\
 V_g^b(F) &= \begin{cases} Ae^{j\theta} \delta(F) & F + F_c \geq 0 \\ 0 & F + F_c < 0 \end{cases} \Leftrightarrow \\
 V_g^b(F) &= Ae^{j\theta} \delta(F) \tag{2.7}
 \end{aligned}$$

Comparing equations (2.6) and (2.7), we can see that

$$V_l^b(F) = H(F_c) \cdot V_g^b(F)$$

And thus, we have that

$$\begin{aligned}
 \mathcal{F}^{-1} \{V_l^b(F)\} &= H(F_c) \cdot \mathcal{F}^{-1} \{V_g^b(F)\} \Leftrightarrow \\
 v_l^b(t) &= H(F_c) \cdot v_g^b(t) \tag{2.8}
 \end{aligned}$$

Inspecting the definition of a phasor  $\mathbf{V}$

$$v(t) = \Re\{\mathbf{V} \cdot e^{j2\pi F_c t}\} \tag{2.9}$$

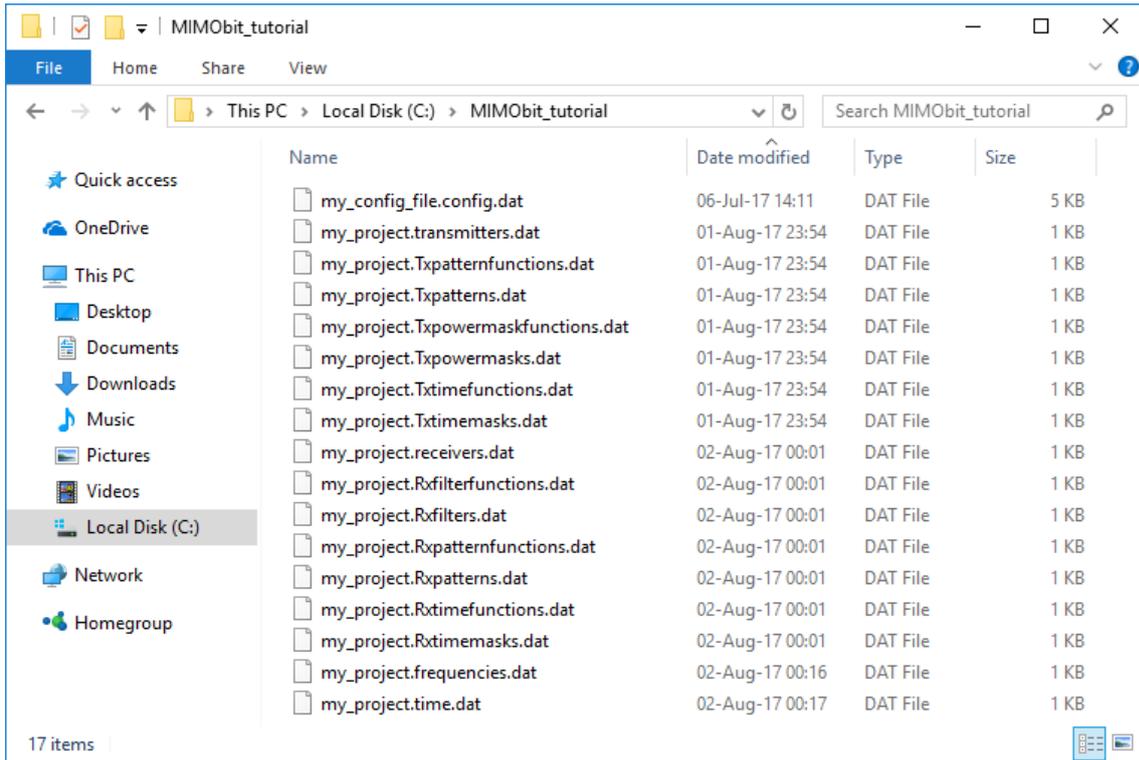


Figure 2.15

we can see that it is very similar to the definition of the baseband equivalent (see Corollary 1.1) of a signal

$$v(t) = \Re\{u^b(t) \cdot e^{j2\pi F_c t}\} \quad (2.10)$$

By direct comparison of (2.9) and (2.10) we can now easily infer that

$$\mathbf{V} = u^b(t)$$

This essentially tells us that the phasor representation of a (sinusoidal) voltage is also the baseband equivalent of this voltage when it is seen as a signal.

Hence, from (2.8) it holds that

$$\mathbf{V}_l = H(F_c) \cdot \mathbf{V}_g \quad (2.11)$$

Comparing this to (2.4) it becomes clear that  $G(F_c) = H(F_c)$ . Therefore, we conclude that the transfer function that relates the source and load voltage phasors at frequency  $F_c$  is exactly the communication system channel frequency response at frequency  $F_c$  as well! This is very important, as it informs us that *the software provides us with the values of the channel frequency response* at the points specified in the frequency file.

### 2.2.2 Transmitters

Taking a look at the "Set Transmitter Parameters" environment (Figure 2.8), we can easily identify five distinct categories of parameters that should be specified in order to create all the required transmitter files. In particular, we have the following categories

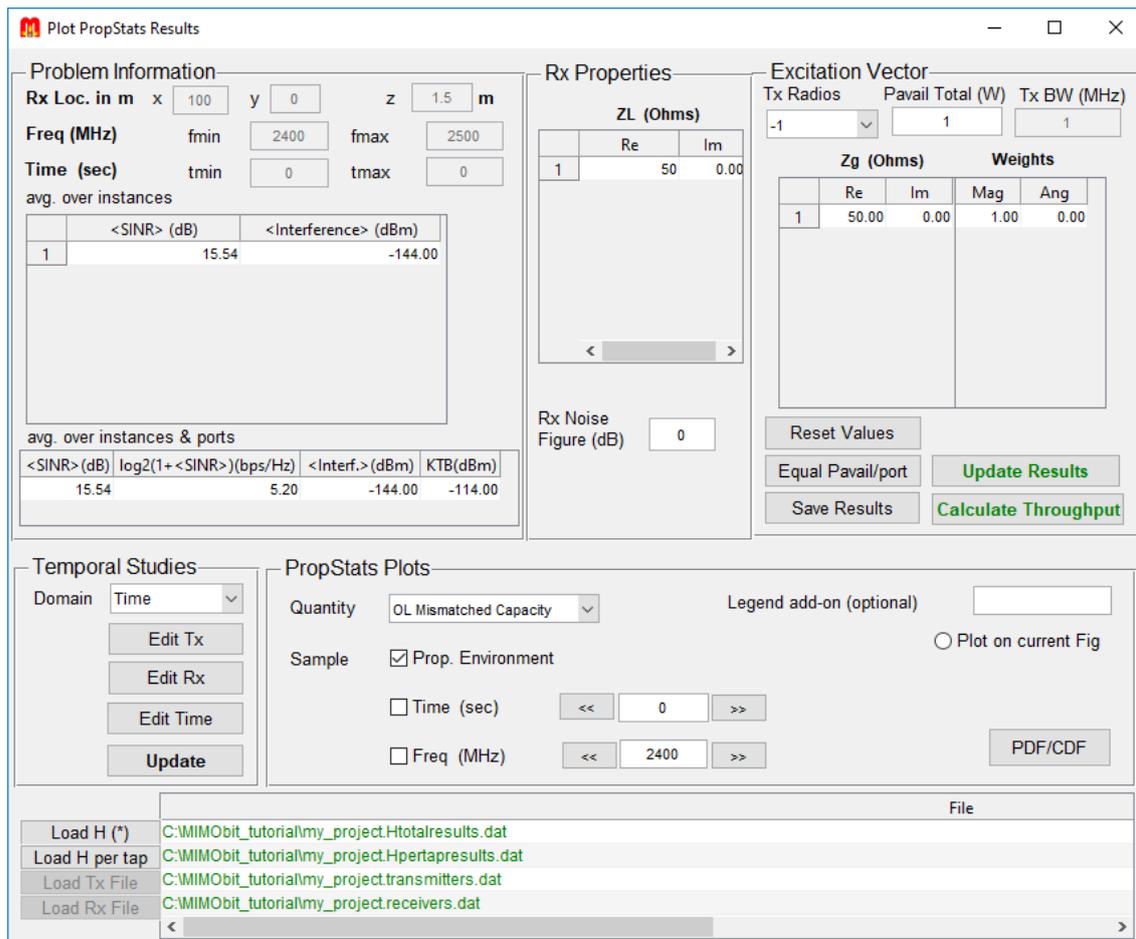


Figure 2.16

## Type

A transmitter can either be a signal transmitter or an interferer. For exact definitions of these transmitter types, please refer to the software manual.

## Tx Radio id

The ids of signal transmitter are positive integers, whereas the ids of the interferers are negative integers.

## Coordinates

The x, y and z coordinates specify the location of the transmitter in space. The units of the coordinates are meters by default, although this can be altered by editing the configuration file (see Figure 2.6).

## Power

In the “Power” group the user specifies the signal that will be transmitted by each transmitter. This is done in the frequency domain by specifying the “Total Available Power”, which is then distributed over the signal bandwidth “BW” as dictated by the “Power Mask”. The “Power Mask” is in some sense similar to the Power Spectral

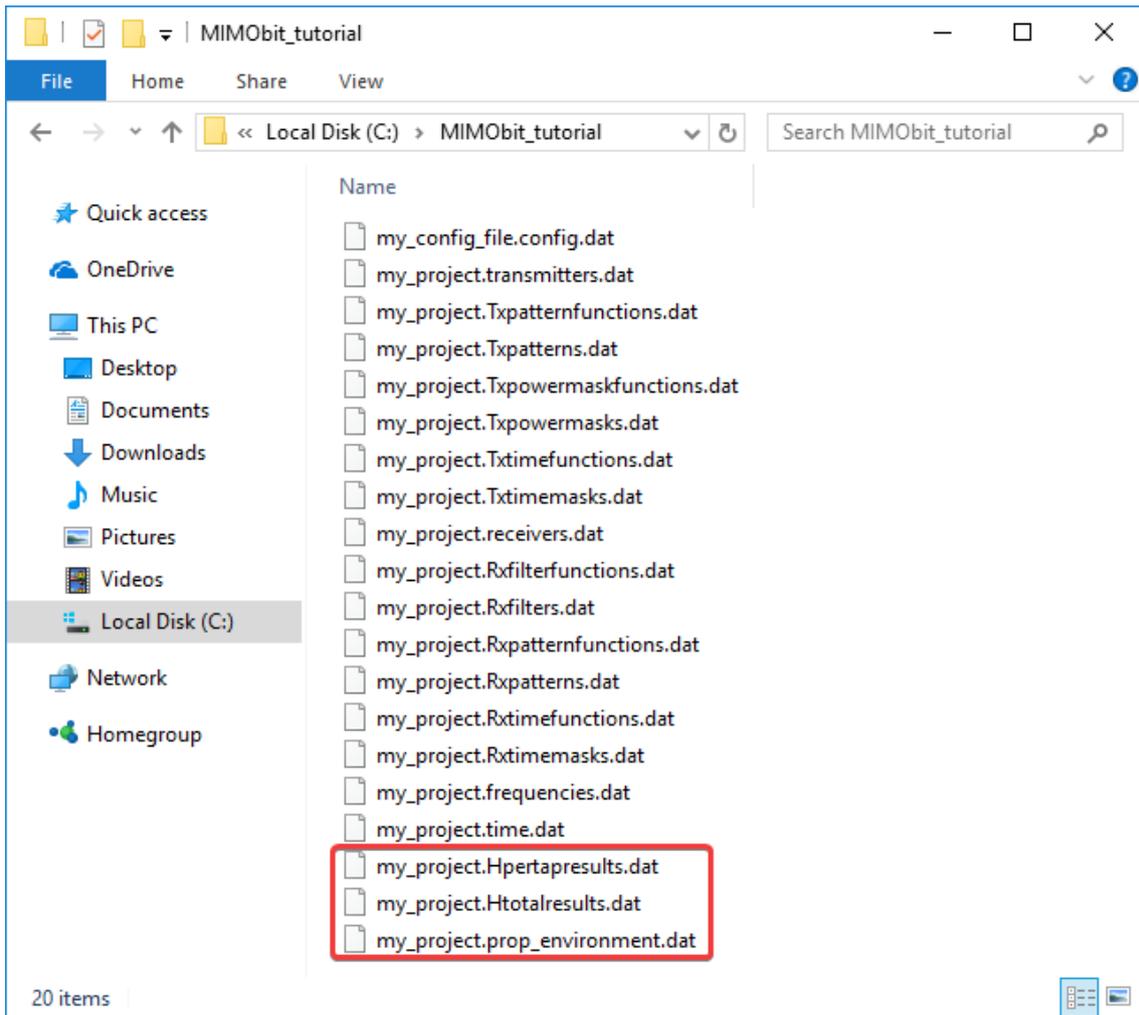
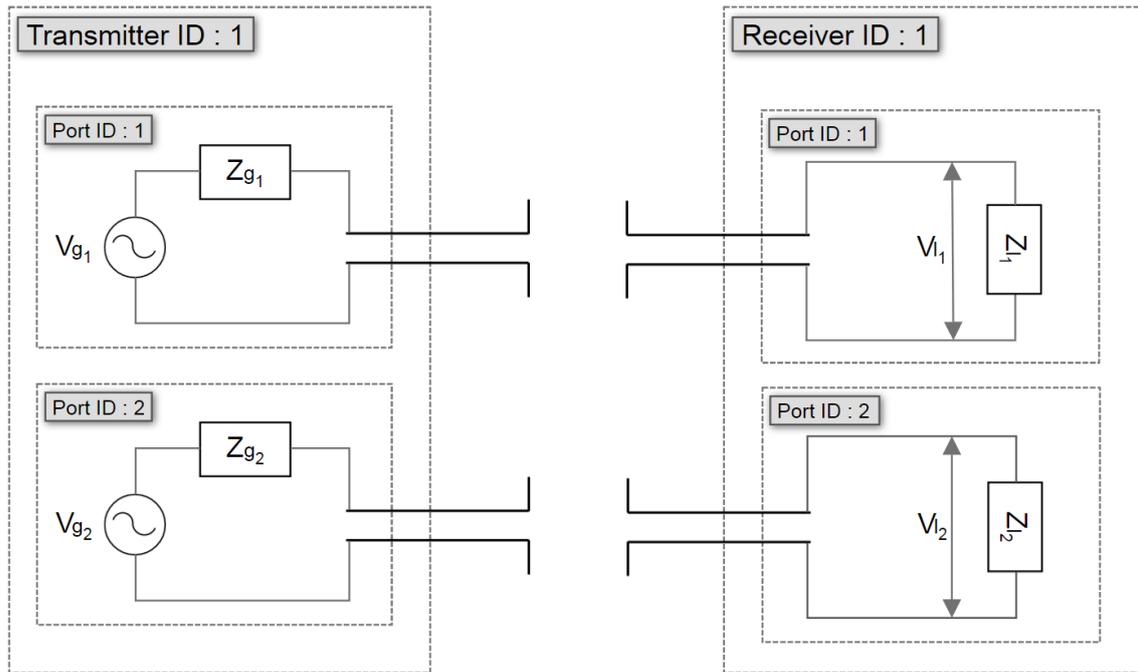


Figure 2.17

Density (PSD) of the signal since it determines the relative differences in power allocation among the various frequencies. However, it differs from a PSD in that it can have non zero values even outside the signal bandwidth and that it has units of dB rather than W/Hz. The “Gmax Frequency” is the frequency at which the maximum of the Power Mask occurs; if there are multiple frequency points at which the maximum is attained, only the first, i.e. lowest, frequency appears. Last, the “Custom Function” field seems to be inaccessible, and it is suspected that is part of redundant code of older versions of the software that has not yet been discarded.

There are three ways to define the “Power Mask” of a transmitter:

- The most convenient option is to select a predefined mask from the “Power Mask” drop down menu. Currently the available options are the “Flat Frequency” mask and a range of approximations for the IEEE 802.11b standard (aka Wi-Fi). In the case that the “Flat Frequency” option is selected, then the “Total Available Power” will be distributed uniformly over the signal bandwidth, “BW”.
- The second option is to select the “Define Power Mask function...” option from the “Power Mask” drop down menu. In this case the window in Figure 2.19 shows up, in which we can design our own power mask using standard Matlab



**Figure 2.18:** Structure of radios in MIMObit. In this example we can see one transmitter and one receiver, each with two ports.

notation. Next, we can click the “Test your function” button in order to obtain a graphical representation of our power mask. Finally, click “OK” to save the Power Mask.

- Another way to design Power Masks and can be accessed by clicking on the “Tx Filter Customisation..” option at the “Transmitters” menu (see Figure 2.7), which opens a new window as shown in Figure 2.20. In this environment we have the ability to design piece-wise linear definitions for the Power Mask, by specifying various points at different frequencies.

For more detailed technical information on the Power Mask design tools, please refer to the software manual.

One might wonder at this point and ask, how does it make sense to define a signal with a nonzero bandwidth, when we know that all the signals involved in the simulations are considered to be sinusoidal?

This inconsistency can in fact be easily explained by first understanding the way that the software treats the channels. As has already been discussed, during a simulation MIMObit computes the values of various channels at the frequencies specified in the frequency file. After the simulation completes, these channels are utilized in order to calculate quantities such as the channel Capacity. To make these calculations, MIMObit assumes that the channel is constant over the signal bandwidth for each transmit antenna. The signal bandwidth is usually centered at each simulation frequency, and the value of the constant channel is considered to be equal to its value at that frequency. For example, if we have defined a signal bandwidth of 20MHz, and the value of the channel calculated at 2400MHz is equal

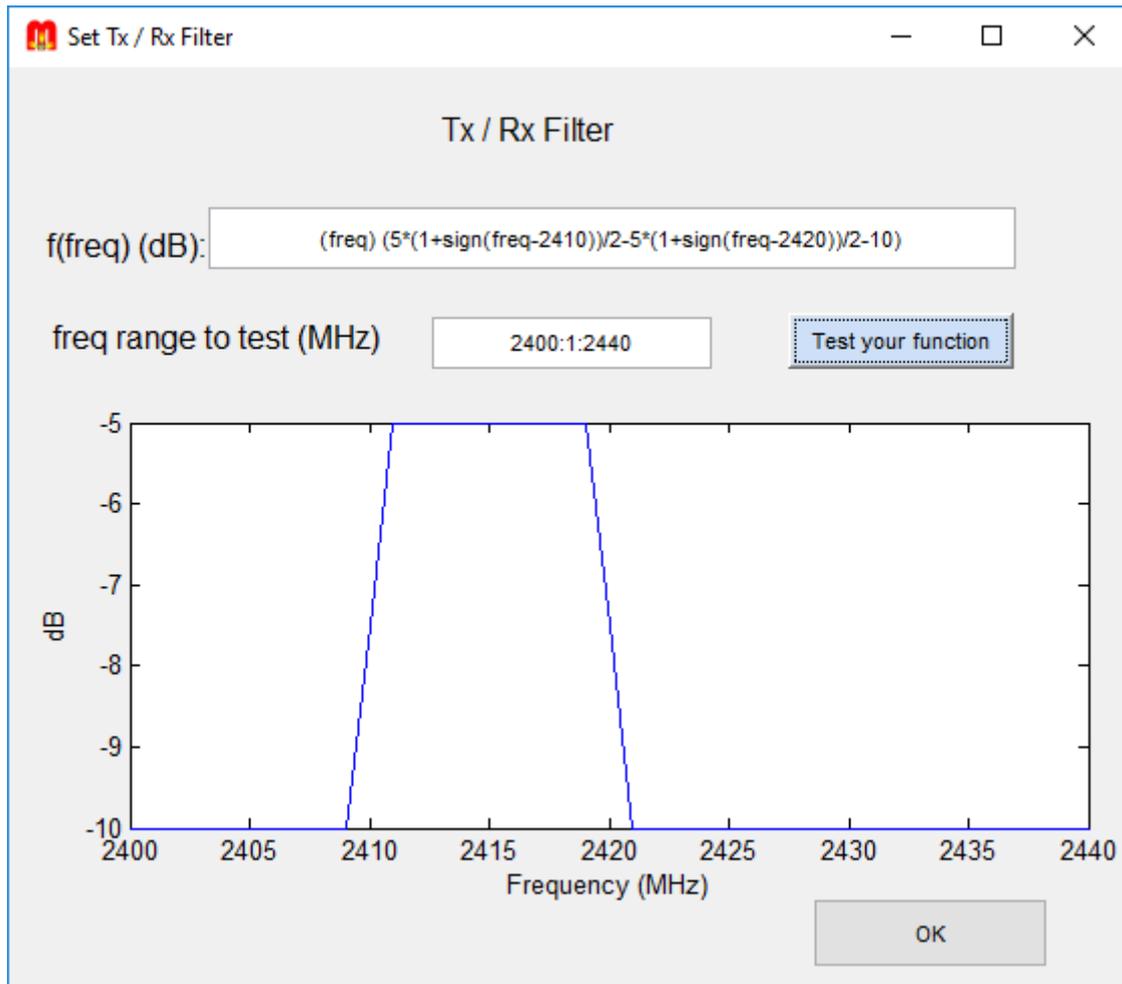


Figure 2.19

to  $0.7 + 0.3i$ , then we have the following definition for the channel:

$$H(F) = \begin{cases} 0.7 + 0.3i & 2390\text{MHz} \leq F \leq 2410\text{MHz} \\ 0 & \text{otherwise} \end{cases} \quad (2.12)$$

*Note that it is not certain whether the signal bandwidth is always centered at the simulation frequency when a Power Mask other than the “Frequency Flat” is loaded. For disambiguation, please contact the software developers.*

## Pattern

In this group we can set all the necessary parameters regarding a transmitting antenna.

- The “Pattern id” field refers to the radiation pattern for which there exist three ways to define it using the “Pattern id” drop down menu.
  - The first option is to select one of the predefined antennas, which are the following
    - \* z-oriented Herzian Dipole
    - \* z-oriented Magnetic Dipole

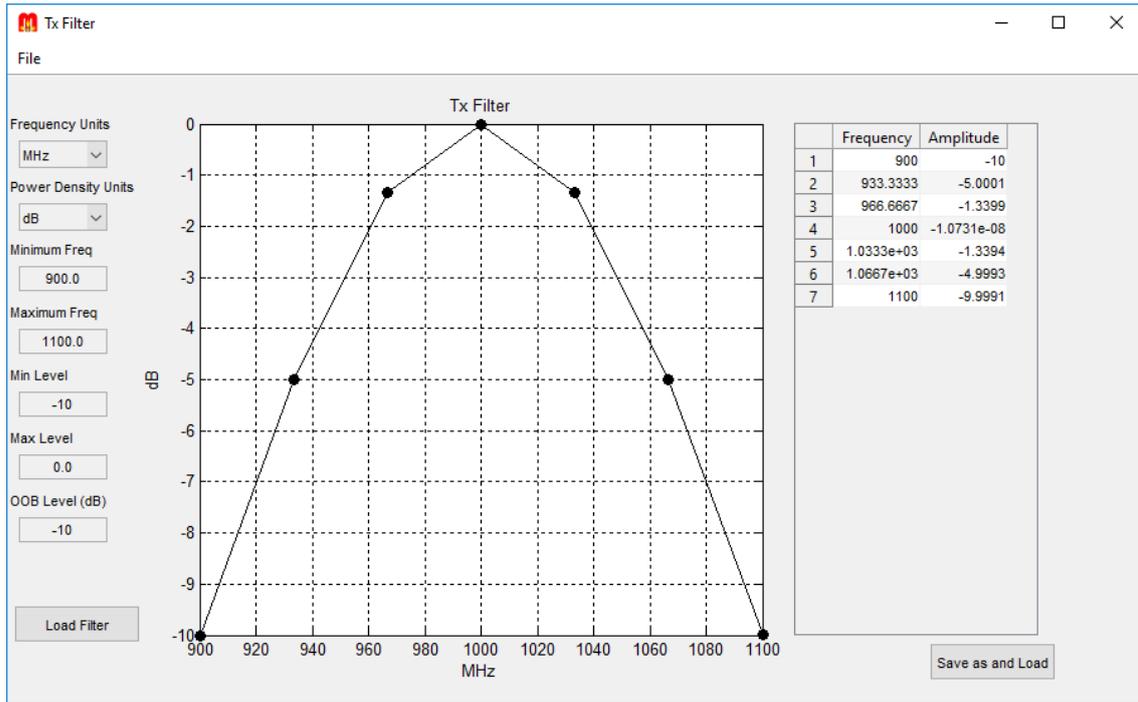


Figure 2.20

\* z-oriented Half-Wavelength Dipole

It is noted that for these antennas there exist closed form formulas which enable us to accelerate dramatically the simulation times. For example, it is preferred to choose the Half-Wavelength Dipole, which has a length equal to half the wavelength at each simulation frequency, rather than design and load a custom dipole of the same length.

- We can also utilize a custom antenna design by selecting the “Load Pattern...” option and then loading the respective \*.lab file. For more information on custom antennas, please refer to subsection 2.2.5.
  - We can also select the “Define Pattern...” or “Define Pattern function..” option. For further information on these features, please refer to the software manual.
- The “Euler phi”, “Euler theta” and “Euler psi” fields specify the angle at which we might need to rotate the antenna for certain telecommunication studies.
  - The “Ant. Zin” parameter is the input impedance of the antenna/
  - The “Zg” parameter is the source impedance.
  - The “Radiation Efficiency” parameter probably refers to the the matched radiation efficiency of the antenna as defined in subsection 1.2.2. For disambiguation please contact the software developers.

### Temporal

This group refers to the behavior that the antenna will have in the time domain. In essence, we define a function that will be multiplied by the transmitted signal

in the time domain, and by doing this we can switch on or off the antenna, or just attenuate or boost the transmitted signal. This way we can run simulations at certain discrete points in time, and thus observe how the communication system behaves in these time instances. The concept of time may be a bit misleading at first, since for example we know that the propagation environments remain always static throughout the simulation (see page 25).

For further technical information on this feature, please refer to the software manual.

### 2.2.3 Receivers

As can be seen in Figure 2.21, the definition of a receiver is almost identical to that of a transmitter differing only in the following:

- There is only a single “Type” of receiver, and its id is always a positive integer.
- In addition to the transmitters predefined antennas at the “Pattern id” drop down menu, receivers also include an “Ideal Probe” option.
- Instead of a transmitter source impedance, “Zg”, in the case of a receiver we have of course a “Load Impedance”.
- The “Noise Figure” parameter is the noise figure of the receiving system.
- For the exact definition of the “Gain” parameter, please contact the software developers.
- The “RBW” field refers to the receiver resolution bandwidth. For the exact definition of the “RBW” parameter, please contact the software developers.

The screenshot shows the 'Set Receiver Parameters' dialog box. It is divided into several sections:

- Coordinates:** x: 0.0 m, y: 0.0 m, z: 1.5 m.
- Receiver:** Rx Filter: Frequency Independ..., Custom Function: [empty], Gain: 0.0 dB, Noise Figure: 0.0 dB, RBW: 1 MHz.
- Temporal:** Mode: [dropdown], Offset: 0.0 sec, Custom Function: [empty].
- Radiation Pattern:** Pattern ID: Ideal Probe, Custom Function: [empty], Ant. Input Impedance: 50 Ohms, Load Impedance: 50.0 Ohms, Radiation Efficiency: 100%, Euler phi: 0.0 deg, Euler theta: 0.0 deg, Euler psi: 0.0 deg.

Buttons at the bottom include 'Get from the Graph', 'Save Receiver', 'Save Only', and 'Save & Load'.

Figure 2.21

## 2.2.4 Apps & Simulation Files

There are currently three simulation types in MIMObit, which are the *RadMap*, the *PropStats* and the *OrientStats* Apps (see Figure 2.13).

At the end of each simulation three new files are generated as shown in Figure 2.17, and these files might have slightly different structure each time a different App is selected. Immediately, a new window appears, different for each App, where we can access all the final simulation data that MIMObit has to offer.

There is also the *Propagator Only* App, which has a little different functionality from the aforementioned Apps.

### PropStats

The main task of PropStats (see Figure 2.16) is to calculate approximations of the Probability Distribution Function (PDF) and the Cumulative Distribution Function (CDF) of the capacity or the throughput over many propagation environment instantiations for a specific receiver. If more than one receivers exist on the map, MIMObit will prompt us to choose which one of them we would like to calculate the aforementioned quantities for. The various groups in the PropStats App window are discussed below

- *PropStats Plots*

Quantity	MIMObit currently offers throughput and three types of capacity calculation (for more information refer to the software manual).
Sample	This parameter specifies over which channels the PDF and the CDF will be calculated. By default, the “Prop. Environment” checkbox is selected, which cannot be deselected, whereas the “Time” and “Freq” are deselected. This way MIMObit will calculate the PDF and CDF over all the propagation environment instantiations at the specific “Freq” and “Time” values. If the “Freq” or “Time” checkbox is also selected, then channels at all frequencies or times are also taken into account, respectively.
Plot on current Fig	When selected we can plot multiple PDFs and CDFs in the same figure.
Legend add-on	We can add a legend on the figure so as to distinguish multiple graphs.
PDF/CDF	Clicking on this button creates the PDF and CDF figures.

- *Problem Information*

$\log_2(1+\langle \text{SINR} \rangle)$  Averaged channel capacity (not certain which samples are averaged, please contact the software developers)

KTB This parameter refers to the thermal noise power (see subsection 1.2.2) and is equal to the Boltzmann constant multiplied by the “Temperature” parameter in the configuration file (see Figure 2.6) multiplied by the “RBW” parameter of the receiver (see Figure 2.21).

$\langle \text{Interference} \rangle$  Although, the meaning of this parameter is straightforward, it is important to mention that sometimes it takes different value from what we would expect. Specifically, in situations where there is little to no interference, then it is set equal to “P2KTB\_FACTOR”\*KTB, with “P2KTB\_FACTOR” having a default value of 0.001. In fact, any signal or interference that is detected to be less than “P2KTB\_FACTOR”\*KTB is artificially set equal to “P2KTB\_FACTOR”\*KTB. This is mostly for visualization purposes and the good behavior of some plots under very low power circumstances, since power is plotted in dBm. “P2KTB\_FACTOR” is a hidden configuration file parameter (see page 22).

- *Rx Properties*

ZL The load impedance. It corresponds to the “ZL” parameter of the receiver (see Figure 2.21).

Rx Noise Figure It corresponds to the “Rx Noise Figure” parameter of the receiver (see Figure 2.21). In essence, when its value is non zero, then it just decrements  $\langle \text{SINR} \rangle$  by this value.

- *Excitation Vector*

Tx Radios	From this dropdown menu we can select the “Tx Radio id” (see Figure 2.8) of the transmitter for which we would like to inspect or alter some of its parameters.
Pavail Total	It corresponds to the “Total Available Power” (see Figure 2.8) parameter of the selected transmitter.
Tx BW	It corresponds to the “BW” (see Figure 2.8) parameter of the selected transmitter.
Zg	It corresponds to the “Zg” (see Figure 2.8) parameter of the respective port of the selected transmitter.
Weights	It corresponds to the source voltage of the respective port of the selected transmitter. We should note that there is a good reason that this parameter has been named “Weights” instead of “Source Voltages” or “Vg”, and it is discussed on page 49.
Reset Values	It resets all the altered parameters, not only those pertaining to the transmitters, to their original state.
Equal Pavail/port	It automatically calculates and sets the values of the “Weights”, so that the “Pavail Total” is distributed uniformly across all the ports of the selected transmitter. This is mainly useful when we have set different source impedances for some or all of the ports. For further elaboration, see page 49.

Update Results	It quickly recalculates all the PropStats capacities, taking into account any changes that we might have done to any parameters from inside the PropStats environment. This feature is useful when we need to study the behavior of the system when certain parameters are altered, without having to start a new simulation all over again. Another benefit is that these calculations are performed on the existing channels, whereas a new simulation would produce completely new propagation environment instantiations, in the case of a non deterministic propagation environment, and thereby completely new channels. It is noted, however, that only the parameters that are included in the PropStats environment are allowed to change, and thus if we attempt to reconfigure other variables by other means (for example by manually editing a file), then MIMObit will very likely provide us with faulty results.
Calculate Throughput	Throughput results are not calculated automatically and this button needs to be clicked on in order to produce them. The <i>*.Hpertapresults.dat</i> file is necessary for this calculation.
Save Results	It creates files containing the simulation results for all the capacity types and the throughput, which can then be used for independent post processing. To export throughput results, we need first to click on the “Calculate Throughput” button. Otherwise, the file that corresponds to the throughput is going to be empty.

### **OrientStats**

This App is almost identical to the PropStats App, differing only in that it repeats the PropStats simulations for various rotations of the receiver. These rotations are specified by the “phi”, “theta” and “psi” angles in the configuration file (see Figure 2.6).

For further information, please refer to the software manual.

### **RadMap**

RadMap is devoted to analyzing the behavior of our communication system spatially. In contrast to PropStats and OrientStats, multiple receivers are allowed in this App, and only one propagation environment instantiation will be generated, thereby obtaining a single channel for each pair of transmitter-receiver ports and for each frequency specified in the frequency file.

We note that even if the value of the “Sample size” in the configuration file (Figure 2.6) is set greater than one, only one propagation environment instantiation will still be generated.

For more information, please refer to the software manual.

### Simulation Files

After each simulation, three new files will be generated containing important information that will be used for post processing tasks in the Apps above, in order to obtain desired quantities such as Capacity.

In particular, the *\*.Htotalresults.dat* file contains information pertaining to the channels, the *\*.prop\_environment.dat* file contains information pertaining to the various taps that were created for the simulation of the selected propagation environment, and the *\*.Hpertapresults.dat* file contains information regarding the channels as in *\*.Htotalresults.dat* file, but for every tap individually.

As an example, we can take a look at the *my-project.Htotalresults.dat* file that was generated after the simulation example in section 2.1 (see Figure 2.22), where we can clearly see that there is one channel calculated for every combination of the following

- 2 propagation environment instantiations
- 6 frequencies specified in the *my-project.frequencies.dat* file
- 1 port of the transmitter

It should be mentioned that the file that contains the channels for each individual tap is not generated by default, and thus the *\*.Hpertapresults.dat* file will always be empty. To enable the generation of these channels during a simulation, we need to edit the configuration file (Figure 2.5) and set the “Calculate H per tap” option in the “Throughput” menu to “YES”.

In the case of PropStats and OrientStats, where more than one propagation environment instantiations may be generated, the produced *\*.prop\_environment.dat* file corresponds only to the last of these instantiations.

In Table 2.6 we can find a list with parameters that are known to NOT be involved in the simulation process of the channels, and thus channels will not change when we alter the values of these parameters. Of course, in non deterministic propagation environments, the channels are always going to be different, but this is due to the probabilistic nature of the propagation environment instantiations, rather than the change in the values of these parameters.

### Propagator Only

This App was designed to enable us to bypass all the simulation stages of the previous Apps, except for the stage where the *\*.prop\_environment.dat* file is generated. This is very useful when we need to study the behavior of our communication system under different set ups, but where the propagation environment stays always the same.

While utilizing non deterministic propagation environments, MIMObit will always generate different instantiations every time we run a new simulation. In order

Location	Parameter
Configuration file	“Temperature”
Transmitter	“Power Mask” “Total Available Power” “Gmax Frequency” “BW” “Mode” “Offset”
Receiver	“Rx Filter” “Gain” “Rx Noise Figure” “RBW” “Mode” “Offset”

Table 2.6

to cope with this issue, MIMObit includes the “Import Prop Model...” feature (see Figure 2.10), which loads a specific *\*.prop\_environment.dat* file, and thus all our simulations will be based on the single propagation environment instantiation that is described in this file.

In summary, we first select the desired propagation environment, then we select the “Propagator Only” App, after that we click on the “Simulation” button in order to produce the *\*.prop\_environment.dat* file which corresponds to a single instantiation of this environment, and finally we utilize the “Import Prop Model...” feature in order to load this file in MIMObit. Of course, all the necessary files pertaining to the transmitters, the receivers, the frequencies and the time instants, have to be created and loaded as well before we can run the simulation.

### 2.2.5 Custom Antennas

MIMObit provides the ability to create custom dipole arrays, which can then be set to be a transmitter or receiver radio. This feature is really useful when we need to study the behavior of our communication system when real physical antennas are utilized.

To create a dipole array we click on the “**Dipole Array...**” option at the “Antennas” menu. A window appears next as shown in Figure 2.23, where we can click on the “New Dipole Array” button that opens a second window, as seen in Figure 2.24, in which we can set all the parameters of the dipole array.

Similarly to the transmitters and receivers set up process, we first need to enter the “Number of Objects” which represents the total number of the antennas that our dipole array will contain. Afterwards, we set all the the parameters to the desired values and click on the “Save Object” button. If more than one “Number of Objects” was specified, we can see at this point that the “Object id” has been incremented by one, which informs us that we are ready to set up the next antenna of the dipole array. After we repeat this process for all the antennas of our dipole

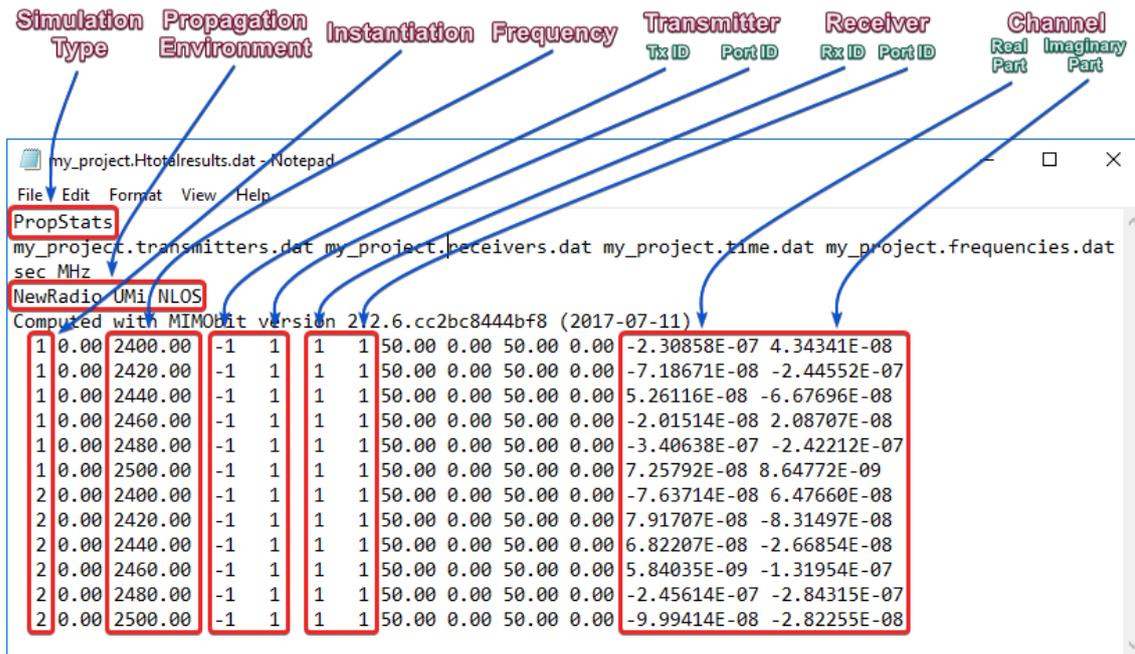


Figure 2.22

array, we finally click on the “Save File” button to store all the parameters to a file.

Returning to the window in Figure 2.23, we observe that the option “**Visualize Geometry**” has been enabled. This option provides a graphical and intuitive three dimensional representation of the dipole array that we have designed, and makes it easy to validate the geometric settings of our design.

However, we are not ready yet to load our dipole array to a transmitter or a receiver, since all we have accomplished so far is only to store the parameters in a *\*.MoMdipolesinfo.dat* file. It is important, thus, to click on the “Simulate” button in order to produce the final input that can be loaded in a radio. When the simulation completes, the “**Visualize LAB file**” option is activated, which opens a new environment where we can inspect a number of properties of our simulated dipole array; this environment can also be accessed by the “**Plot Lab file...**” option in the “Antennas” menu.

The necessary file to be loaded in transmitter or receiver has now been created and it can be identified by its *\*.lab* extension (Figure 2.25); a *\*.slp* file has also been created.

Regarding the parameters of our dipole array, although most of them are self explanatory, there some interesting remarks that should be made.

- First of all, the source voltage and the source impedance correspond to the “Voltage” and “Impedance” parameters in the “Properties” group, respectively (Figure 2.24).
- The frequencies can be set using standard Matlab notation. If we click on the “Select FF Frequencies” button, a list with all these frequencies appears where undesirable values can be deselected. Of course, this can be accomplished directly during the first step.
- The “Object id” of each port of a radio is always a positive integer.

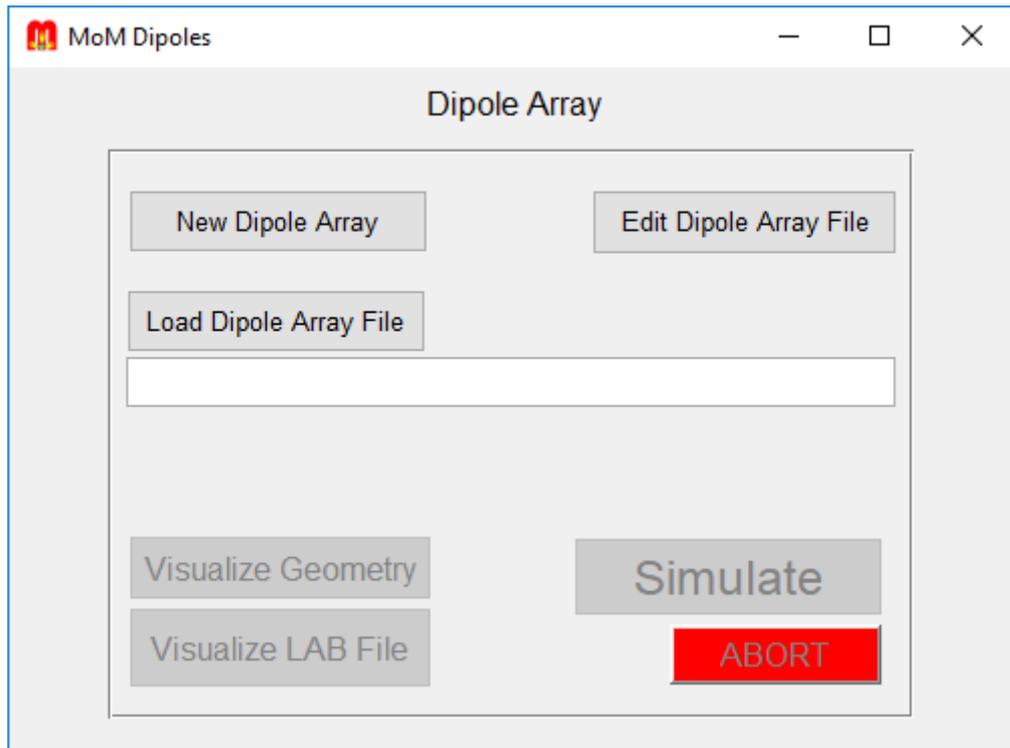


Figure 2.23

- It is known that in general the behavior of an antenna is in fact a function of the frequency. Thus, we should be careful to simulate our antenna at all the frequencies that we intend to run the simulations of our communication system at. Specifically, the frequencies in the dipole array “General” group, should match exactly the values stored in our frequency file (\*.frequencies.dat). In particular, if we attempt to run a simulation at a frequency for which we do not have the calculated behavior of a dipole array which is loaded in a transmitter or receiver, then MIMObit by default will proceed by utilizing the simulated behavior of that dipole array at the nearest available frequency. For example, if we need to run a simulation at 2400MHz, but we load in transmitter or receiver a dipole array that has been simulated only at 2390MHz and 2500MHz, then MIMObit will proceed by utilizing the behavior of the antenna at 2390MHz.

Apart from using the nearest frequency, there is also the option of zero padding (for definition please contact the software developers), and you can choose between the nearest frequency or zero padding options by setting the value of the “INTERPOLATION” variable accordingly. Normally, when MIMObit comes across this, it shows a window that prompts you in essence to select the value of the “INTERPOLATION” variable.

- It is important to comprehend that the “Voltage” parameter in the “Properties Group” (Figure 2.24) does not correspond to the real source voltages by which the port will be excited. To explain this strange discrepancy let us first assume for simplicity that we have a dipole array with only one dipole, and consider

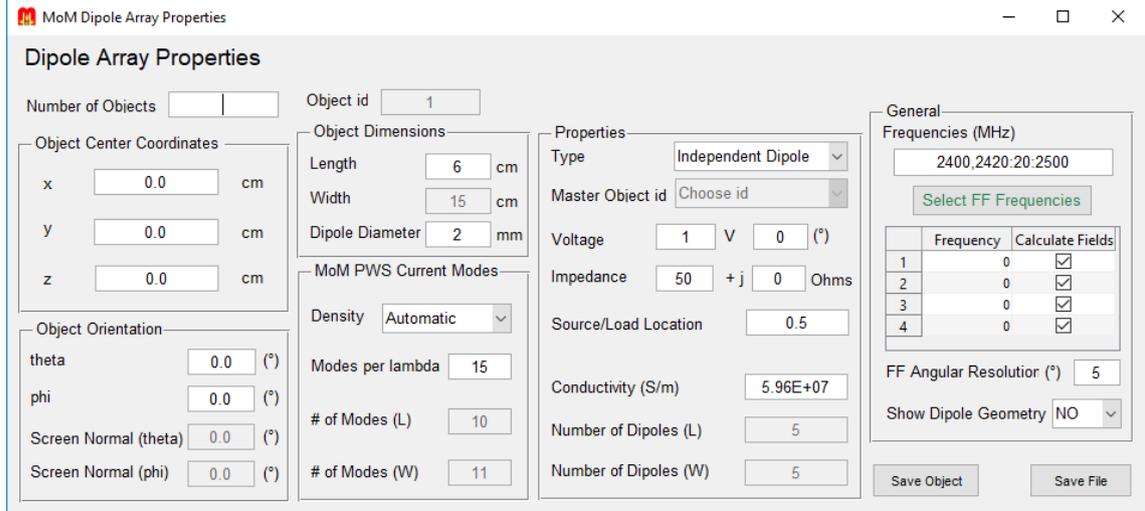


Figure 2.24

again the maximum power transfer formula (see subsection 1.2.2)

$$P_{avail_{tot}} = \frac{|V_g|^2}{8\Re\{Z_g\}} \quad (2.13)$$

This formula relates the total available power,  $P_{avail_{tot}}$ , to the source voltage,  $V_g$ , and the source impedance,  $Z_g$ , and holds for every dipole in our dipole array; the total available power corresponds to the transmitters “Total Available Power” parameters in the “Power” group (Figure 2.8).

It becomes evident at this point that if we fix two of the variables in (2.13) to a specific value, then the third variable cannot take arbitrary values because the equation would be violated. Since MIMObit provides the ability to set the values of all three variables, it will be necessary to cope with the problem. To this end, it automatically scales the magnitude of the source voltage that we have entered, in such a way that (2.13) holds true.

One might wonder now, why would we even be granted the ability to set the magnitude of the source voltage then?

The answer for this lies in our need to have control over the distribution of power across the different dipoles when our dipole array is composed of more than one dipoles. In essence, the voltage magnitude acts as a weight that specifies the portion of the total available power that will be provided to the respective dipole.

To make this clear, consider a dipole array with two ports as in Figure 2.18. We know that in this case the “Total Available Power” is the sum of the available power at each one of the ports, and specifically it will hold that

$$P_{avail_{tot}} = P_{avail_{port1}} + P_{avail_{port2}} \\ P_{avail_{tot}} = \frac{|V_{g1}|^2}{8\Re\{Z_{g1}\}} + \frac{|V_{g2}|^2}{8\Re\{Z_{g2}\}} \quad (2.14)$$

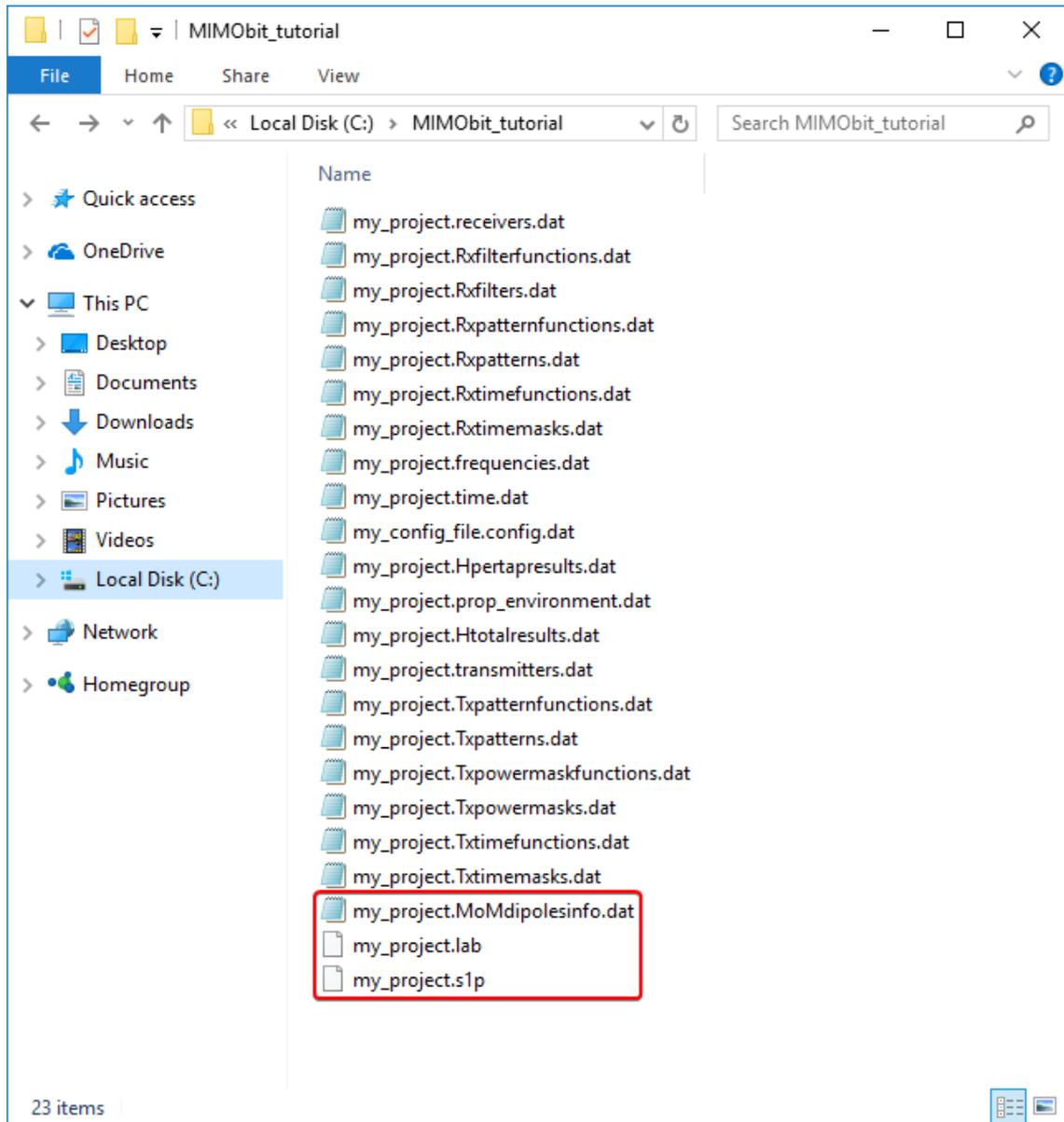


Figure 2.25

The voltages in (2.14) are not the voltages that we have specified, but scaled versions of them by the same factor so that (2.14) holds true.

In other words, what we should only be careful about is the relative differences among the magnitudes of the source voltages of all the dipole array ports.

# Chapter 3

## Channel Capacity

### 3.1 AWGN Channel

#### 3.1.1 Baseband

Consider the following continuous-time AWGN channel with bandwidth  $W$  Hz, average power constraint of  $\bar{P}$  Watts, and white Gaussian noise with a PSD of  $N_0/2$

$$y(t) = x(t) + n(t) \quad (3.1)$$

where  $x(t)$  is the transmitted baseband signal,  $y(t)$  is the received signal, and  $n(t)$  is white Gaussian noise.

Before we continue we make the following observations

- The average power of the useful received signal is equal to the average transmitted signal power, since the channel is ideal.
- The average power of noise is equal to  $N_0/2 \cdot 2W$ . This is true because the PSD is defined as the average of the Fourier transform magnitude squared over a large interval, as

$$\lim_{T \rightarrow \infty} E \left\{ \frac{1}{2T} \left| \int_{-T}^T n(t) e^{-j2\pi ft} dt \right|^2 \right\}$$

which essentially provides us with the average distribution of the noise power in the frequency domain. Hence, since the average power distribution of the white Gaussian noise is constant and equal to  $N_0/2$ , the average power over the desired bandwidth is simply  $N_0/2$  multiplied by this bandwidth,  $2W$ .

We remind that by the *Sampling theorem* we know that any signal  $s(t)$  bandlimited to  $[-W, W]$ , can be described completely by its samples  $s(n/(2W))$  at rate  $2W$ . In particular, by Shannon's interpolation formula, in our case we have that

$$x(t) = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2W}\right) \text{sinc}(2Wt - n) \quad (3.2)$$

What we have essentially achieved with this formula is that the transmitted signal has been represented by the discrete set of symbols  $\{x(n/(2W))\}$ , and thus we can

claim that instead of transmitting the continuous signal,  $x(t)$ , we can equivalently “transmit” this set of symbols. If we can also find a scheme, where the receiver would need to detect this discrete set of symbols instead of estimating the continuous signal, then we would have in fact discovered an equivalent discrete representation of our continuous time model.

To this end, we first slightly manipulate (3.2) as follows

$$x(t) = \sum_{n=-\infty}^{\infty} \frac{1}{\sqrt{2W}} x\left(\frac{n}{2W}\right) \sqrt{2W} \text{sinc}(2Wt - n) \quad (3.3)$$

and thus we have that

$$x(t) = \sum_{n=-\infty}^{\infty} x_n \phi_n(t) \quad (3.4)$$

where

$$x_n = \frac{1}{\sqrt{2W}} x\left(\frac{n}{2W}\right) \quad (3.5)$$

are the modified symbols, and

$$\phi_n(t) = \sqrt{2W} \text{sinc}(2Wt - n) \quad (3.6)$$

form an orthonormal basis, since

$$\int_{-\infty}^{+\infty} \phi_n(t) \phi_m(t) dt = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases} \quad (3.7)$$

As a result, the receiver detects the following signal

$$y(t) = \sum_{n=-\infty}^{\infty} x_n \phi_n(t) + n(t) \quad (3.8)$$

The receiver can convert this continuous signal to a discrete set of symbols as follows

$$\int_{-\infty}^{\infty} y(t) \phi_m(t) dt = \int_{-\infty}^{\infty} \left( \sum_{n=-\infty}^{\infty} x_n \phi_n(t) + n(t) \right) \phi_m(t) dt \quad (3.9)$$

$$\int_{-\infty}^{\infty} y(t) \phi_m(t) dt = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x_n \phi_n(t) \phi_m(t) dt + \int_{-\infty}^{\infty} n(t) \phi_m(t) dt \quad (3.10)$$

$$\int_{-\infty}^{\infty} y(t) \phi_m(t) dt = \sum_{n=-\infty}^{\infty} x_n \int_{-\infty}^{\infty} \phi_n(t) \phi_m(t) dt + \int_{-\infty}^{\infty} n(t) \phi_m(t) dt \quad (3.11)$$

and consequently, when  $n = m$  we have that

$$y_n = x_n + n_n \quad (3.12)$$

where

$$y_n = \int_{-\infty}^{\infty} y(t) \phi_n(t) dt \quad (3.13)$$

$$n_n = \int_{-\infty}^{\infty} n(t) \phi_n(t) dt \quad (3.14)$$

and it can also be proved that

$$x_n = \int_{-\infty}^{+\infty} x(t) \phi_n(t) dt \quad (3.15)$$

To achieve this result in practice, we can use at the transmitter a shaping filter with impulse response  $\phi_0$ , which is in fact a square root raised cosine (SRRC) pulse, and an input signal  $\sum_{n=-\infty}^{\infty} x_n \delta(t - \frac{n}{2W})$ . The receiver will be equipped with the matched filter, which in this case has also an impulse response  $\phi_0$ . Finally, by sampling the signal at the receiver at a rate of  $2W$  symbol/s, we obtain (3.12).

On the other hand, if we attempted to utilize a different square root raised cosine pulse, we know that we would have a transmission rate, of

$$\frac{2W \text{ symbol}}{1+a} s \quad (3.16)$$

which will always be lower than  $2W$ , since the parameter  $a$  is a positive number.

It is very important to keep in mind that every result that we derive from now on, refers specifically to the use of a sinc function as our shaping pulse. When other types of square root raised cosine pulses or completely different types of modulation are employed, then the following calculations may be irrelevant.

This might seem absurd at first, since it is known that different SRRC pulses may have various desirable properties in real life, as opposed to the pulse in our formulation. However, we insist on our idealized formulation using sinc pulses, since our ultimate purpose is to calculate theoretical upper bounds on the transmission rate that we can achieve at a given channel.

We note that the signals in telecommunications usually represent quantities like voltages or electric fields. Hence, it would be convenient to define the instantaneous power of a signal  $s(t)$  as follows

**Definition 4.** Signal Power

$$P(t) = |s(t)|^2 \quad (3.17)$$

This power is directly proportional to the real power in Watts to which our signal is associated with. Consider for example the simple circuit in Figure 3.1, where we have a resistor,  $R$ , connected to a DC voltage source,  $V_g$ . We know that the power dissipated in the resistor will be equal to  $V_g^2/R = 10^2/5 = 20 \text{ Watt}$ . If we now switch back to the abstract world of signals and systems, we can formulate the voltage,  $V_g$ , as a signal which is described by the constant function  $s(t)=10$ ; note that this signal is unitless. By the definition of the power of a signal we get that the instantaneous power of  $s(t)$  is equal to  $|s(t)|^2 = |10|^2 = 100$ ; note that this power is also unitless. Thus, the power of our signal will be numerically directly proportional to the physical power dissipated as a result of the voltage drop,  $V_g$ , across the resistor,  $R$ .

Assume now that we need to replace the resistor with a new  $10\Omega$  resistor.

In this case, the power dissipated in the new resistor will be equal to  $V_g^2/R = 10^2/10 = 10Watt$ , whereas the power of the respective signal will remain at the value of 100.

It should be evident at this point that, even though our abstraction of the voltage seen as a signal is not lossless, it can come in useful, since it eliminates the burden of having to include problem specific information in our formulation, such as the value of  $R$  in our example.



Figure 3.1

Similarly, we can define the total energy,  $E_{tot}$ , of a signal,  $s(t)$ , taking into account that, in the real world, the integral of power over time defines the work performed, which in turn leads us to the following

**Definition 5.** Signal Energy

$$E_{tot} = \int_{-\infty}^{+\infty} |s(t)|^2 dt \quad (3.18)$$

Without getting into further details, we can now derive by the Pythagorean theorem that

$$E_{tot} = \int_{-\infty}^{+\infty} |x(t)|^2 dt = \int_{-\infty}^{+\infty} \left| \sum_{n=-\infty}^{\infty} x_n \phi_n(t) \right|^2 dt = \int_{-\infty}^{+\infty} \sum_{n=-\infty}^{\infty} |x_n \phi_n(t)|^2 dt \quad (3.19)$$

Interchanging the order of the summation and the integration, we conclude that

$$E_{tot} = \sum_{n=-\infty}^{\infty} \int_{-\infty}^{+\infty} |x_n \phi_n(t)|^2 dt \quad (3.20)$$

This formula informs us that the total energy in our signal is in fact the sum of the energies of  $\{x_n \phi_n(t)\}$ . Thus, we can intuitively associate these energies with the respective symbols  $\{x_n\}$  which gives us the following

**Definition 6.** Energy per Symbol

$$E_n = \int_{-\infty}^{+\infty} |s_n \phi_n(t)|^2 dt \quad (3.21)$$

Using this definition, we can also derive that

$$E_n = \int_{-\infty}^{+\infty} |x_n \phi_n(t)|^2 dt = |x_n|^2 \int_{-\infty}^{+\infty} |\phi_n(t)|^2 dt = |x_n|^2 \quad (3.22)$$

by which we can calculate the value of the average energy per symbol as follows

$$E\{x_n x_m\} = E \left\{ \left( \int_{-\infty}^{+\infty} x(t) \phi_n(t) dt \right) \left( \int_{-\infty}^{+\infty} x(\tau) \phi_m(\tau) d\tau \right) \right\} \quad (3.23)$$

$$= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} E\{x(t)x(\tau)\} \phi_n(t) \phi_m(\tau) d\tau dt \quad (3.24)$$

$$= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} R_x(t - \tau) \phi_n(t) \phi_m(\tau) d\tau dt \quad (3.25)$$

$$= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \bar{A} \delta(t - \tau) \phi_n(t) \phi_m(\tau) d\tau dt \quad (3.26)$$

$$= \bar{A} \int_{-\infty}^{+\infty} \phi_n(t) \left( \int_{-\infty}^{+\infty} \delta(t - \tau) \phi_m(\tau) d\tau \right) dt \quad (3.27)$$

$$= \bar{A} \int_{-\infty}^{+\infty} \phi_n(t) \phi_m(t) dt \quad (3.28)$$

$$= \begin{cases} \bar{A} & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases} \quad (3.29)$$

Hence,

$$\bar{E}_{sym} := E\{E_n\} = E\{|x_n|^2\} \quad (3.30)$$

On the calculations above, we have further assumed that the transmitted signal  $x(t)$  is a Wide-Sense Stationary random process and that the values of the process at different times are uncorrelated. These are actually important assumptions, since otherwise we would end up with a different expected value for each symbol, which would have undesirable algebraic implications on our next steps.

**Definition 7.** Wide-Sense Stationarity

A stochastic process  $s(t)$  is Wide-Sense Stationary (WSS) when

- its expected value,  $E\{s(t)\}$ , is constant
- its autocorrelation function,  $E\{s(t_1)s(t_2)\}$ , depends only on  $t_2 - t_1$ . For this reason, we will denote its autocorrelation function as  $R_s(t_2 - t_1)$ .

Even though these assumptions might seem too constrictive, it is common in practice for our signals to, at least approximately, have these characteristics. For

example, we know that in practice the transmitted symbols are usually independent identically distributed random variables, due to coding algorithms that are applied on the data before they are transmitted. In fact, the existence of these coding algorithms are an essential component for the calculation of the desired upper bound of the transmission rate at a given channel.

We are now ready to uncover the relationship between the average energy per symbol,  $\bar{E}_{sym}$ , and the average power constraint,  $\bar{P}$ , of the whole signal. We will denote the (infinite) duration of our signal by  $T_0$ , and we have that

$$\frac{E\{E_{tot}\}}{T_0} \leq \bar{P} \text{ J/s} \stackrel{(3.20)}{\Rightarrow} \quad (3.31)$$

$$\frac{1}{T_0} E \left\{ \sum_{n=-\infty}^{\infty} \int_{-\infty}^{+\infty} |s_n \phi_n(t)|^2 dt \right\} \leq \bar{P} \text{ J/s} \Rightarrow \quad (3.32)$$

$$\frac{1}{T_0} \sum_{n=-\infty}^{\infty} E \left\{ \int_{-\infty}^{+\infty} |s_n \phi_n(t)|^2 dt \right\} \leq \bar{P} \text{ J/s} \Rightarrow \quad (3.33)$$

$$\frac{1}{T_0} (T_0 \text{ s} \cdot 2W \text{ symbol/s}) E \left\{ \int_{-\infty}^{+\infty} |s_n \phi_n(t)|^2 dt \right\} \leq \bar{P} \text{ J/s} \stackrel{(3.22)}{\Rightarrow} \quad (3.34)$$

$$(2W \text{ symbol/s}) \bar{E}_{sym} \leq \bar{P} \text{ J/s} \Rightarrow \quad (3.35)$$

$$\bar{E}_{sym} \leq \bar{P} \text{ J/s} \frac{1}{2W} \text{ %symbol} \Rightarrow \quad (3.36)$$

$$\bar{E}_{sym} \leq \frac{\bar{P}}{2W} \text{ J/symbol} \quad (3.37)$$

Finally, the noise samples,  $\{n_n\}$ , are actually jointly Gaussian random variables, since, loosely speaking, they are “linear combinations” of infinite Gaussian random variables (see (3.14)).

Their variance can be derived immediately from the result in (3.29), and is equal to  $N_0/2$ . This is true, because it is known that its autocorrelation function will be the inverse Fourier Transform of its PSD, and thus it holds that  $R_n(t_2-t_1) = \frac{N_0}{2} \delta(t_2-t_1)$ . Apart from the variance, (3.29) also reveals that the noise samples are uncorrelated as well, and since they are jointly Gaussian, they should be independent too.

For their expected value, we can calculate that

$$E\{n_n\} = E \left\{ \int_{-\infty}^{\infty} n(t) \phi_n(t) dt \right\} = \int_{-\infty}^{\infty} E\{n(t)\} \phi_n(t) dt = \int_{-\infty}^{\infty} 0 \phi_n(t) dt = 0 \quad (3.38)$$

In summary, at this point we should understand that the discrete representation of our transmission reflects the properties of its continuous counterpart, and has the following characteristics

- The transmission rate is  $2W$  symbol/s
- The average transmitted signal energy per symbol,  $\bar{E}_{sym}$ , is limited to  $\bar{P}/(2W)$  J/symbol
- The set of noise samples,  $\{n_n\}$ , is composed of real i.i.d. zero-mean

Gaussian random variables, with variance  $N_0/2$ . In other words,  $n_n \sim \mathcal{N}(0, N_0/2)$ .

- The signal-to-noise ratio,  $SNR$ , is  $\frac{\bar{E}_{sym}}{N_0/2} = \frac{\bar{P}/2W}{N_0/2} = \frac{\bar{P}}{WN_0}$

### 3.1.2 Passband

Consider again a continuous-time AWGN channel with bandwidth  $W$  Hz, average power constraint of  $\bar{P}$  Watt, and white Gaussian noise with a PSD of  $N_0/2$ .

$$y(t) = x(t) + n(t) \quad (3.39)$$

where  $x(t)$  is the transmitted *passband* signal,  $y(t)$  is the received signal, and  $n(t)$  is white Gaussian noise.

In a similar fashion to subsection 3.1.1 we can also prove the following

- The transmission rate is  $W$  (complex) symbol/s
- The average transmitted signal energy per (complex) symbol,  $\bar{E}_{sym}$ , is limited to  $\bar{P}/W$  J<sub>symbol</sub>
- The set of noise samples,  $\{n_n\}$ , is composed of complex i.i.d. zero-mean Gaussian random variables, with variance  $N_0$ . In other words,  $n_n \sim \mathcal{CN}(0, N_0)$ . In addition, the real and imaginary parts of each  $n_n$  are i.i.d. zero-mean Gaussian random variables with variance  $N_0/2$ , namely  $\Re\{n_n\} \sim \mathcal{N}(0, N_0/2)$ , and  $\Im\{n_n\} \sim \mathcal{N}(0, N_0/2)$ .
- The signal-to-noise ratio,  $SNR$ , is  $\frac{\bar{E}_{sym}}{N_0} = \frac{\bar{P}/W}{N_0} = \frac{\bar{P}}{WN_0}$

## 3.2 Flat Channel

The Shannon-Hartley theorem states that the capacity,  $C$ , of an AWGN channel, with bandwidth,  $B$ , average received signal power over the bandwidth,  $S$ , average power of the noise and interference over the bandwidth,  $N$ , is the following

$$C = B \cdot \log_2 \left( 1 + \frac{S}{N} \right) \text{ bit/s} \quad (3.40)$$

Now consider the discrete time baseband equivalent of our channel

$$y[m] = x[m] + n[m] \quad (3.41)$$

It can be proved that, for all  $m$ ,  $n[m]$  are i.i.d complex normal random variables with variance  $N_0$ , namely  $n[m] \sim \mathcal{CN}(0, N_0)$ . It can also be shown that the real and imaginary parts of  $n[m]$  are i.i.d. normal random variables with variance  $N_0/2$ , namely  $\Re\{n[m]\} \sim \mathcal{N}(0, N_0/2)$  and  $\Im\{n[m]\} \sim \mathcal{N}(0, N_0/2)$ . Of course, the capacity is still the same as its continuous counterpart.

Now consider the following case where the channel is flat, but not ideal, and time invariant

$$y[m] = h \cdot x[m] + n[m] \quad (3.42)$$

$$y[m] = h \cdot x[m] + n[m] \quad (3.43)$$

If we multiply our output by  $h^*/|h|$ , we have that

$$\frac{h^*}{|h|}y[m] = |h| \cdot x[m] + \frac{h^*}{|h|}n[m] \quad (3.44)$$

It can be shown that the new noise has the same statistics as  $n[m]$  and that the received energy per symbol remains the same, since  $h^*/|h|$  just rotates each symbol without scaling it. Hence, this problem is equivalent to the former in terms of receive SNR.

We can further manipulate our formulation of the problem, and assume that we have an AWGN channel, but in this case we scale all our transmitted symbols by  $|h|$

$$y[m] = x_{scaled}[m] + n[m] \Rightarrow \quad (3.45)$$

$$y[m] = |h| \cdot x[m] + n[m] \quad (3.46)$$

This third formulation is also equivalent to the other two in terms of receive SNR. Furthermore, we know that we can detect our symbols independently of each other, and therefore we may discard the index “m”, which results to the following model

$$y = x_{scaled} + n \Rightarrow \quad (3.47)$$

$$y = |h| \cdot x + n \quad (3.48)$$

Hence, we can now derive the capacity of the original channel through the third formulation as

$$C = W \cdot \log_2 \left( 1 + \frac{E \{|x_{scaled}|^2\}}{N_0} \right) \Rightarrow \quad (3.49)$$

$$C = W \cdot \log_2 \left( 1 + \frac{E \{|h|x|^2\}}{N_0} \right) \Rightarrow \quad (3.50)$$

$$C = W \cdot \log_2 \left( 1 + \frac{|h|^2 E \{|x|^2\}}{N_0} \right) \Rightarrow \quad (3.51)$$

$$C = W \cdot \log_2 \left( 1 + \frac{|h|^2 \bar{E}_{sym}}{N_0} \right) \quad (3.52)$$

Thus, we have that

$$\boxed{C = W \cdot \log_2 \left( 1 + |h|^2 \frac{\bar{P}}{N_0 W} \right) \text{ bit/s}} \quad (3.53)$$

### 3.3 Frequency-selective Channel

Consider the following time-invariant L-tap frequency-selective AWGN channel

$$y[m] = \sum_{l=0}^{L-1} h_l x[m-l] + w[m] \quad (3.54)$$

where we again constrain our signal to an average power of  $\bar{P}$  Watt, a passband bandwidth of  $W$  Hz, while the noise is white Gaussian with PSD  $N_0/2$ .

If we employ Orthogonal Frequency-Division Multiplexing (OFDM), we can diminish the intersymbol interference (ISI) caused by the frequency selective channel, by sending our symbols on a set of flat AWGN channels. Each symbol will be transmitted into its own flat AWGN channel, without any interaction with the other symbols or channels. In information theory jargon, a set of non-interfering sub-channels where independent noise corresponds to each of them, is called a parallel channel.

Specifically, we first group our symbols in blocks of size  $N_c$ , and then we use each of these blocks as an input to the OFDM modulator of the transmitter. At the output of the modulator, we get a new block of intermediate symbols of size  $N_c + L - 1$ , which is the set of symbols that are actually transmitted. At the receiver, we have the corresponding OFDM demodulator, which receives the transmitted block of intermediate symbols, and finally retrieves at its output a modified version of the original OFDM block, where each of its elements has been independently multiplied by a complex number, and corrupted by some additive noise,  $\tilde{w}_n$ . It can be proved that these complex numbers correspond to the DFT,  $\{\tilde{h}_n\}$ , of the  $L$  channel taps, and that the additive noise, is in fact white Gaussian with the same statistics as the noise in our original problem formulation. In summary, our transmission process has been modified as

$$\tilde{\mathbf{y}}_k = \tilde{\mathbf{h}} \cdot \mathbf{x}_k + \tilde{\mathbf{w}}_k, \quad , k \in 0, 1, 2, \dots \quad (3.55)$$

where

$$\mathbf{x}_k := \begin{bmatrix} x[0 + kN_c] \\ x[1 + kN_c] \\ x[2 + kN_c] \\ \vdots \\ x[(N_c - 1) + kN_c] \end{bmatrix} \quad (3.56)$$

is the  $k$ -th OFDM block,

$$\tilde{\mathbf{w}}_k := \begin{bmatrix} \tilde{w}[0 + kN_c] \\ \tilde{w}[1 + kN_c] \\ \tilde{w}[2 + kN_c] \\ \vdots \\ \tilde{w}[(N_c - 1) + kN_c] \end{bmatrix} \quad (3.57)$$

is a complex white Gaussian noise vector,

$$\tilde{\mathbf{y}}_k := \begin{bmatrix} \tilde{y}[0 + kN_c] \\ \tilde{y}[1 + kN_c] \\ \tilde{y}[2 + kN_c] \\ \vdots \\ \tilde{y}[(N_c - 1) + kN_c] \end{bmatrix} \quad (3.58)$$

is the  $k$ -th OFDM block corrupted by the channel and the noise, and

$$\tilde{\mathbf{h}} := \begin{bmatrix} \tilde{h}_0 \\ \tilde{h}_1 \\ \tilde{h}_2 \\ \vdots \\ \tilde{h}_{N_c-1} \end{bmatrix} \quad (3.59)$$

Thus, another way to express (3.55) is as follows

$$\tilde{y}[m] = \tilde{h}_l x[m] + \tilde{w}[m] \quad (3.60)$$

where

$$l = m \bmod N_c \quad (3.61)$$

By comparing (3.60) to (3.54), it becomes evident that the frequency-selective channel has indeed been converted to a flat channel. Note, however, that the number of symbols,  $N_c + L - 1$ , at the output of the OFDM modulator is greater than the number of symbols,  $N_c$ , at its input. This implies that the overall transmission rate using OFDM modulation will be lower, and equal to  $N_c/(N_c + L - 1)$  times the transmission rate of our original problem formulation. Equivalently, we may say that after the receiver has collected all the symbols of a transmitted OFDM block, it will have to wait for a time interval of  $L - 1$  samples before collecting symbols for the next OFDM block, despite the fact that the transmitter never stops transmitting; this happens because there is no useful information in the symbols during this time interval, and thus they are discarded by the receiver. On the other hand, as the size,  $N_c$ , of the OFDM block gets bigger, the number,  $L$ , of the channel taps becomes negligible and thus the coefficient  $N_c/(N_c + L - 1)$  will tend to become equal to 1. This in turn implies that when the size of the OFDM block tends to infinity, then the transmission rate using OFDM modulation will tend to the transmission rate of the original transmission formulation.

Thus, if we calculate the capacity,  $C_{N_c}$ , for the OFDM formulation with a finite OFDM block size of  $N_c$ , then we can actually figure out the capacity of our original transmission scheme by taking the limit of  $C_{N_c}$  when  $N_c$  tends to infinity.

Before we proceed with the capacity calculations, we need first to convert the power constraint of our symbols in time to a power constraint in frequency, by employing the Parseval theorem for DFT. Without getting into further details, we can prove that this can be achieved as follows

$$\sum_{n=0}^{N_c-1} \bar{E}_n \leq N_c \frac{\bar{P}}{\bar{W}} \quad (3.62)$$

where we assumed a  $\bar{E}_n$  J/symbol power constraint for the  $n$ -th element of the OFDM blocks.

Since each sub-channel is considered flat, the capacity of the  $n$ -th sub-channel, can now be calculated as

$$W \log_2 \left( 1 + \frac{|\tilde{h}_n|^2 \bar{E}_n}{N_0} \right) \text{ bit/s} \quad (3.63)$$

where dividing by  $W$  symbol/s leads to a capacity of

$$\log_2 \left( 1 + \frac{|\tilde{h}_n|^2 \bar{E}_n}{N_0} \right) \text{ bit/symbol} \quad (3.64)$$

Thus, for the whole OFDM block we have a rate (not capacity) of

$$\sum_{n=0}^{N_c-1} \log_2 \left( 1 + \frac{|\tilde{h}_n|^2 \bar{E}_n}{N_0} \right) \text{ bit/OFDM block} \quad (3.65)$$

We can now multiply this by  $1/N_c$  OFDM block/symbol and then by  $W$  symbol/s to calculate the total rate of the system as

$$R_{N_c}(\bar{\mathbf{E}}) = \frac{1}{N_c} \sum_{n=0}^{N_c-1} W \log_2 \left( 1 + \frac{|\tilde{h}_n|^2 \bar{E}_n}{N_0} \right) \text{ bit/s} \quad (3.66)$$

where

$$\bar{\mathbf{E}} = [\bar{E}_0 \quad \bar{E}_1 \quad \bar{E}_2 \quad \dots \quad \bar{E}_{N_c-1}]^T \quad (3.67)$$

In order to determine the capacity,  $C_{N_c}$ , of the whole channel, we also need to determine the optimal values of  $\bar{E}_n$ , so that (3.66) is maximized. In other words, we need to properly allocate the available signal power across the  $N_c$  sub-channels. Specifically, we have that

$$C_{N_c} = \max_{\bar{\mathbf{E}}} \{R_{N_c}(\bar{\mathbf{E}})\} = \max_{\bar{\mathbf{E}}} \left\{ \frac{1}{N_c} \sum_{n=0}^{N_c-1} W \log_2 \left( 1 + \frac{|\tilde{h}_n|^2 \bar{E}_n}{N_0} \right) \right\} \quad (3.68)$$

subject to

$$\sum_{n=0}^{N_c-1} \bar{E}_n \leq N_c \frac{\bar{P}}{W} \quad (3.69)$$

and

$$\bar{E}_n \geq 0 \quad n = 0, 1, 2, \dots, N_c - 1 \quad (3.70)$$

This is in fact a convex optimization problem, which can be solved by the water-filling algorithm as explained in subsection 3.3.1. For clarity we should also mention that, if we change the base of the logarithm from 2 to  $e$ , then (3.68) can be modified as

$$C_{N_c} = \frac{W}{N_c \ln 2} \max_{\bar{\mathbf{E}}} \left\{ \sum_{n=0}^{N_c-1} \ln \left( 1 + \frac{|\tilde{h}_n|^2 \bar{E}_n}{N_0} \right) \right\} \quad (3.71)$$

and our maximization problem is now fully compatible with the water-filling problem formulation in subsection 3.3.1. Finally, the algorithm provides us with the following two equations

$$\boxed{\bar{E}_n = \max \left\{ 0, \frac{1}{v} - \frac{N_0}{|\tilde{h}_n|^2} \right\}} \quad (3.72a)$$

$$\sum_{i=0}^{N_c-1} \bar{E}_n = N_c \frac{\bar{P}}{W} \quad (3.72b)$$

Taking now into account (3.72a) and (1.42), we can define a continuous function of frequency,  $\bar{E}(f)$ , such that

$$\bar{E} \left( \frac{nW}{N_c} \right) = \bar{E}_n = \max \left\{ 0, \frac{1}{v} - \frac{N_0}{\left| G_d \left( \frac{n}{N_c} \right) \right|^2} \right\} \quad (3.73)$$

and

$$\sum_{i=0}^{N_c-1} \bar{E} \left( \frac{iW}{N_c} \right) = N_c \frac{\bar{P}}{W} \quad (3.74)$$

When the number of sub-channels gets big enough and tends to infinity, we can infer by (3.73) that

$$\lim_{N_c \rightarrow +\infty} \bar{E} \left( \frac{nW}{N_c} \right) = \lim_{N_c \rightarrow +\infty} \max \left\{ 0, \frac{1}{v} - \frac{N_0}{\left| G_d \left( \frac{n}{N_c} \right) \right|^2} \right\}, n = 0, \dots, N_c - 1$$

which is equivalent to

$$\bar{E}(F) = \max \left\{ 0, \frac{1}{v} - \frac{N_0}{\left| G_d \left( \frac{F}{W} \right) \right|^2} \right\}, F \in [0, W) \quad (3.75)$$

Likewise, by (3.74) we will have that

$$\sum_{i=0}^{N_c-1} \bar{E} \left( \frac{iW}{N_c} \right) \frac{1}{N_c} = \frac{\bar{P}}{W} \Rightarrow \quad (3.76)$$

$$\lim_{N_c \rightarrow +\infty} \sum_{i=0}^{N_c-1} \bar{E} \left( \frac{iW}{N_c} \right) \frac{W}{N_c} = \lim_{N_c \rightarrow +\infty} \bar{P} \stackrel{\delta F = \frac{W}{N_c}}{\Leftrightarrow} \quad (3.77)$$

$$\lim_{\delta F \rightarrow 0} \sum_{i=0}^{\frac{W}{\delta F}-1} \bar{E}(i \cdot \delta F) \delta F = \bar{P} \Leftrightarrow \quad (3.78)$$

$$\int_0^W \bar{E}(F) dF = \bar{P} \Leftrightarrow \quad (3.79)$$

$$\int_{-\frac{W}{2}}^{\frac{W}{2}} \bar{E}(F) dF = \bar{P} \quad (3.80)$$

In a similar fashion, (3.66) gives the following

$$\begin{aligned} \lim_{N_c \rightarrow +\infty} R_{N_c}(\bar{E}(F)) &= \lim_{N_c \rightarrow +\infty} \frac{1}{N_c} \sum_{n=0}^{N_c-1} W \log_2 \left( 1 + \frac{|G_d\left(\frac{n}{N_c}\right)|^2 \bar{E}\left(\frac{nW}{N_c}\right)}{N_0} \right) \Rightarrow \\ R(\bar{E}(F)) &= \int_0^W \log_2 \left( 1 + \frac{|G_d\left(\frac{F}{W}\right)|^2 \bar{E}(F)}{N_0} \right) dF \Leftrightarrow \\ R(\bar{E}(F)) &= \int_{-\frac{W}{2}}^{\frac{W}{2}} \log_2 \left( 1 + \frac{|G_d\left(\frac{F}{W}\right)|^2 \bar{E}(F)}{N_0} \right) dF \Leftrightarrow \\ R(\bar{E}(F)) &= \int_{-\frac{W}{2}}^{\frac{W}{2}} \log_2 \left( 1 + \frac{|H^b(F)|^2 \bar{E}(F)}{N_0} \right) dF \end{aligned}$$

In summary, we conclude that, when we have *passband* transmission in a frequency-selective AWGN channel, with the following constraints

- an average signal power of  $\bar{P}$  Watt
- a bandwidth of  $W$  Hz
- white Gaussian noise with PSD  $N_0/2$  Watt/Hz

then, for a given power distribution,  $\bar{E}(F)$ , over this bandwidth, the maximum transmission rate for reliable communication is

$$R(\bar{E}(F)) = \int_{F_c - \frac{W}{2}}^{F_c + \frac{W}{2}} \log_2 \left( 1 + \frac{|H(F)|^2 \bar{E}(F)}{N_0} \right) dF \quad (3.81)$$

and we can calculate the capacity of the channel by determining the optimal power allocation over the signal bandwidth as follows

$$\bar{E}(F) = \max \left\{ 0, \frac{1}{v} - \frac{N_0}{|H(F)|^2} \right\} \quad (3.82a)$$

$$\int_{F_c - \frac{W}{2}}^{F_c + \frac{W}{2}} \bar{E}(F) dF = \bar{P} \quad (3.82b)$$

### 3.3.1 Water-filling

We define the following optimization problem

$$\begin{aligned} \underset{\mathbf{P}}{\text{maximize}} \quad & f(\mathbf{P}) = \sum_{i=0}^{N_c-1} \ln \left( 1 + \frac{P_i |\tilde{h}_i|^2}{N_0} \right) \\ \text{subject to} \quad & g(\mathbf{P}) = \left( \sum_{i=0}^{N_c-1} P_i \right) - N_c P_{tot} \leq 0 \\ & f_n(\mathbf{P}) = -P_n \leq 0 \quad , n = 0, 1, 2, \dots, N_c - 1 \end{aligned}$$

where  $\mathbf{P} = [P_0 \ P_1 \ P_2 \ P_3 \ \dots \ P_{N_c-1}]^T$

For simplicity and clarity reasons, we are going to denote the fraction  $N_0/|\tilde{h}_n|^2$  by  $a_n$  from now on. We should also mention that since both  $N_0$  and  $|\tilde{h}_n|^2$  are positive numbers,  $a_n$  will be positive for every  $n = 0, 1, 2, \dots, N_c - 1$  as well.

It is easy to see that this is actually a convex optimization problem, since  $f(\mathbf{P})$ ,  $g(\mathbf{P})$  and  $f_n(\mathbf{P})$  are all convex functions of  $\mathbf{P}$ . It is also helpful to notice that the Slater condition is satisfied for this problem, since for  $P_n = P_{tot}/2$ , we have that  $g(\mathbf{P}) < 0$  and  $f_n(\mathbf{P}) < 0$ . Therefore, the Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient for optimality, which in turn implies that we can calculate the optimal  $\mathbf{P}$  by solving the following set of constraints

$$\left\{ \begin{array}{l} \nabla f(\mathbf{P}) = v \nabla g(\mathbf{P}) + \sum_{i=0}^{N_c-1} \lambda_i \nabla f_i(\mathbf{P}) \\ \lambda_n \geq 0 \\ v \geq 0 \\ \lambda_n P_n = 0 \\ v \cdot g(\mathbf{P}) = 0 \\ g(\mathbf{P}) \leq 0 \\ f_n(\mathbf{P}) \leq 0 \end{array} \right. \quad (3.83)$$

$$\lambda_n \geq 0 \quad (3.84)$$

$$\lambda_n P_n = 0 \quad (3.85)$$

$$v \cdot g(\mathbf{P}) = 0$$

$$g(\mathbf{P}) \leq 0$$

$$f_n(\mathbf{P}) \leq 0 \quad , n = 0, 1, 2, \dots, N_c - 1$$

Beginning with (3.83), we have that

$$\begin{aligned} \nabla \sum_{i=0}^{N_c-1} \ln \left( 1 + \frac{P_i}{a_i} \right) &= v \nabla \left[ \left( \sum_{i=0}^{N_c-1} P_i \right) - N_c P_{tot} \right] + \sum_{i=0}^{N_c-1} \lambda_i \nabla (-P_i) \Rightarrow \\ \sum_{i=0}^{N_c-1} \nabla \ln \left( 1 + \frac{P_i}{a_i} \right) &= v \sum_{i=0}^{N_c-1} \nabla P_i - \sum_{i=0}^{N_c-1} \lambda_i \nabla P_i \Rightarrow \\ \begin{bmatrix} \frac{1}{(1+P_0/a_0)} \frac{1}{a_0} \\ \frac{1}{(1+P_1/a_1)} \frac{1}{a_1} \\ \vdots \\ \frac{1}{(1+P_{N_c-1}/a_{N_c-1})} \frac{1}{a_{N_c-1}} \end{bmatrix} &= v \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} - \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{N_c-1} \end{bmatrix} \Rightarrow \\ \frac{1}{a_n + P_n} &= v - \lambda_n \Rightarrow \quad , n = 0, 1, 2, \dots, N_c - 1 \\ \lambda_n &= v - \frac{1}{a_n + P_n} \quad , n = 0, 1, 2, \dots, N_c - 1 \end{aligned} \quad (3.86)$$

Plugging (3.86) into (3.84) and (3.85), we get the following set of constraints

$$\left\{ \begin{array}{l} v - \frac{1}{a_n + P_n} \geq 0 \\ v \geq 0 \end{array} \right. \quad (3.87)$$

$$\left\{ \begin{array}{l} \left( v - \frac{1}{a_n + P_n} \right) P_n = 0 \\ v \left[ \left( \sum_{i=0}^{N_c-1} P_i \right) - N_c P_{tot} \right] = 0 \end{array} \right. \quad (3.88)$$

$$\left\{ \begin{array}{l} \left( \sum_{i=0}^{N_c-1} P_i \right) - N_c P_{tot} \leq 0 \end{array} \right. \quad (3.89)$$

$$\left\{ \begin{array}{l} P_n \geq 0 \quad , n = 0, 1, 2, \dots, N_c - 1 \end{array} \right. \quad (3.90)$$

If we assume that  $v = 0$ , then from (3.87) we have that

$$\begin{aligned} -\frac{1}{a_n + P_n} \geq 0 &\Rightarrow \\ a_n + P_n \leq 0 &\Rightarrow \\ P_n \leq -a_n & \end{aligned}$$

where the last equation contradicts (3.90), and thus it necessarily holds that  $v > 0$ .

Looking at (3.88), we can see that since  $v > 0$ , we should have that

$$\begin{aligned} \left( \sum_{i=0}^{N_c-1} P_i \right) - N_c P_{tot} = 0 &\Rightarrow \\ \sum_{i=0}^{N_c-1} P_i = N_c P_{tot} & \end{aligned} \quad (3.91)$$

*Equation (3.91) has an important physical interpretation, since it confirms our intuition that, if we need to calculate the optimal power allocation, then we indeed have to utilize all the available power.*

Regrouping again all our remaining constraints, we take the following set

$$\left\{ \begin{array}{l} v \geq \frac{1}{a_n + P_n} \\ v > 0 \end{array} \right. \quad (3.92)$$

$$\left\{ \begin{array}{l} \left( v - \frac{1}{a_n + P_n} \right) P_n = 0 \\ \sum_{i=0}^{N_c-1} P_i = N_c P_{tot} \\ P_n \geq 0 \quad , n = 0, 1, 2, \dots, N_c - 1 \end{array} \right. \quad (3.93)$$

We now consider the following two cases

**Case 1:**  $v < 1/a_n$

If we assume that  $P_n = 0$ , then from (3.92) we can see that  $v \geq 1/a_n$ , which clearly contradicts our assumption that  $v < 1/a_n$ . Hence, it holds that  $P_n > 0$ , which, together with (3.93), implies that

$$\begin{aligned} v - \frac{1}{a_n + P_n} = 0 &\Rightarrow \\ v &= \frac{1}{a_n + P_n} \Rightarrow \\ P_n &= \frac{1}{v} - a_n \end{aligned}$$

We should note that from our case assumption we have that

$$\begin{aligned} v &< \frac{1}{a_n} \stackrel{v \geq 0}{\Rightarrow} \\ \frac{1}{v} - a_n &> 0 \Rightarrow \\ P_n &> 0 \end{aligned}$$

which means that the constraint  $P_n > 0$  is redundant.

Regrouping once more the remaining constraints, we take the following set

$$\left\{ \begin{array}{l} v \geq \frac{1}{a_n + P_n} \\ v > 0 \\ P_n = \frac{1}{v} - a_n \\ \sum_{i=0}^{N_c-1} P_i = N_c P_{tot} \end{array} \right.$$

**Case 2:**  $v \geq 1/a_n$

If we assume that  $P_n > 0$ , then we have that

$$v \geq \frac{1}{a_n} > \frac{1}{a_n + P_n}$$

And thus, it holds that

$$\begin{aligned} v - \frac{1}{a_n + P_n} &> 0 \Rightarrow \\ P_n \left( v - \frac{1}{a_n + P_n} \right) &> 0 \end{aligned}$$

which directly contradicts our constraint in (3.93), and therefore we get that  $P_n = 0$ .

Regrouping one more time the remaining constraints, we take the following set

$$\left\{ \begin{array}{l} v \geq \frac{1}{a_n + P_n} \\ v > 0 \\ \sum_{i=0}^{N_c-1} P_i = N_c P_{tot} \\ P_n = 0 \quad , n = 0, 1, 2, \dots, N_c - 1 \end{array} \right.$$

Combining the constraint sets of both cases, we can now obtain the following constraint set

$$\left\{ \begin{array}{l} v \geq \frac{1}{a_n + P_n} \\ v > 0 \end{array} \right. \quad (3.94)$$

$$\left\{ \begin{array}{l} \sum_{i=0}^{N_c-1} P_i = N_c P_{tot} \end{array} \right. \quad (3.95)$$

$$P_n = \begin{cases} \frac{1}{v} - a_n & v < \frac{1}{a_n} \\ 0 & v \geq \frac{1}{a_n} \end{cases} \quad (3.96)$$

Manipulating (3.96), we can see that

$$\begin{aligned} P_n &= \begin{cases} \frac{1}{v} - a_n & v < \frac{1}{a_n} \\ 0 & v \geq \frac{1}{a_n} \end{cases} \quad \begin{matrix} u \geq 0 \\ \Rightarrow \end{matrix} \\ P_n &= \begin{cases} \frac{1}{v} - a_n & a_n < \frac{1}{v} \\ 0 & a_n \geq \frac{1}{v} \end{cases} \quad \Rightarrow \\ P_n &= \begin{cases} \frac{1}{v} - a_n & \frac{1}{v} - a_n > 0 \\ 0 & \frac{1}{v} - a_n \leq 0 \end{cases} \quad \Rightarrow \\ P_n &= \max \left\{ 0, \frac{1}{v} - a_n \right\} \quad (\geq 0) \end{aligned} \quad (3.97)$$

We can also manipulate (3.94) to get the following

$$\begin{aligned} v &\geq \frac{1}{a_n + P_n} \quad \begin{matrix} P_n \geq 0, \\ \Leftrightarrow \\ v > 0 \end{matrix} \\ a_n + P_n &\geq \frac{1}{v} \quad \Leftrightarrow \\ P_n &\geq \frac{1}{v} - a_n \end{aligned}$$

If we look carefully, we can observe that all the information in this inequality is contained in (3.97), and thus we can conclude that (3.94) is redundant.

Furthermore, if we plug (3.97) into (3.95), we take

$$\sum_{i=0}^{N_c-1} \max \left\{ 0, \frac{1}{v} - a_i \right\} = N_c P_{tot} \quad (3.98)$$

from which we can easily see that when  $v \leq 0$ , then it must hold that

$$\sum_{i=0}^{N_c-1} 0 = N_c P_{tot}$$

which, of course, is impossible. Thus, (3.98) implies that  $u > 0$ , and consequently the latter is a redundant constraint as well.

Finally, we can replace  $a_n$  with  $N_0/|\tilde{h}_n|^2$ , and obtain the following reduced constraint set

$$\boxed{\begin{aligned} P_n &= \max \left\{ 0, \frac{1}{v} - \frac{N_0}{|\tilde{h}_n|^2} \right\} \\ \sum_{i=0}^{N_c-1} \max \left\{ 0, \frac{1}{v} - \frac{N_0}{|\tilde{h}_i|^2} \right\} &= N_c P_{tot} \end{aligned}}$$

To solve these equations, we begin with the second one from which we calculate the optimal value for  $v$ . Next, we plug it into the first formula, which in turn gives us the optimal values for all  $P_n$ .

## 3.4 Real-World Channels

In this chapter we present the TucStats software, which can prove a useful tool for analysis of MIMObit generated data, and we discuss various results that have been produced using it in comparison with the respective results that MIMObit provides. The main purpose of this comparison is to stretch MIMObit's capabilities and study the extend to which its end-products adhere to valid wireless communications theoretical models.

### 3.4.1 TucStats Software

TucStats is a Matlab based graphical tool that aims to provide a comprehensive, flexible and interactive experience for analysis on simulation data generated by MIMObit. It is compatible only with data produced by the PropStats App of MIMObit and it is strongly recommended that you do not attempt to import files produced by the RadMap or OrientStats Apps. Note that it has been specifically designed for scenarios with a single multipoint transmitter and a single multipoint receiver, and thus you should only consider this setup if you intend to do post-processing using TucStats. Specifically for the capacity calculations, only the first port of the receiver and the transmitter are considered, and hence one cannot study the capacity corresponding to other ports. Nevertheless, the code design of the software is flexible

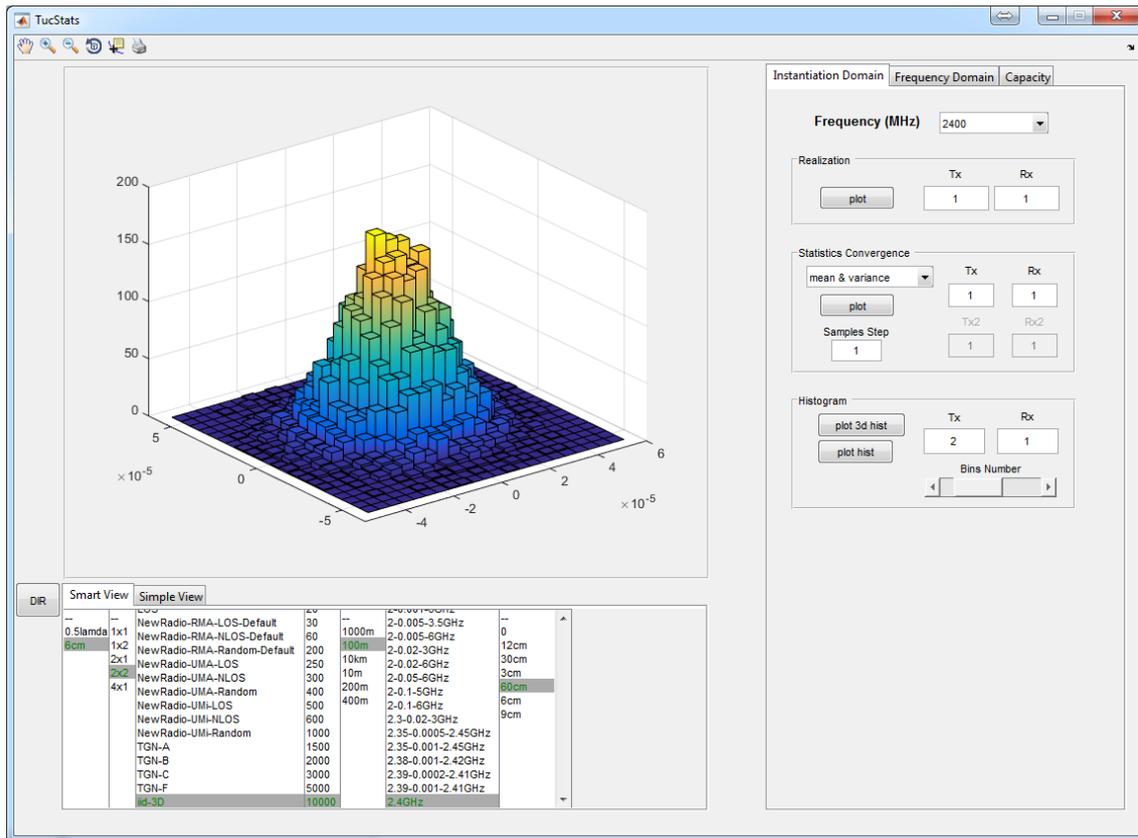


Figure 3.2

enough, that it would be relatively easy to be upgraded to support general MIMObit simulation scenarios. Last, the script of TucStats is compatible only with Matlab 2014b or later, and it is recommended that you have installed all the toolboxes that the license of the Technical University of Crete includes.

The graphical interface of TucStats can be broken down into three sections:

### Plots

All the plots in TucStats appear at the upper-left portion of the interface. There is also the option to pan, zoom and rotate the plots, and navigate through the graph data. Finally, there is the option to create an independent figure of the plot for use outside the TucStats environment.

### Importing \*.Htotalresults.dat files

The bottom-left part allows us to select the PropStats \*.Htotalresults.dat file that we need to load in TucStats. Specifically, we can use the “Simple View” tab, which simply lists all the \*.Htotalresults.dat files in the current directory, or the “Smart View” tab which offers a more flexible and efficient way to search through the various files; in both cases, when a file is selected, it is immediately loaded.

Although, “Smart View” can be very helpful when we deal we large numbers of files, the files need to adhere to a specific naming format, which in turn corresponds to a specific MIMObit simulation setup. The naming convention is as follows

```
<dipole_length>_
```

```

<Tx Ports No>x<Rx Ports No>_
<propagation environment>_
<instantiations No>samples_
<Tx Rx distance>_
<simulation frequencies>_
<ports distance>_
.Htotalresults.dat

```

and the corresponding lists in the interface are ordered the same way from left to right. When we have selected some items in a few lists, then the items in other lists that correspond to combinations for a file that exists turn green. It is important to remember that the rest of MIMObit parameters that are not included in this file naming convention, are assumed to have their default values during a MIMObit simulation.

Note that, for aesthetic purposes, the scroll bars for the most lists have been removed. In order to scroll, we can click the scroll button of the mouse in a list, and then move the pointer to the desired scroll direction.

Finally, the “DIR” button allows the user to specify the path of the files; the default path is the path of TucStats script.

### Data Analysis

The available analysis tools in TucStats are located at the right part of the graphical interface, and can be split into three categories:

- **Instantiation Domain** (see Figure 3.2)

In this tab, we first select the desired simulation frequency, and then we can plot the realization, the convergence of various quantities like mean, variance and correlation, and finally plot the histogram of the samples at that frequency. In the case of a multiport transmitter or receiver, we also have an option for the desired combination of ports.

- **Frequency Domain** (see Figure 3.3)

This tab offers the ability to study the behavior of an antenna system for a specific propagation environment instantiation, at the frequency level. The only option currently available is the ability to plot the spectrum for an instantiation, or the average over all instantiations. Further, we can manipulate the spectrum by altering the  $G(F)$  function. For example,  $H.*freq$  will plot a graph where the channel has been multiplied by the corresponding frequency at each point.

We can also focus in the frequency range that we are interested in, by setting the x-axis limits using the respective bars.

- **Capacity** (see Figure 3.4)

Capacity tab gives us the ability to study four different kinds of capacity

- *Flat*

The channel is considered flat at the specified bandwidth and equal to

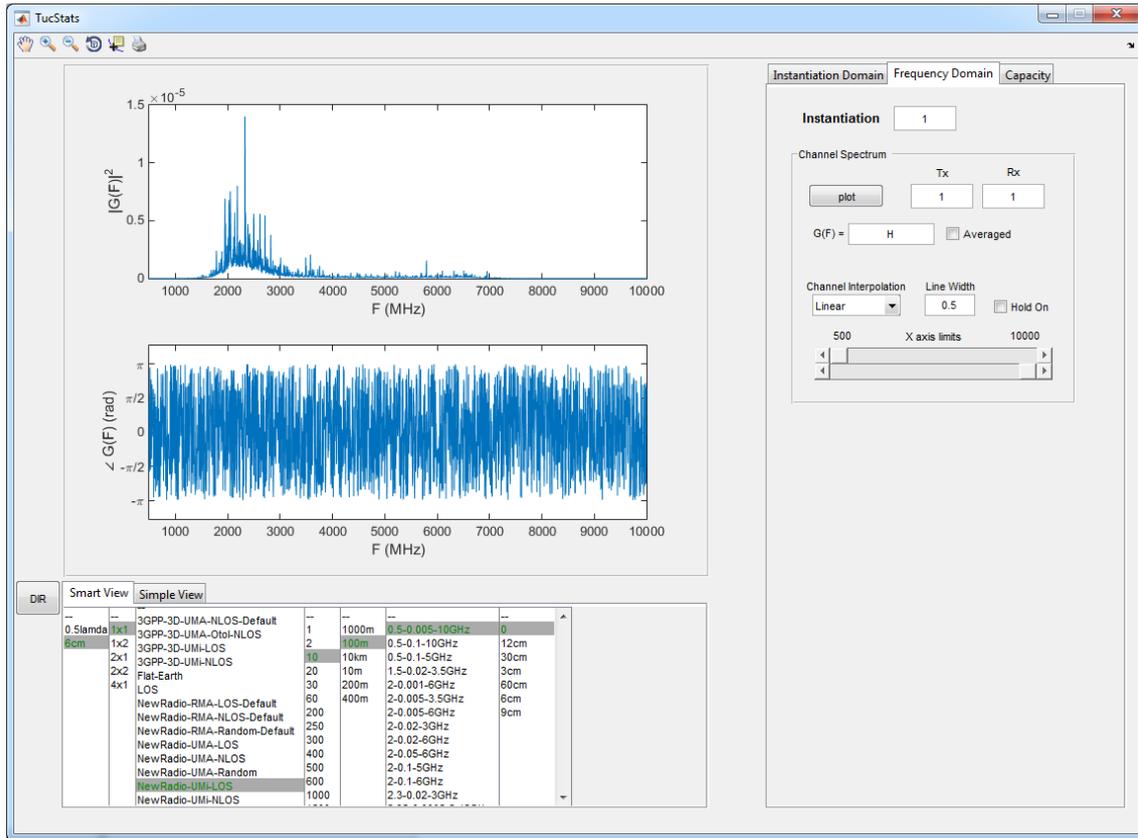


Figure 3.3

the value of the simulated channel at the carrier frequency. As implied by formula (3.40), the available power is allocated uniformly over the bandwidth. We note that, in the case of a flat channel, this is also the optimal power allocation.

– *Open-Loop*

The formula (3.81) for the frequency-selective channel is utilized along with a linear interpolated channel approximation and a uniform power distribution over the specified bandwidth.

The channel is approximated by a function that occurs from using linear interpolation on the frequency points that we have run the selected MI-MObit simulation at. Note that this function is defined to have a value of zero at points outside the simulated frequency region.

– *Closed-Loop*

Same as the Open-Loop capacity, but instead of a uniform power distribution, the water-filling algorithm (see subsection 3.3.1) is employed in order to achieve an optimal power distribution. *It is also really important to keep in mind that, if not handled properly, the implemented algorithm for the calculation of this capacity type can produce completely erroneous results. Specifically, we make use of Matlab's "fsolve()" function in order to calculate a solution for equations (3.82), which can return very low accuracy solutions. In the current version of TucStats, we can manually remedy this issue by restarting the algorithm with a different starting point until the error in our solution is acceptable.*

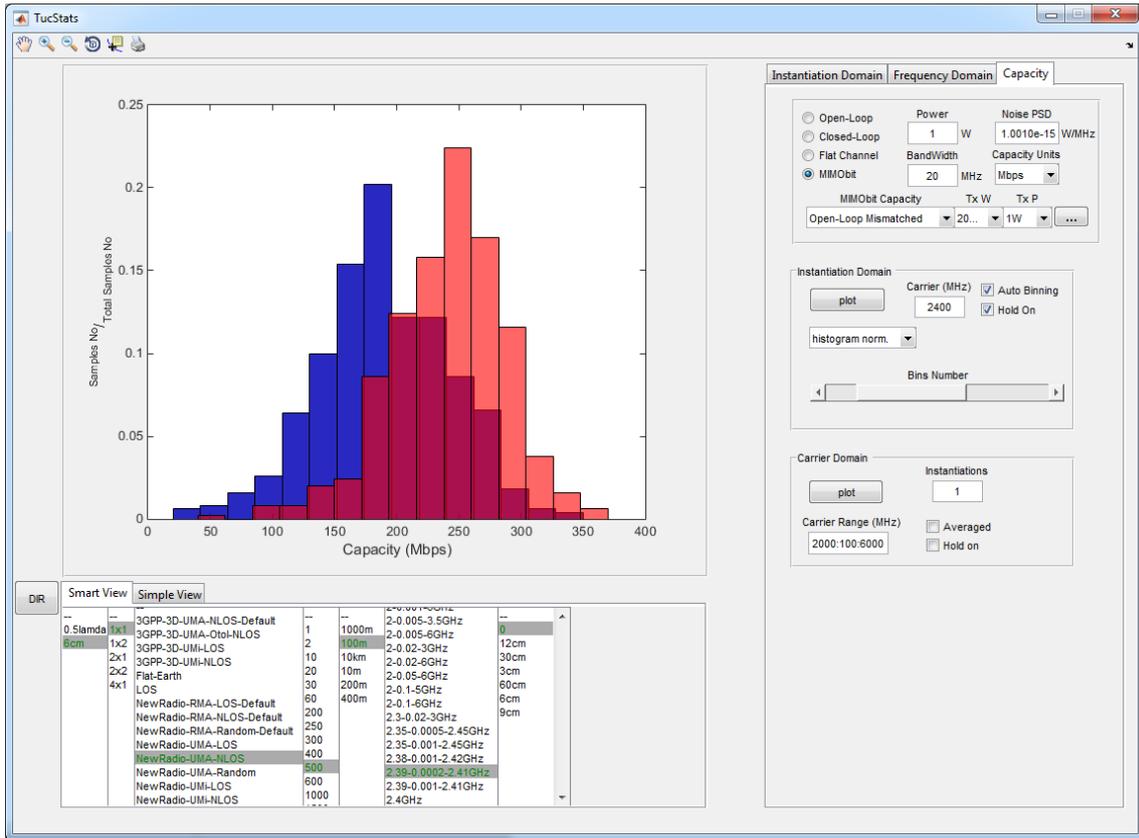


Figure 3.4

### – MIMObit

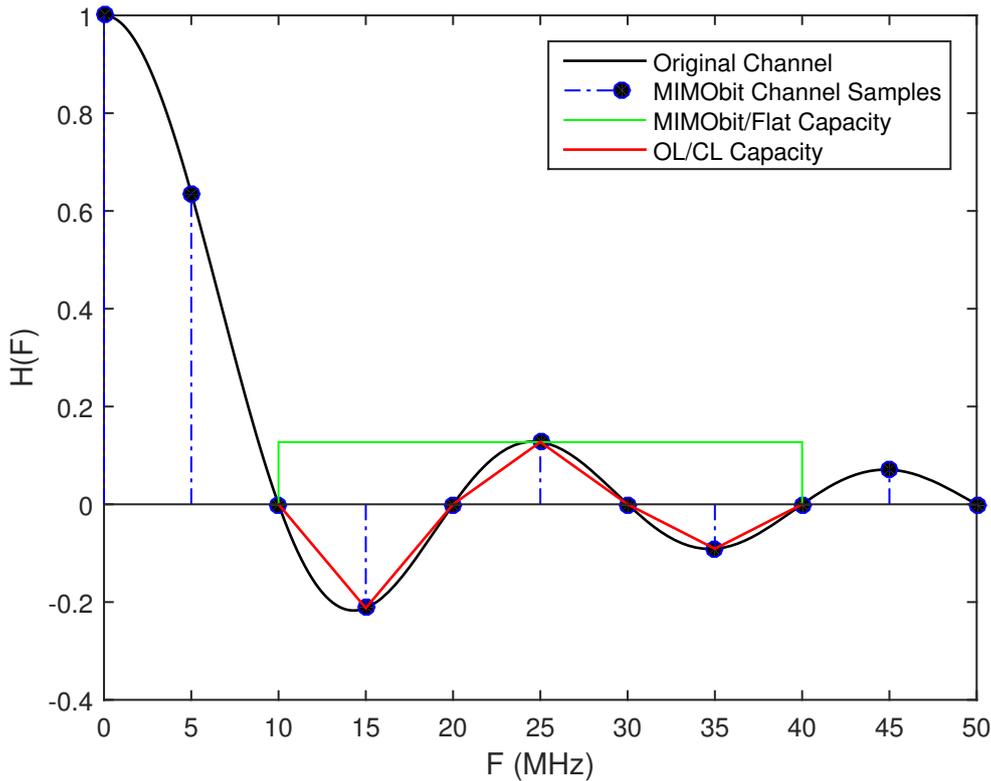
There are two ways to import MIMObit capacity files into TucStats. The first option is to click the “...” button and manually select the desired file. The second option is to have TucStats automatically load MIMObit capacity files each time we select a new simulation file.

To achieve this, we need to create a folder at the same path and with the same name as the corresponding \*.Htotalresults.dat file. Then, we need to save the MIMObit capacity files in this folder and give them the same name as the folder, with the addition of a suffix of the form “<Transmit Bandwidth >MHz\_<Transmit Power >W\_”.

As an example, consider the channel in Figure 3.5 and assume that we need to calculate the capacity for a bandwidth of 30MHz centered at 25MHz. MIMObit and our Flat Channel algorithm are going to use the value of the channel sample at 25MHz for the whole bandwidth of 30MHz. On the other hand, our Open-loop and Closed-loop algorithms utilize all the available samples in the desired bandwidth, and approximate the original channel by performing linear interpolation on these samples.

The “Noise PSD” corresponds to the  $N_0$  parameter in the capacity formulas (see equation (3.81)).

The “Instantiation Domain” group is designed to calculate the capacity of the channel for all instantiations at a specific carrier frequency. Both CDF and PDF can be plotted.



**Figure 3.5:** Channel approximations for a 30MHz bandwidth centered at 25 MHz

The “Carrier Domain” group calculates the capacity at a range of carriers for a specific instantiation. If we specify a range of instantiations, TucStats will plot all of them one after the other; in this case the “Hold on” option should be enabled. Finally, there is the “Averaged” option that calculates the average capacity over the instantiations specified.

Note that the carriers can be set to any frequency, and are not constricted to the discrete set of MIMObit simulation frequencies. This becomes possible owing to the fact that the approximated function for our channel is a continuous function of frequency.

### 3.4.2 Calculations on MIMObit Data

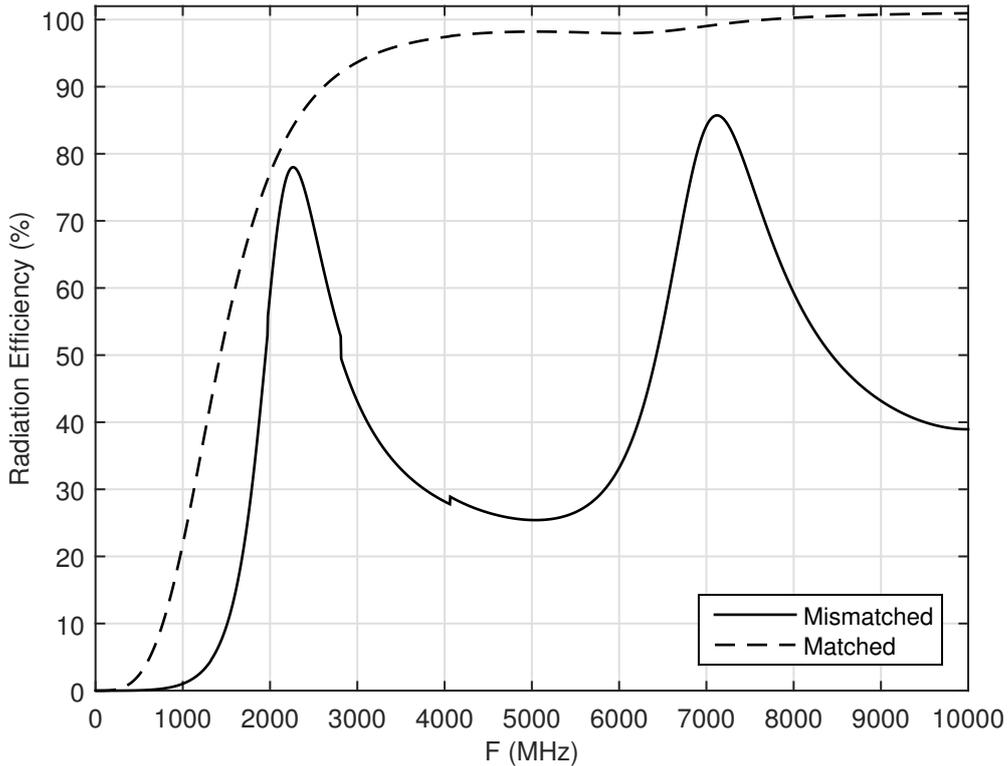
In this subsection we are going to analyze the results that the various types of capacity in TucStats produce for simulated MIMObit channels, and compare them with MIMObit’s respective results.

#### Setup of MIMObit Experiments

First of all, keep in mind that for any MIMObit parameter that is not mentioned in this subsection we are going to imply that it has its default value.

For the antenna setup we used a single-port transmitter and a single-port receiver which are 100 meters away. Also, the transmitter is at a height of 10 meters, while the receiver is at 1.5 meter.

A dipole with a length of 6 centimeters and a diameter of 2 millimeters was installed at the port of both the transmitter and the receiver. The radiation efficiency of this dipole can be seen in Figure 3.6, where we can easily spot two distinct peaks for the mismatched version. Note that these peaks are located approximately at the frequencies that correspond to 0.5 and 1.5 wavelengths of the 6 centimeters half-wavelength dipole. This observation can provide us with useful insight regarding the behavior of the channel capacity as a function of the frequency to which the signal bandwidth is centered.



**Figure 3.6:** Default MIMObit dipole - Length:6cm, Width:2mm

Further, the available power at the transmitter is 1 Watt and all the PDFs and CDFs have been calculated for a signal bandwidth of 20MHz, unless stated otherwise. Note that all the results below correspond to high SNR scenarios.

It is also important to mention that, due to an incomplete understanding on how MIMObit calculates the channel capacity, there has been some experimental work including an attempt to reverse engineer some features of the software. Obviously, making highly educated guesses is the best that we can do in this case. In particular, there have been some attempts to comprehend how thermal noise is modeled in MIMObit and how it can be translated to a potential known and well defined capacity relevant quantity. What we know about noise in MIMObit is first that it is added artificially as white Gaussian noise at the end of the simulation, and second that it is equal to  $kTB$  (see page 43). Taking also into account the fact that MIMObit makes a flat channel assumption, we implemented our own algorithm for flat channel capacity calculation, in order to look for a way to match the results of both algorithms. To this end, we have specified an available power of 1Watt and a signal

bandwidth of 20MHz in both cases, and it turned out that we could obtain a match if we entered a spectral density of  $kT/4$  W/Hz or equivalently  $kT \cdot 10^6/4$  W/MHz in TucStats. A more general formula that has been found to match perfectly in all the tested scenarios is the following

$$N_0 = \frac{k \cdot T \cdot RBW}{4 \min\{W, RBW\}} 10^6 \text{ W/MHz}$$

where  $W$  is the bandwidth of the transmitted signal, and  $RBW$  is the resolution bandwidth of the receiver. In our experiments we have that  $T = 290\text{K}$  and  $RBW = 1\text{MHz}$ . Again, use this formula at your own risk.

We should also mention that throughout the whole experiments process, we ended up having to comply with some trade-offs between frequency samples number, instantiation samples number and total simulation time. For this reason, the number of instantiations for the experiments vary from 10 for simulations with a lot of simulation frequencies, to 500 for simulations with much less frequencies.

Finally, note that we accept a solution of (3.82), only when the integral differs from the total average power,  $\bar{P}$ , by a factor of  $10^{-3}$  or less.

### TucStats vs MIMObit Capacity

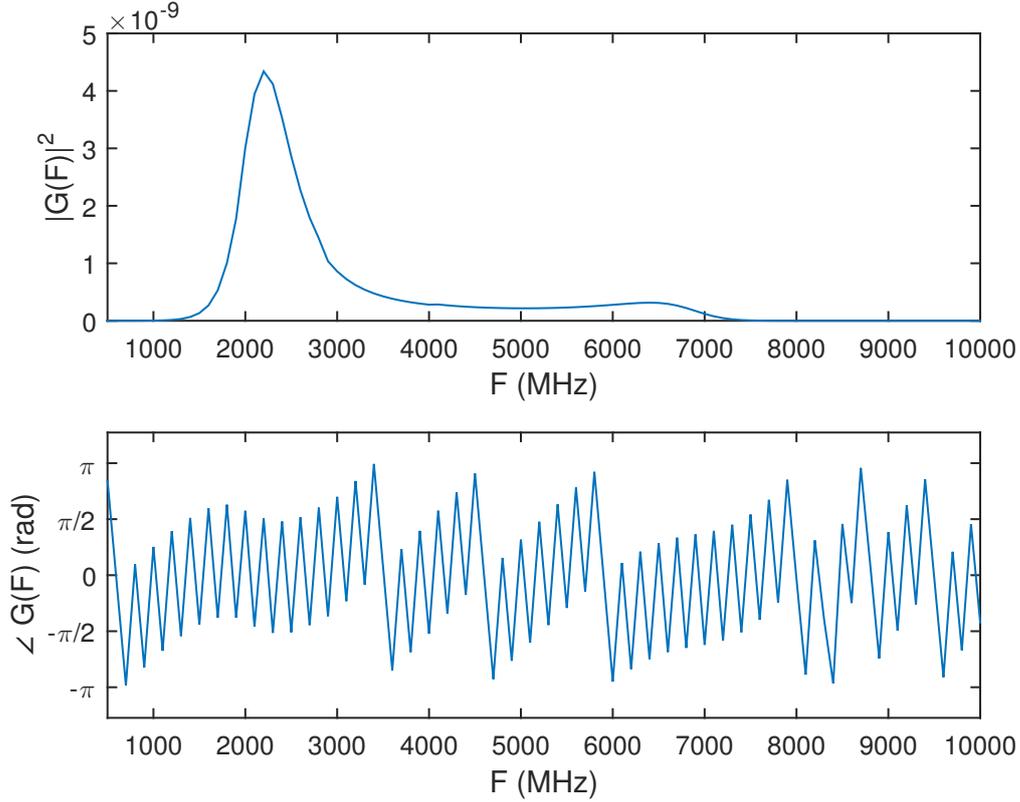
We begin our analysis with the simple scenario of the deterministic Line of Sight propagation environment. First, we can see in Figure 3.7 that the magnitude of the channel frequency response has two peaks at the exact same frequencies as the peaks of the dipole radiation efficiency plot. We may also argue that the reason that the peak at the high frequency is much lower is probably linked to the fact that the free space path loss is inversely proportional to the frequency squared. Free space path loss is generally a multiplicative factor in the received power.

Figures 3.8 and 3.9 provide some insight as to how the capacity behaves as a function of the carrier frequency for various bandwidths. Interestingly, the plots of all the capacity types, including that of MIMObit, coincide for bandwidths as big as even 1000MHz, except for the flat capacity version, which begins to differ for bandwidths greater than 500MHz. This result in essence informs us that the channel can be approximated fairly well as flat for up to 500MHz of bandwidth.

Taking a look now at figures 3.10, 3.11 and 3.12, we can see that the same arguments in general hold for the case of the Flat Earth (aka 2-Ray) propagation environment as well. The most significant discrepancy, compared to the previous environment, is that the flat channel stops being a good approximation for bandwidths greater than 128MHz. It is also noteworthy that the wavy pattern in the channel frequency response and the capacities is due to the constructive and destructive interference of the two rays at the receiver.

We now continue to the probabilistic New Radio UMi propagation environment. MIMObit has implemented three versions of this environment; the Line of Sight (LOS), the Non Line of Sight (NLOS) and the Random. The latter just produces randomly LOS and NLOS instantiations with a probability that depends on the distance between the transmitter and the receiver.

Beginning with the NLOS version, we can find in figures 3.13 and 3.14 the channel frequency response for the first instantiation and for the average over all instantiations, respectively. Again, we can distinguish the two peaks as in the previous propagation environments.



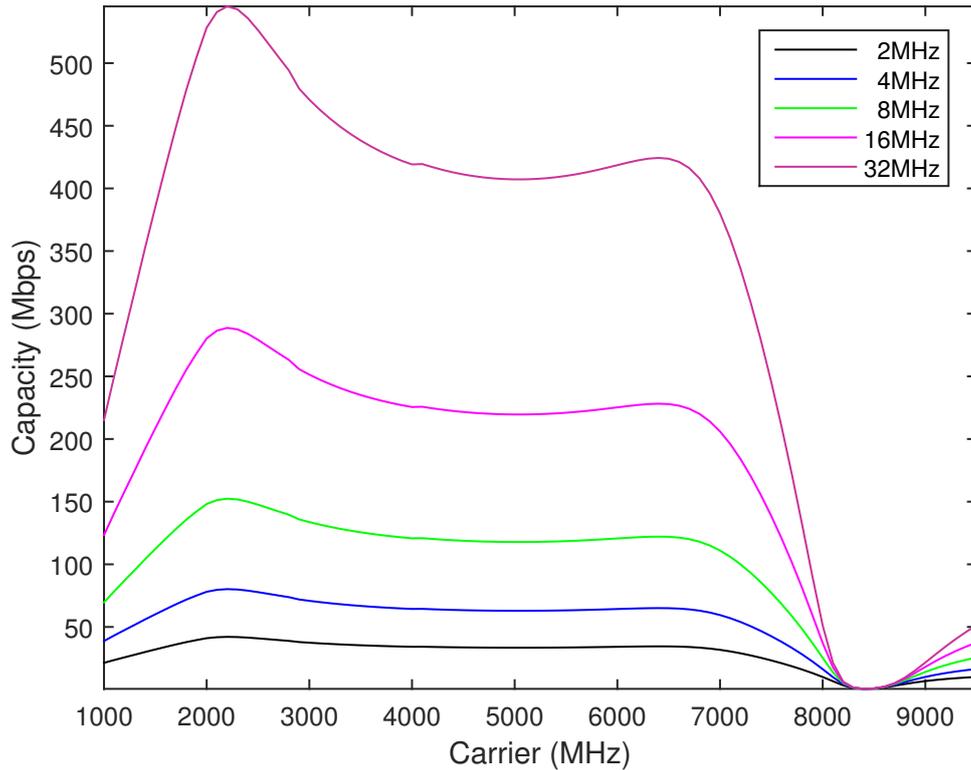
**Figure 3.7:** Line of Sight - Channel frequency response

Since, this environment has a probabilistic nature, it would be interesting to study the statistics of the various capacity types. First of all, it is important to remember that the density of the simulation frequencies in a specific bandwidth can have a severe impact on the variance of the open-loop and closed-loop capacities, as we can see in Figure 3.15. Specifically, we observe that as the number of samples in this bandwidth gets larger, then the variance of the capacity is reduced, while the sample mean remains the same. Note that from now on we will only consider the channel with the most frequency samples.

Using a bandwidth of 20MHz at a carrier frequency of 2400MHz, we can obtain the capacity CDFs that we see in Figure 3.16. Notice how the MIMObit capacity plot coincides with the flat capacity plot, and the open-loop capacity plot coincides with the closed-loop capacity plot. This behavior actually holds for various bandwidths as we show in Figure 3.17, where we can also see that as the bandwidth gets larger then the variance of the MIMObit capacity grows exponentially faster than the variance of the closed-loop capacity (notice that the x-axis has a logarithmic scale). Hence, we conclude that we should be really careful when we employ a flat channel approximation, since the real capacity variance is usually a lot smaller, and the error is more noticeable at larger bandwidths.

By inspecting figures 3.18 to 3.22 we can conclude that the same arguments hold for the LOS version of the model as well.

Again, figures figures 3.24 to 3.26 and 3.28 lead us to the same conclusions for the Random version of the model. In addition to these conclusions, we can also clearly see in Figure 3.27 that making a flat channel assumption not only results to

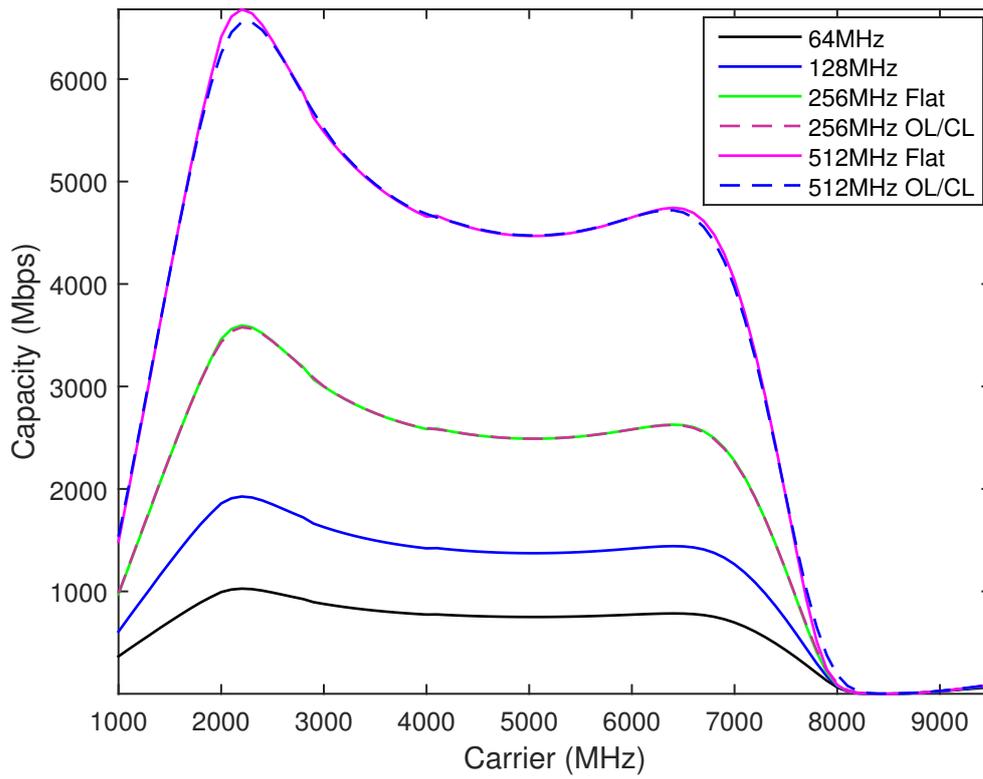


**Figure 3.8:** Line of Sight - Capacity for various bandwidths - Plots coincide for all capacity types

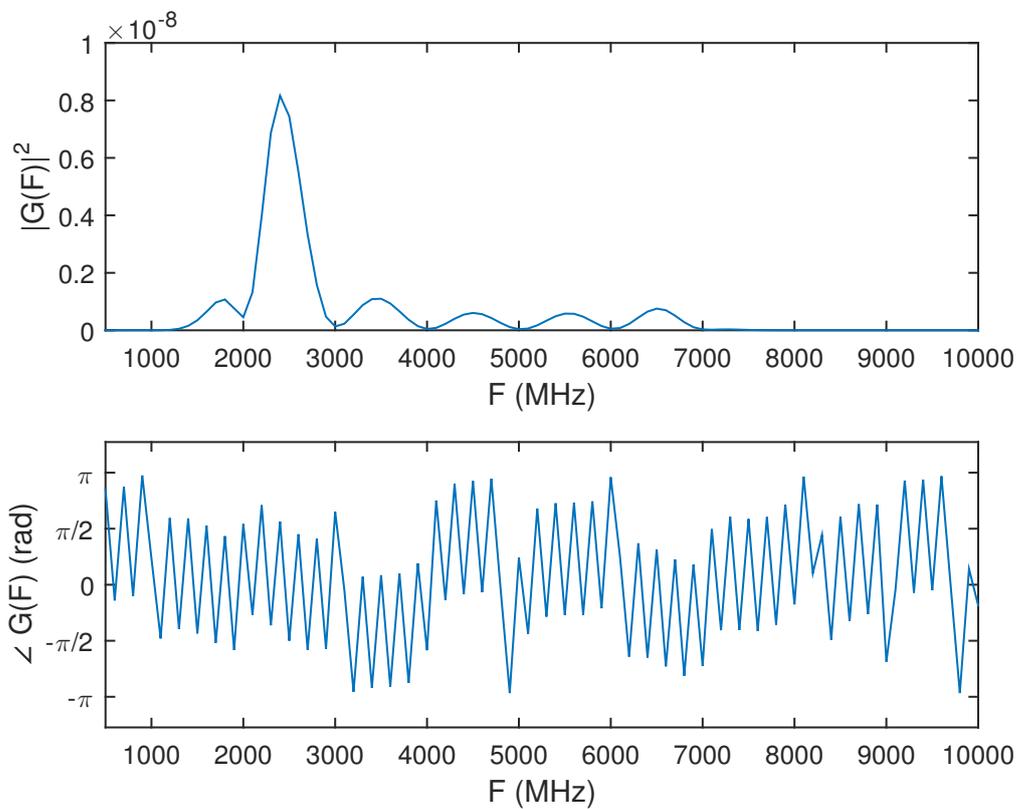
an erroneous variance in this case, but the PDF of the capacity will in fact consist of two distinct masses. This is completely different from the closed-loop capacity where only one mass is produced.

In summary, in Figure 3.29 we make a brief comparison of all three versions of the New Radio UMi propagation environment, whereas in Figures 3.30 and 3.31 we provide the same information for two more New radio variants, the UMA and the RMA. Note that, in general, the same arguments hold for these two variants as well.

Before we conclude our analysis, it would be interesting to observe that, even though our 6 centimeters dipole would in practice be used for signal transmission in a narrow bandwidth centered at about 2100-2500 MHz, it turns out that the capacity in all the propagation environments above is surprisingly not reduced dramatically for carriers of up to about 8000MHz!



**Figure 3.9:** Line of Sight - Capacity for various bandwidths - Plots coincide for all capacity types with up to about 512MHz of bandwidth



**Figure 3.10:** Flat Earth - Channel frequency response

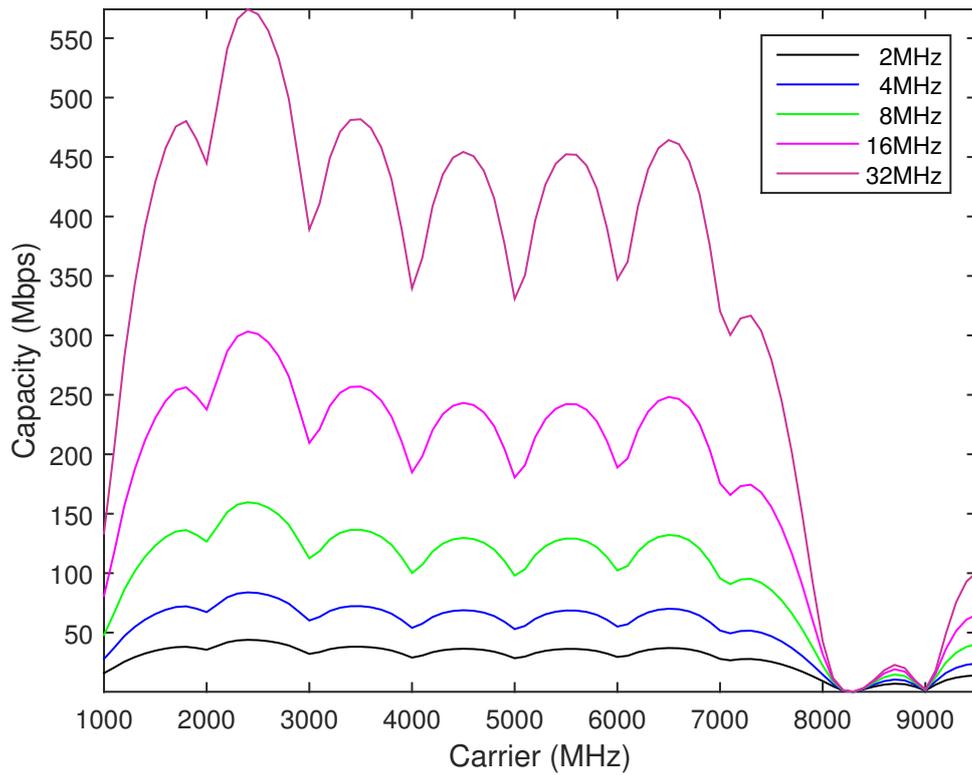


Figure 3.11: Flat Earth - Capacity for various bandwidths - Plots coincide for all capacity types

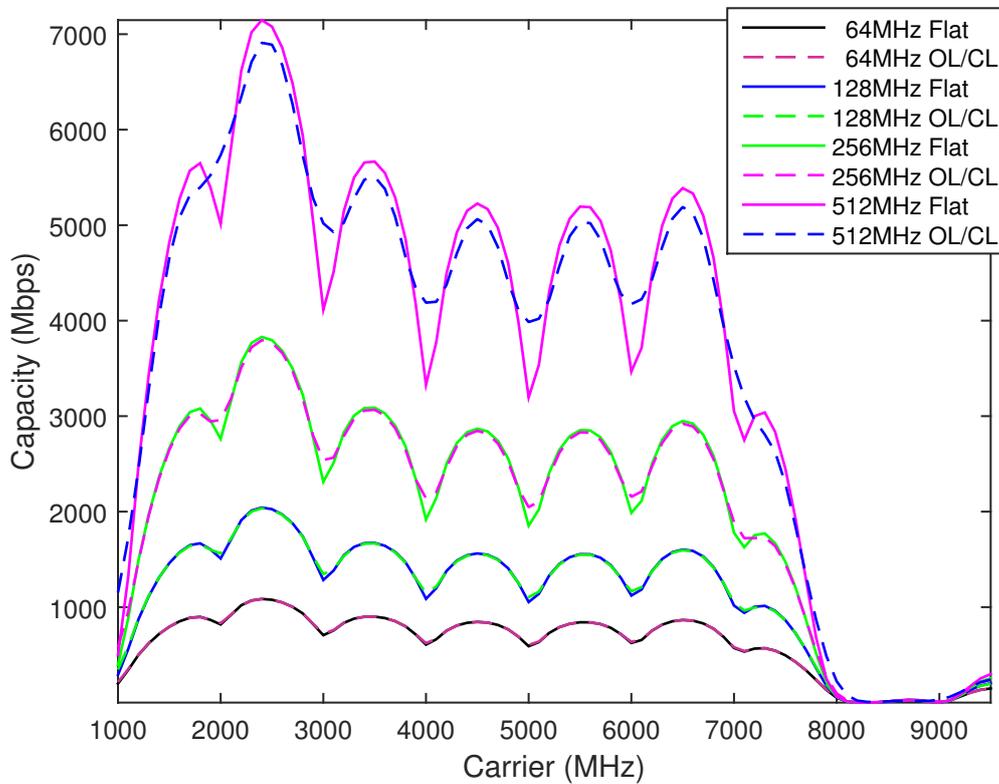
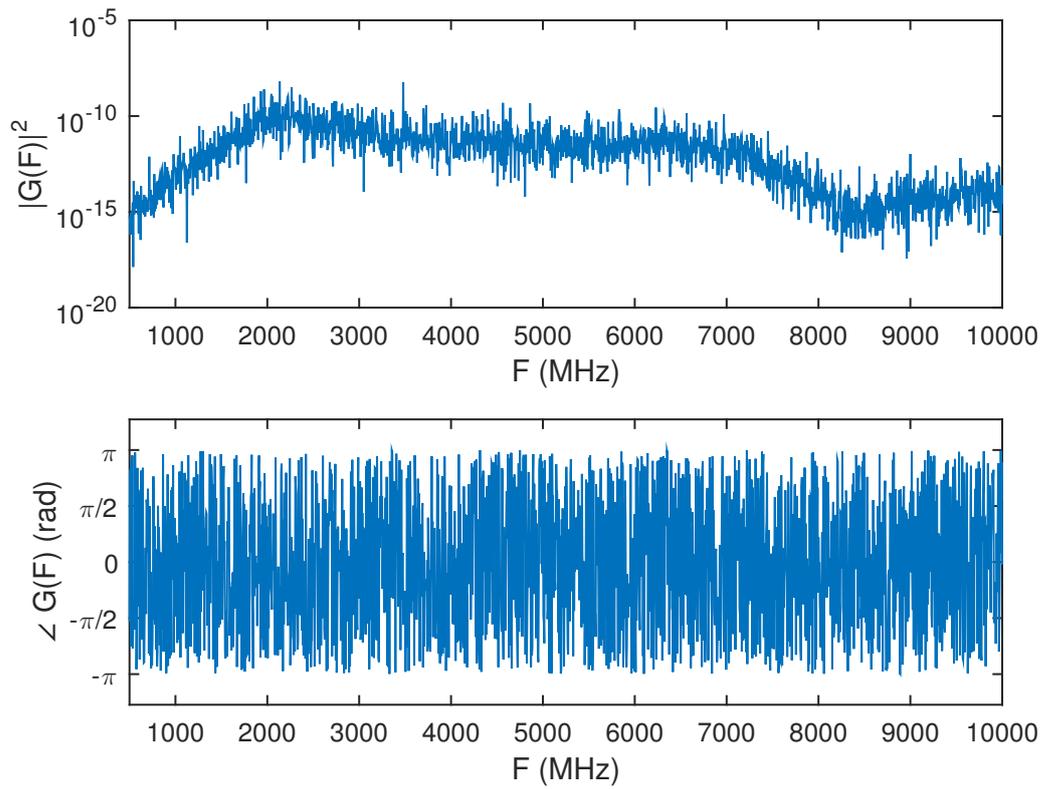
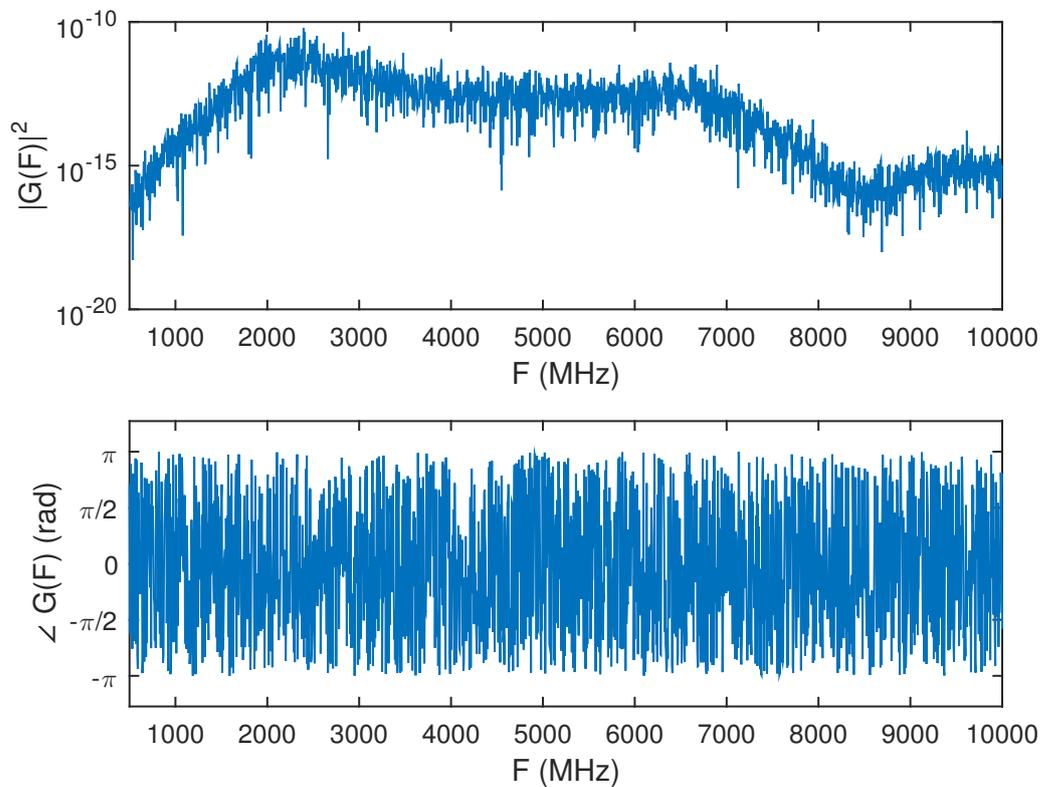


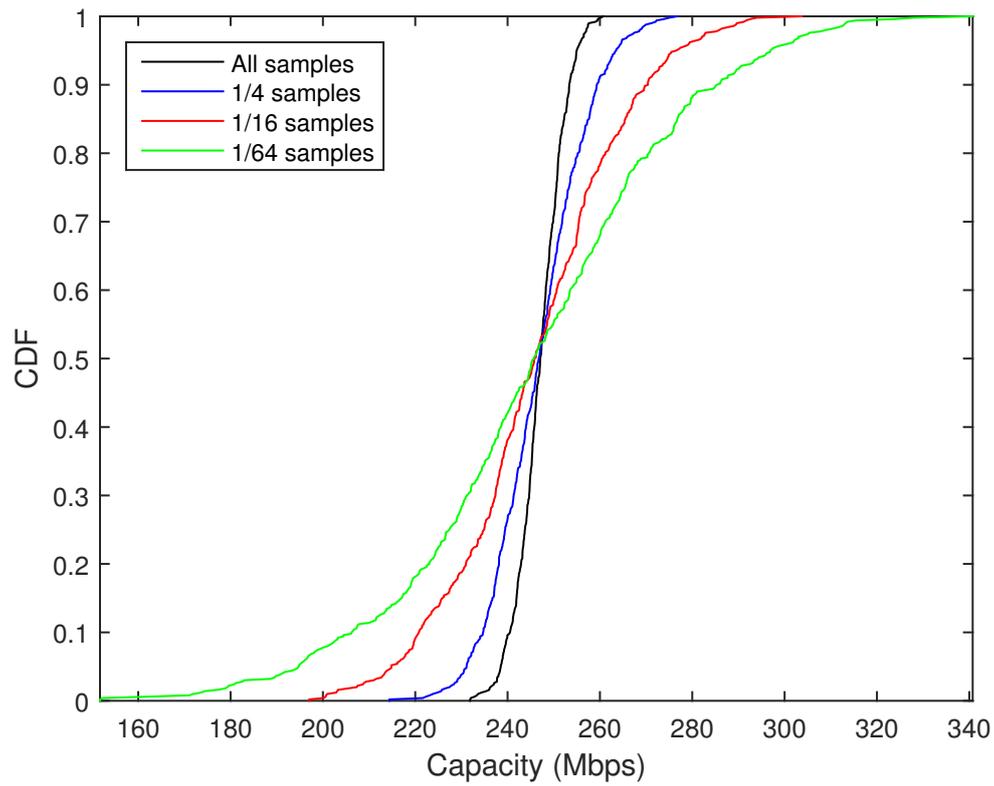
Figure 3.12: Flat Earth - Capacity for various bandwidths - Plots coincide for all capacity types with up to about 128MHz of bandwidth



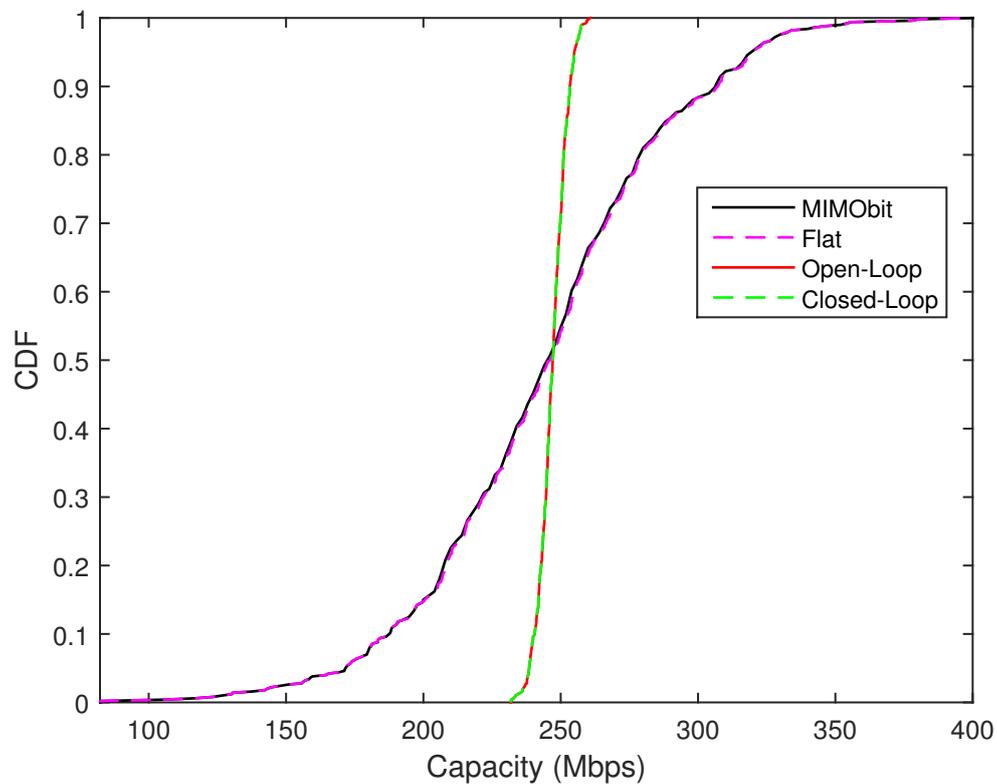
**Figure 3.13:** New Radio UMi NLOS - Channel frequency response - First instantiation



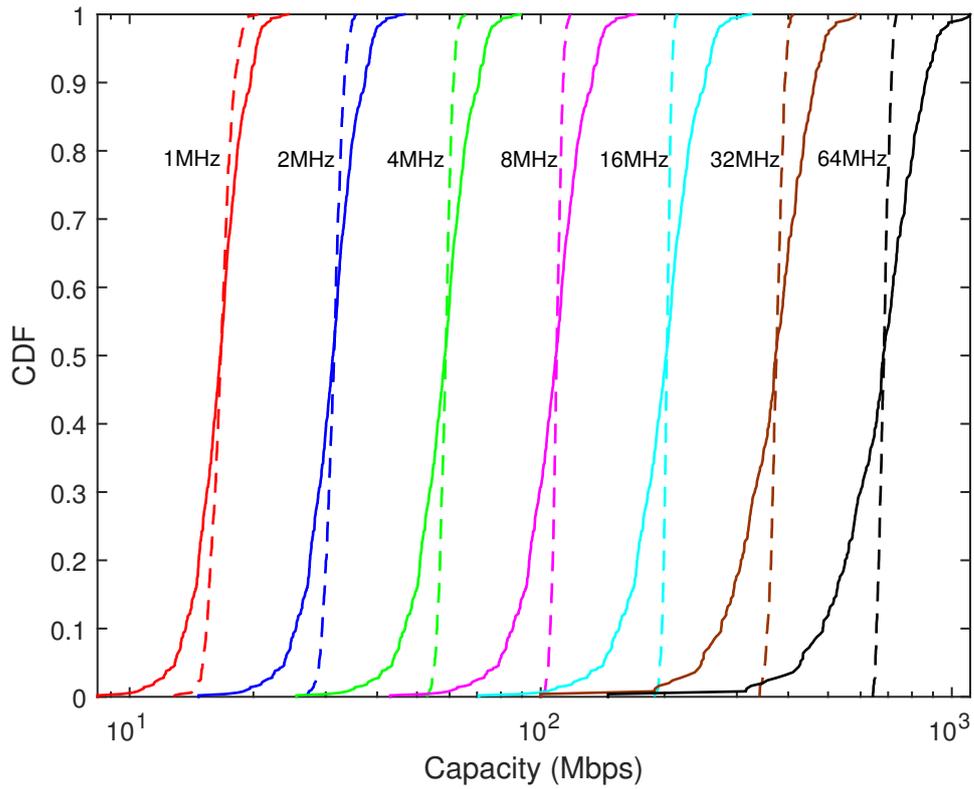
**Figure 3.14:** New Radio UMi NLOS - Averaged channel frequency response - 30 samples



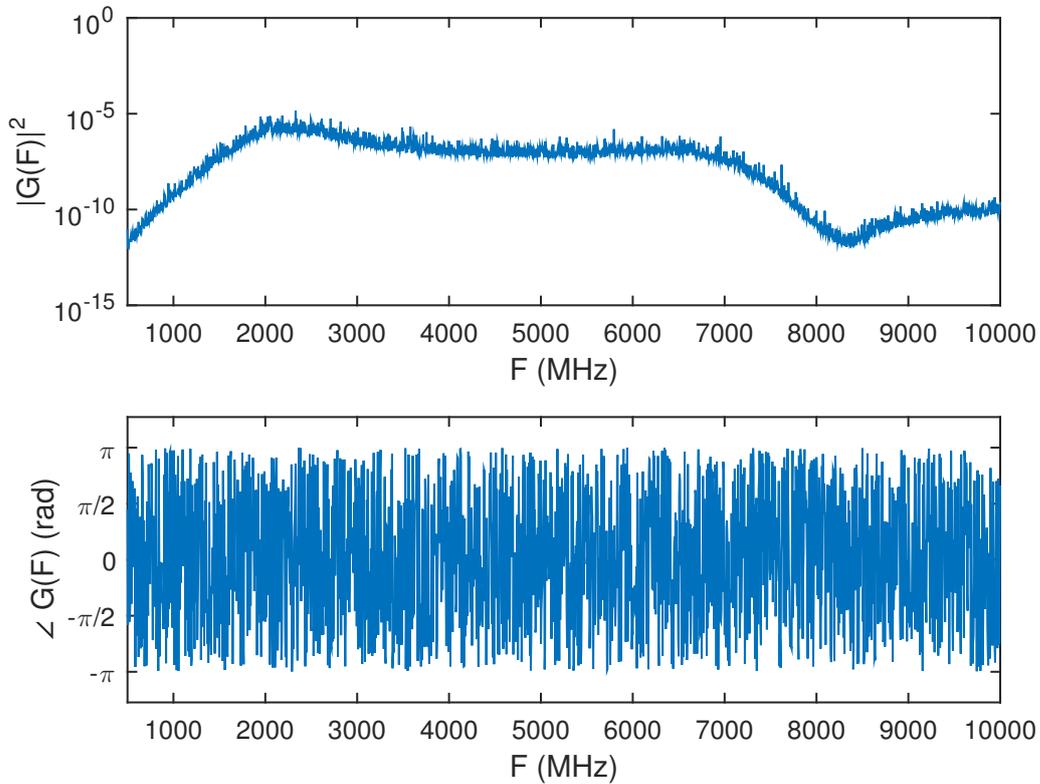
**Figure 3.15:** New Radio UMi NLOS - Open-loop/Closed-loop capacity for various channel sampling rates



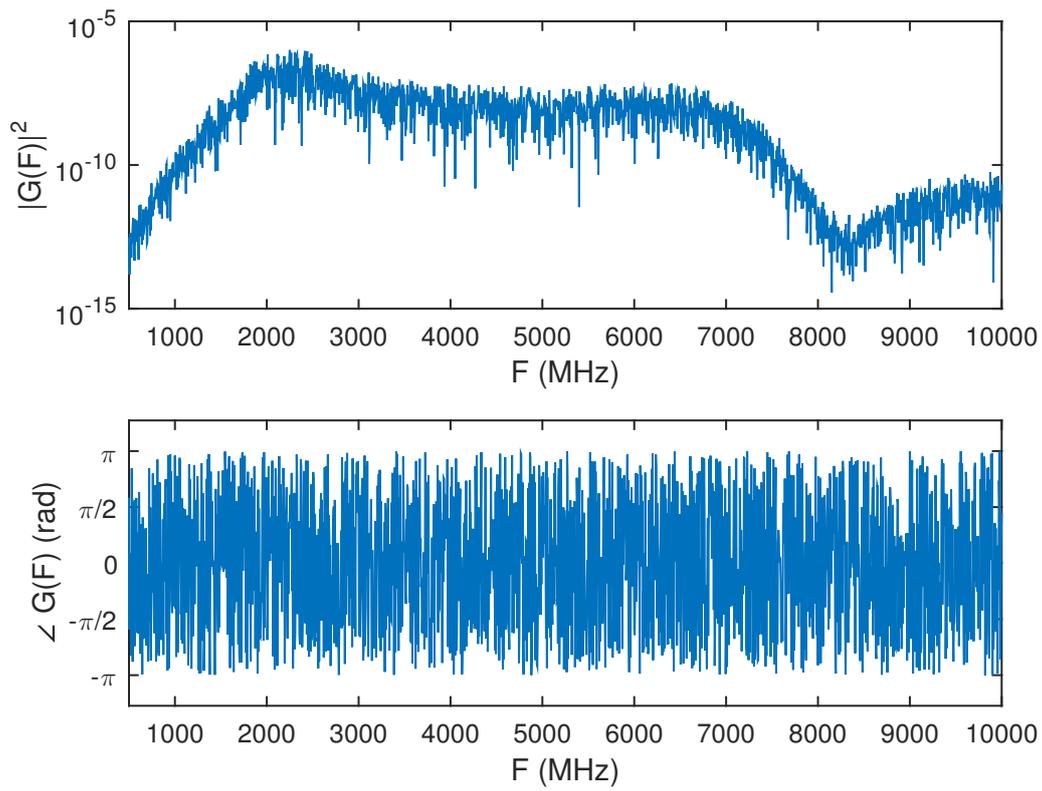
**Figure 3.16:** New Radio UMi NLOS - Capacity types comparison



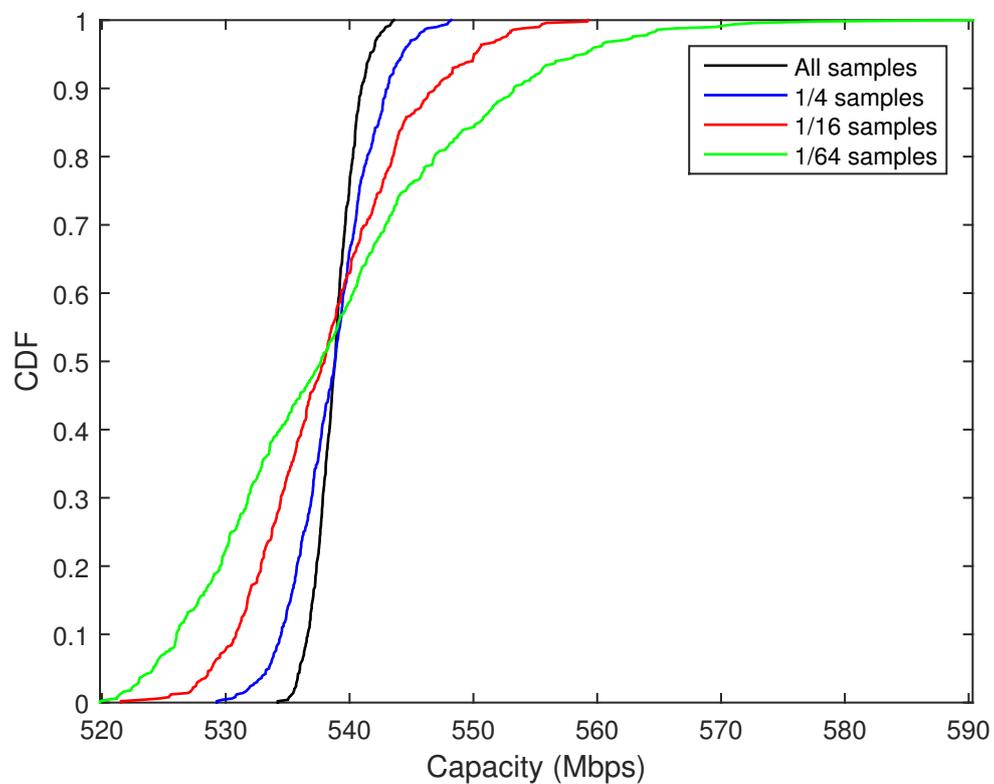
**Figure 3.17:** New Radio UMi NLOS - Capacity types comparison for various bandwidths - solid line:MIMObit/Flat, dashed line:Open-loop/Closed-loop



**Figure 3.18:** New Radio UMi LOS - Channel frequency response - First instantiation



**Figure 3.19:** New Radio UMi LOS - Averaged channel frequency response - 10 samples



**Figure 3.20:** New Radio UMi LOS - Open-loop/Closed-loop capacity for various channel sampling rates

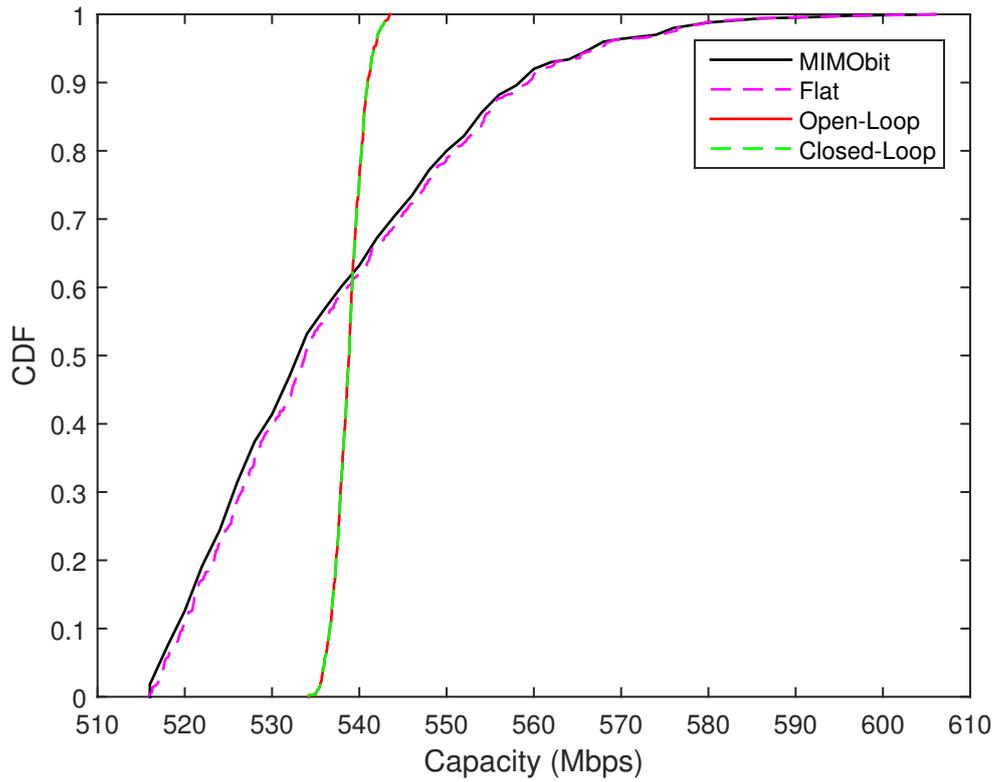


Figure 3.21: New Radio UMi LOS - Capacity types comparison

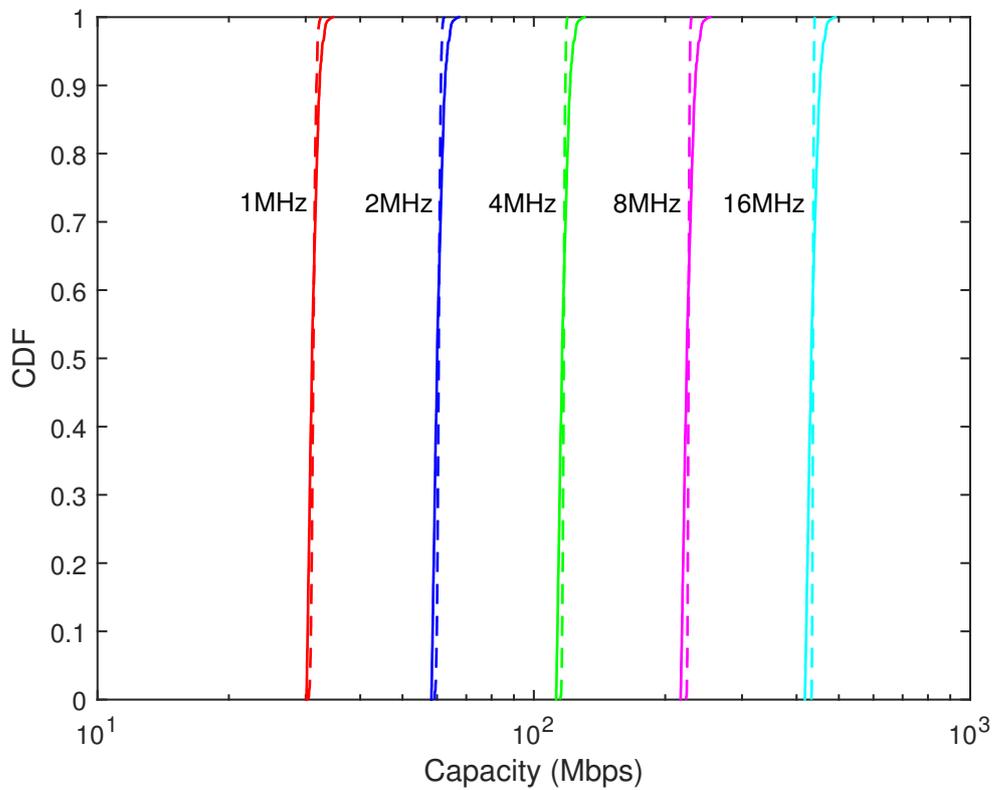
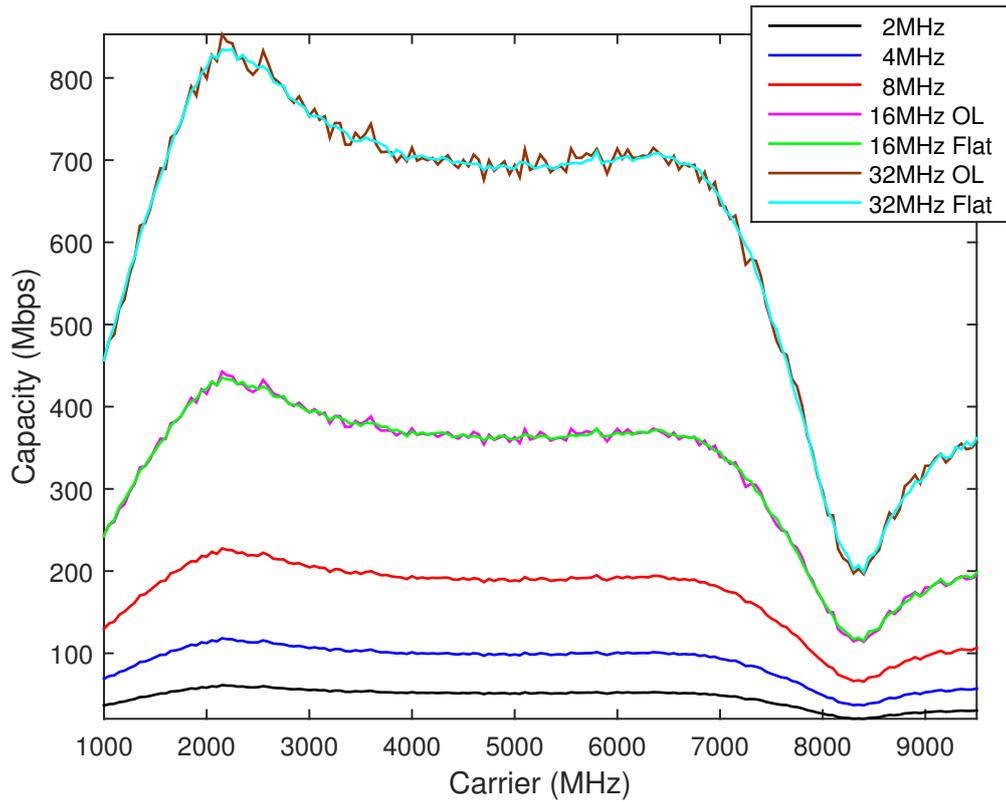
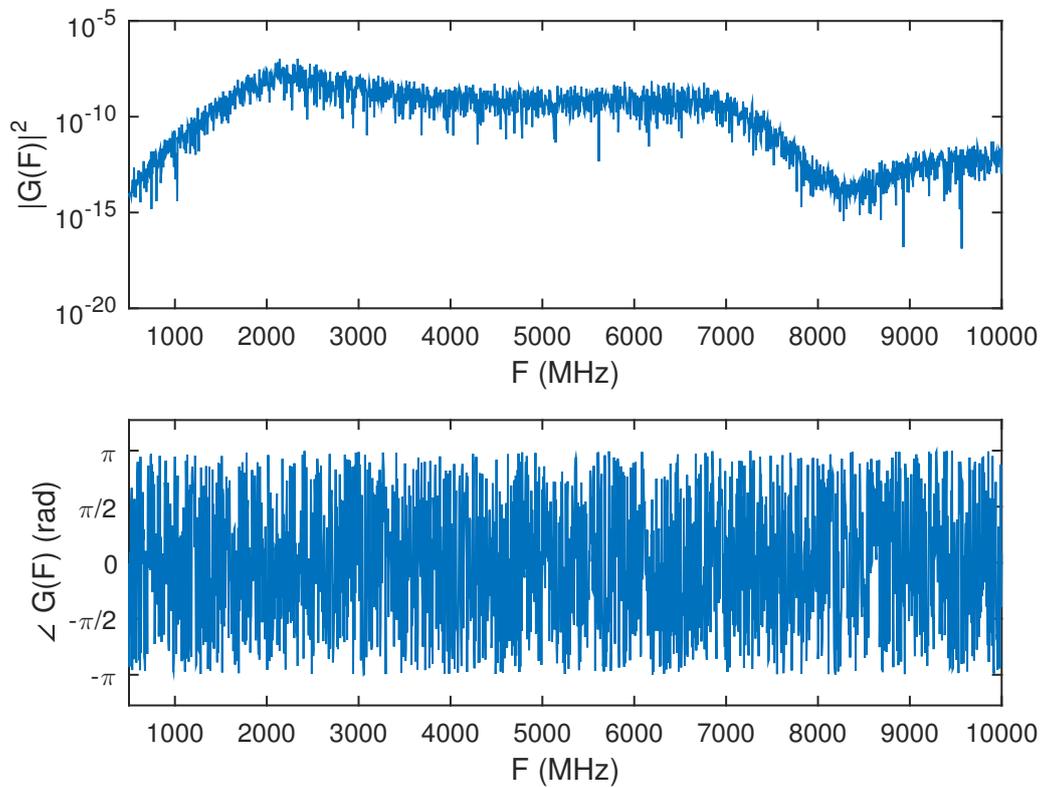


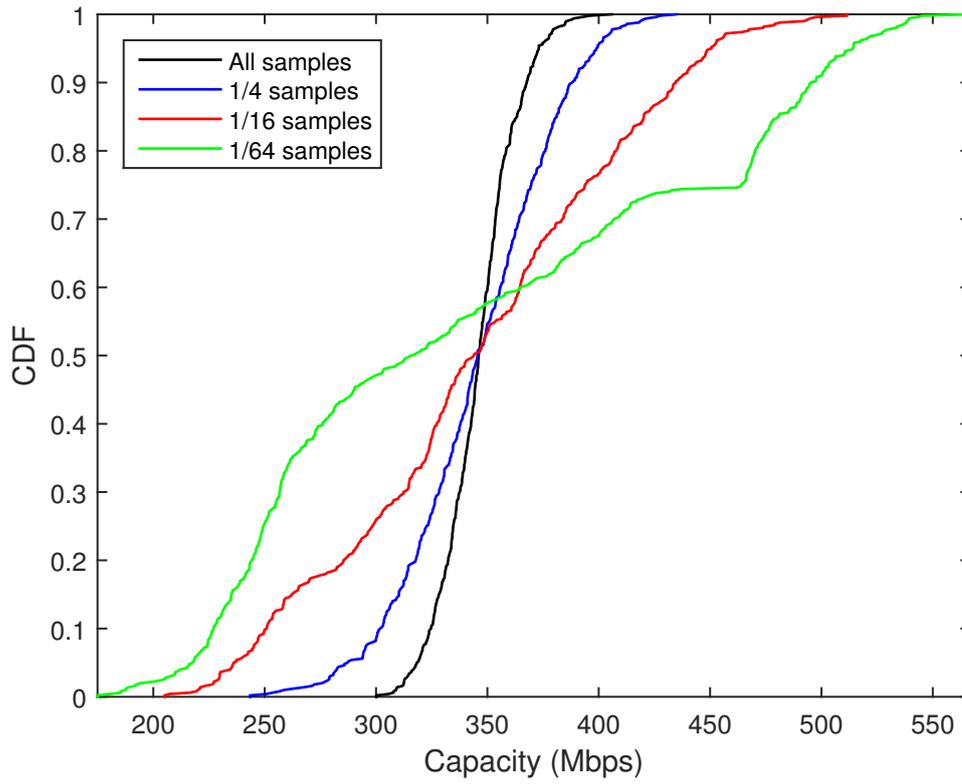
Figure 3.22: New Radio UMi LOS - Capacity types comparison for various bandwidths - solid line:MIMObit/Flat, dashed line:Open-loop/Closed-loop



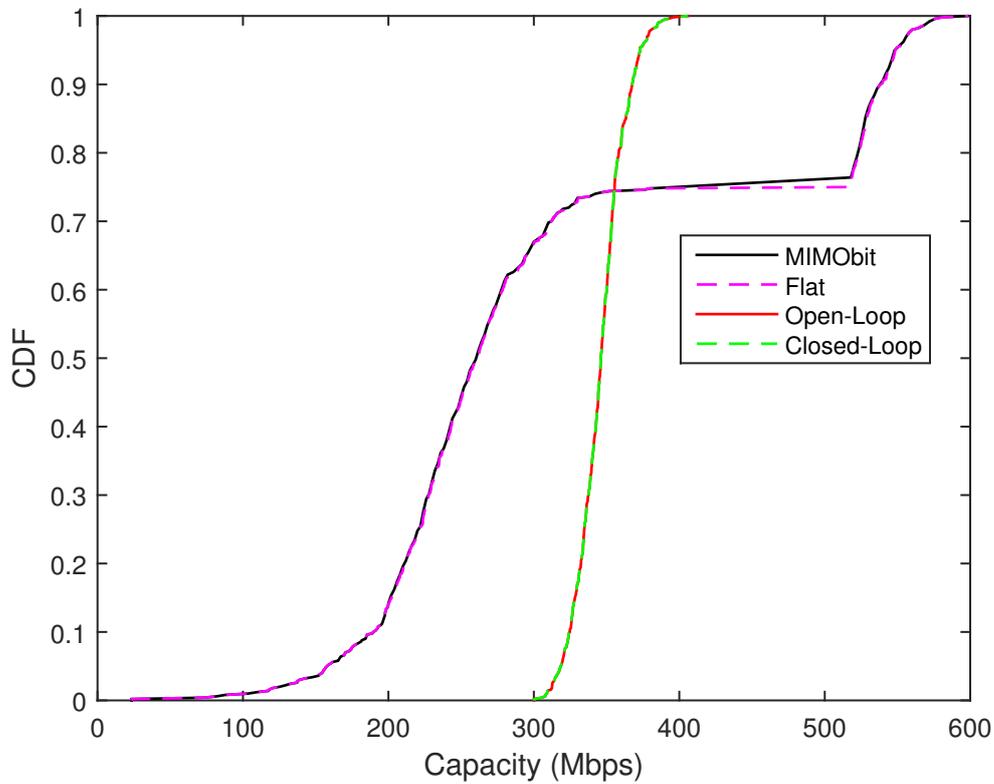
**Figure 3.23:** New Radio UMi LOS - Averaged capacity for various bandwidths - Plots coincide for MIMObit, flat and open-loop capacities with up to about 8MHz of bandwidth - 10 samples



**Figure 3.24:** New Radio UMi Random - Averaged channel frequency response - 30 samples



**Figure 3.25:** New Radio UMi Random - Open-loop/Closed-loop capacity for various channel sampling rates



**Figure 3.26:** New Radio UMi Random - Capacity types comparison

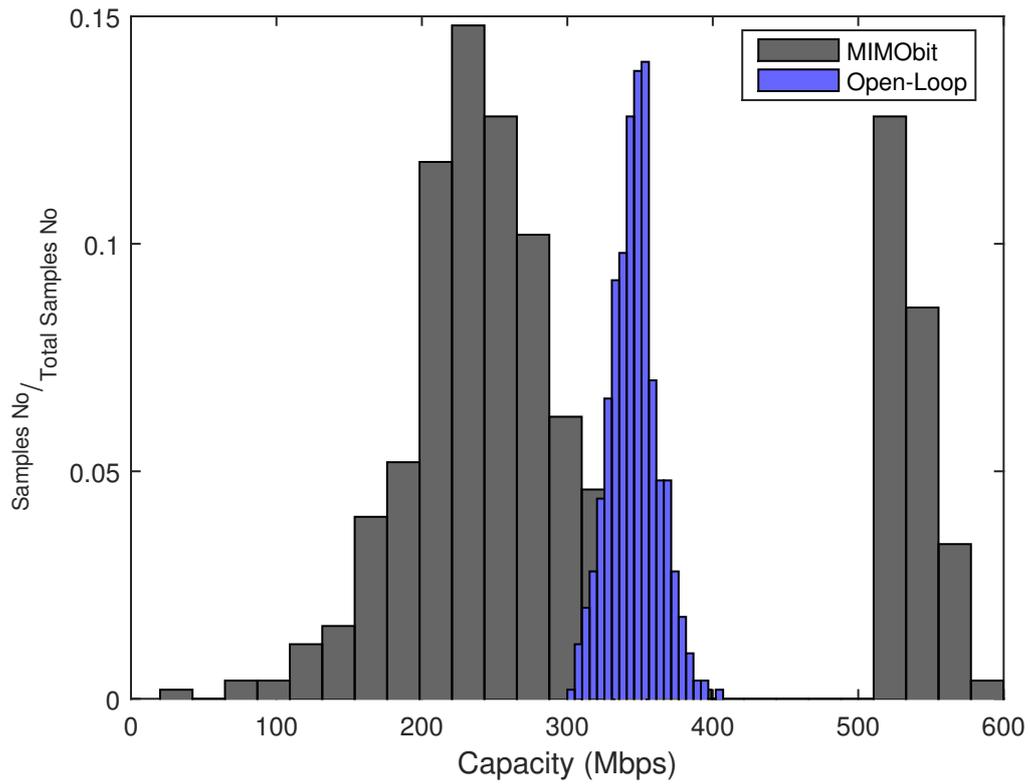


Figure 3.27: New Radio UMi Random - Capacity types comparison

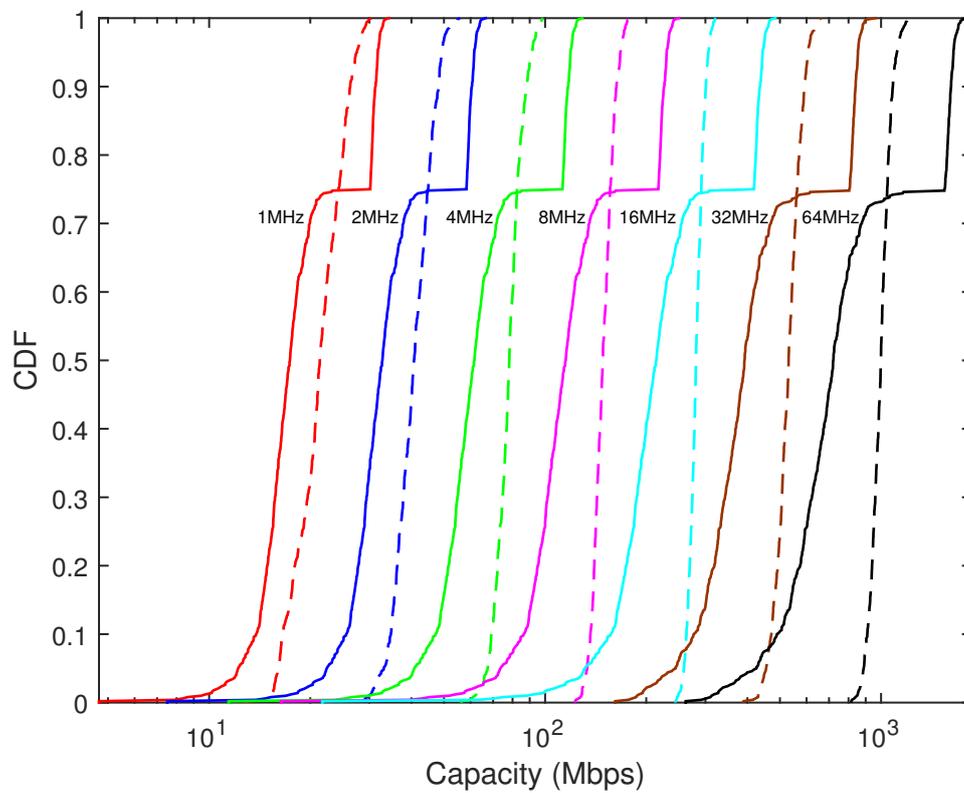


Figure 3.28: New Radio UMi Random - Capacity types comparison for various bandwidths - solid line:MIMObit/Flat, dashed line:Open-loop/Closed-loop

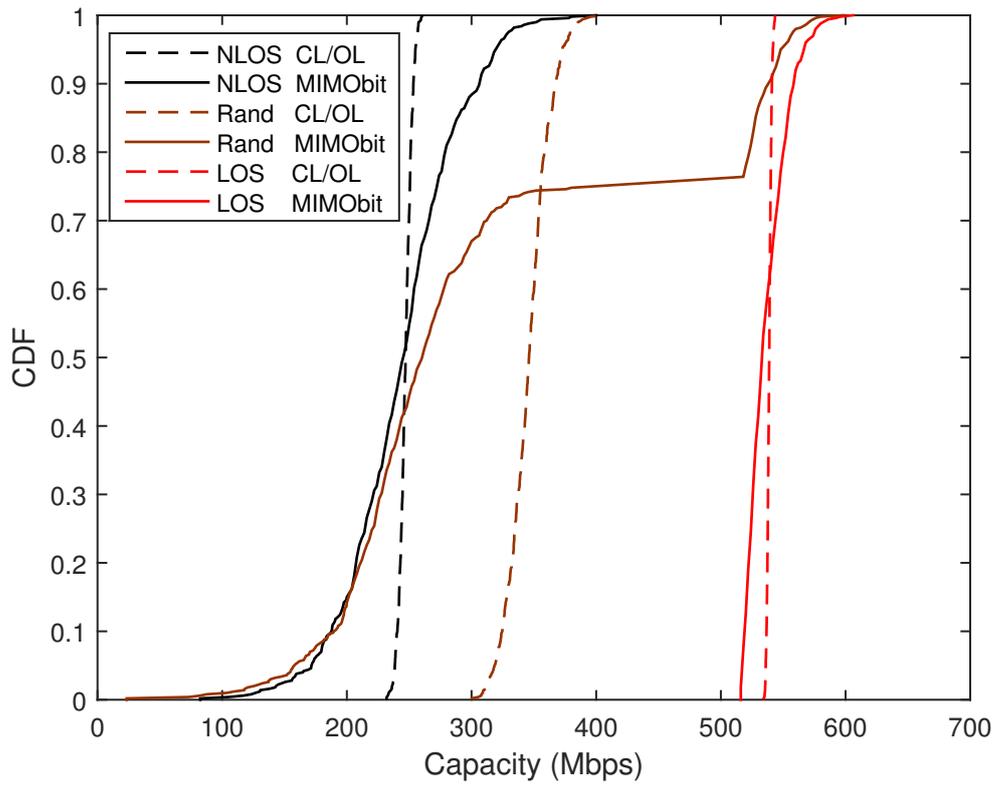


Figure 3.29: New Radio UMi - Capacity types comparison

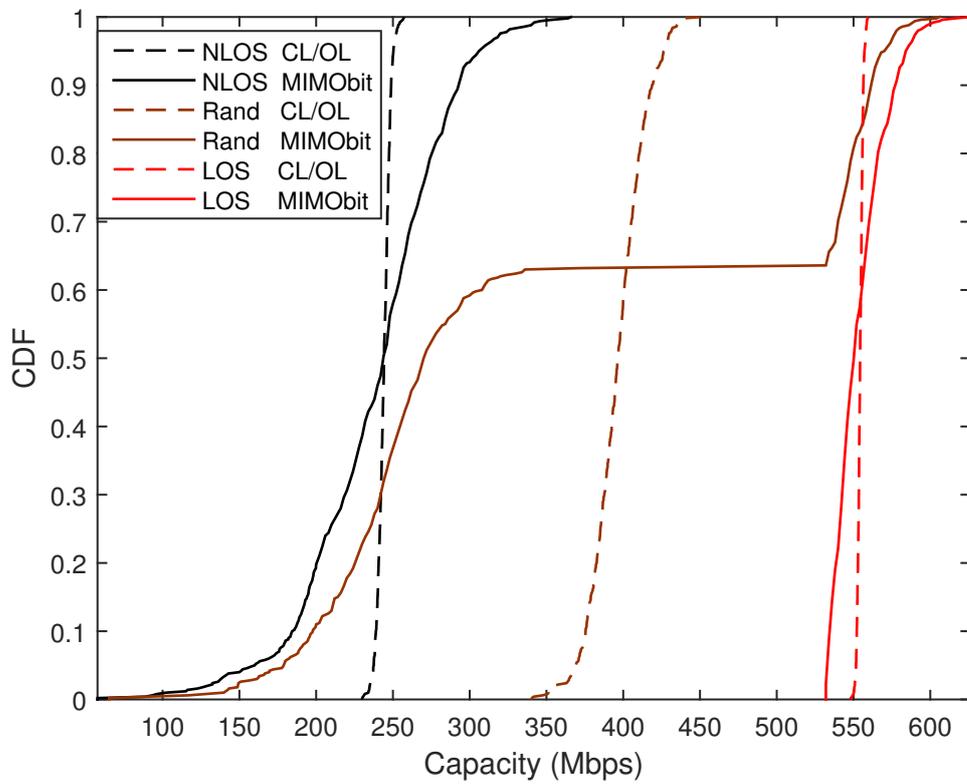
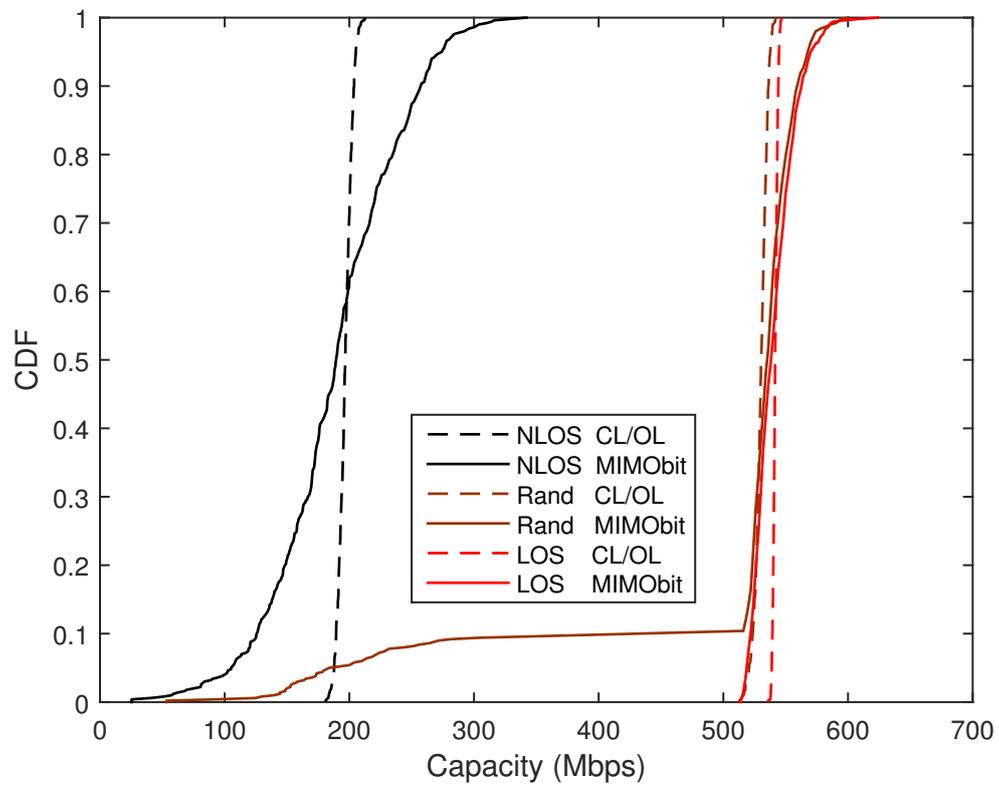


Figure 3.30: New Radio UMA - Capacity types comparison



**Figure 3.31:** New Radio RMA - Capacity types comparison

# Chapter 4

## Conclusion And Future Work

In this project, we have made an effort to explore MIMObit's capabilities and discover a connection between the world of physical antenna systems and the abstract models of wireless communications. To this end, we have developed TucStats, which can provide the user with an intuitive and efficient way to analyze MIMObit generated data. TucStats main focus is on SISO channel capacity calculations, although it offers the ability to study the behavior of the channel itself too.

We have discovered various interesting results regarding the capacity of the channel when employing the Line of Sight, the Flat Earth (2-Ray) and the New Radio UMi propagation environments. Specifically, we show that in simulated antenna systems at high SNRs using roughly up to 100MHz of bandwidth for transmission, there is no remarkable difference when the transmitter has knowledge about the channel and/or when water-filling is applied for optimal power distribution. Further, we show that MIMObit's flat channel assumption over the transmission bandwidth can produce inaccurate capacity calculations, with the error increasing exponentially as the transmission bandwidth also increases. Finally, it is noteworthy that when the probabilistic propagation environment New Radio UMi is employed, then the closed-loop capacity of the channel tends to become deterministic as we consider a better approximation of the channel.

Despite these findings, however, in this work we have explored and analyzed only a small portion of the wireless communication scenarios that MIMObit can simulate. Most importantly, TucStats can be upgraded to support MIMO scenarios while utilizing simulated versions of antennas and propagation environments that are commonly used in practice. Finally, it would be of interest to study how these channels behave in low SNRs, and at what extend our results hold true in this case as well.

# Bibliography

- [1] Peter Joseph Bevelacqua. *Antennas and Antenna Theory*. 2009. URL: <http://www.antenna-theory.com>.
- [2] Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2004.
- [3] Kenneth R. Demarest. *Circuit Analysis using Phasors, Laplace Transforms, and Network Functions*. URL: <https://people.eecs.ku.edu/~demarest/212/Phasor%20and%20Laplace%20review.pdf>.
- [4] Richard Fitzpatrick. *Electromagnetism and Optics*. 2007. URL: <http://farside.ph.utexas.edu/teaching/3021/lectures/lectures.html>.
- [5] G.J. Foschini and M.J. Gans. “On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas”. In: *Wireless Personal Communications* 6 (1998), pp. 311–335.
- [6] Andrea Goldsmith. *Wireless Communications*. 2004.
- [7] Sean Victor Hum. *Noise in Radio Systems*. 2016. URL: <http://www.waves.utoronto.ca/prof/svhum/ece422/notes/21-noise.pdf>.
- [8] Upamanyu Madhow. *Fundamentals of Digital Communication*. Cambridge University Press, 2008.
- [9] Kirk T. McDonald. *Maximum Power from DC Current and Voltage Sources*. 2013. URL: <http://www.physics.princeton.edu/~mcdonald/examples/maxpow.pdf>.
- [10] LLC NEBENS. *MIMObit 2 User’s Manual*. 2017.
- [11] Nikolaos Papamarkos. *Ανάλυση Ηλεκτρικών Κυκλωμάτων*. Β. Γκιούρδας Εκδοτική.
- [12] Marvin K. Simon and Mohamed-Slim Alouini. *Digital Communication over Fading Channels*. John Wiley & Sons, Inc., 2005.
- [13] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.