Diogo José Fernandes Gonçalves

Licenciatura em Ciências da Engenharia Eletrotécnica e de Computadores

Wideband LNA optimal sizing using an Optimization Platform, based on Cadence

Dissertação para obtenção do Grau de Mestre em

Engenharia Eletrotécnica e de Computadores

Orientador: Rui Manuel Leitão Santos Tavares, Professor Auxiliar, FCT-UNL

Júri

Presidente: Doutor João Almeida das Rosas
Arguente: Doutor Luís Augusto Bica Gomes da Silva
Vogal: Doutor Rui Manuel Leitão Santos Tavares

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2019

Wideband LNA optimal sizing using an Optimization Platform, based on Cadence

Copyright © Diogo José Fernandes Gonçalves, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.
A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "novathesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Para a minha família e amigos.

Acknowledgements

First of all, I would like to thank the Department of Electrical Engineering of FCT-UNL for the necessary resources in order to develop the work.

I would like to thank to my advisor, Prof. Rui Tavares, for providing the project and all the guidance through the project development. I am also grateful for the support, advice and knowledge provided by Prof. Luís Oliveira and Miguel Santos.

I am also very grateful to my former colleagues Diogo Coelho, Luís do Ó, Fábio Dias and João Fernandes for sharing their knowledge and providing support during our journey.

Finally i would like to thank my family, friends and girlfriend for all their patient and good environment throughout this journey.

Abstract

In this thesis an Optimization Platform capable of guaranteeing a proper sizing for a Low Noise Amplifier, is proposed. The developed platform, which runs in a python client, contains a Genetic Algorithm capable of managing the circuits' components dimensions, to fulfill the necessary requirements imposed by high- frequency architectures. To improve the results, the Cadence simulator is used, guaranteeing a more accurate simulation process when compared to traditional sizing methods.

The Optimization Platform contains an interface, which provides different optimization methods, each one with different approaches to provide the circuits' optimal sizing.

Keywords: CMOS Wideband LNA, Noise and Distortion cancelling, Evolutionary Algorithms, Genetic Algorithm, Optimization Platform for electronic circuits.

Resumo

Nesta tese será proposta uma plataforma de otimização, capaz de garantir um dimensionamento aceitável para um Amplificador de Baixo Ruído. A plataforma desenvolvida, desenvolvida num ambiente Python, contém um Algoritmo Genético capaz de gerir as dimensões dos componentes dos circuitos implementados, por forma a garantir os requisitos associados às arquiteturas de alta-frequência.

Para que seja possível obterem-se resultados mais precisos, quando comparados com métodos de otimização tradicionais, é utilizado o otimizador do Cadence no processo de simulação.

A plataforma de otimização contém uma interface, que dispõe ao utilizador diferentes métodos de otimização, por forma a garantir um dimensionamento óptimo dos componentes dos circuitos.

Keywords: CMOS Wideband LNA, Noise and Distortion cancelling, Evolutionary Algorithms, Genetic Algorithm, Optimization Platform for electronic circuits.

Contents

List of Figures

List of Tables

Acronyms

CG Common-Gate.

CIW Command Interpreter Window.

CMOS Complementary Metal Oxide Semiconductor.

CS Common-Source.

DEAP Distributed Evolutionary Algorithms in Python.

EA Evolutionary Algorithm.

EC Evolutionary Computing.

FOM Figure of Merit.

GA Genetic Algorithm.

IIP2 Second-order Input Referred Intercept Point.

IIP3 Third-order Input-referred Intercept Point.

IM3 Third-order Intermodulation Product.

IMN Input Matching Network.

IP3 Third-order Intercept Point.

LNA Low Noise Amplifier.

NF Noise Figure.

NSGA Non-dominated Sorting Genetic Algorithm.

OIP3 Third-order Output-referred Intercept Point.

OMN Output Matching Network.

PSO Particle Swarm Optimization.

SOCAD Sockets Communication for Cadence program.

UML Unified Modeling Language.

Chapter 1

Introduction

1.1 Background and Motivation

The technology and electronics devices, in our days, play a huge role in the society. Companies are starting to develop and improve their own systems by reducing their costs, power consumption and spent time, in order to achieve the society's needs.

Since circuits have improved their complexity, the search for other methods of optimization have started to become more and more present with the improvement of the technology. Traditional methods, require bigger costs and time to the companies when compared with new unconventional methods.

Low Noise Amplifiers play a huge role in the electronic systems and are widely used in receivers architectures capable of amplifying weak signals, which ensure the society's global communications. This kind of amplifiers need to be low power, which is becoming a major issue when more components are added to their architectures. Although considered simple circuits, their sizing need to address multiple variables, where computer based algorithms can be a proper solution to achieve the optimal circuit sizing.

Evolutionary computing has also been an area of huge improvement due to the development of computers and algorithms that are able to solve more complex real world problems. Having this powerful tool, the industries are starting to apply it to their own systems.

Considering the issues regarding the sizing of Low Noise Amplifiers due to the complexity of the imposed requirements, when compared with low frequency circuits, results in a more complex optimization process, which can be applied to computing methods capable of providing a more viable and faster solution when compared to traditional sizing methods.

1.2 Proposed Solution

The proposed solution stands on an optimization platform capable of replacing the traditional optimization methods. The developed platform, uses a Genetic Algorithm to generate the LNA components size, which will be further analyzed using the Cadence Spectre tools to provide an accurate result for the rating of the LNA sizing. The communication between the Genetic Algorithm and the Cadence Software is ensured by the Sockets communication for Cadence [14].

The used topology to optimize is a well known wideband LNA with noise and distortion cancellation [1] which is a simple architecture and provides good performance. To boost the LNA performance some changes are applied to the architecture, replacing the passive loads for active loads [10], increasing the search space for the GA to operate.

The optimization platform, provides a menu with the implemented optimization methods, where each one shows the taken approaches to obtain a viable sizing for the circuit.

1.3 Thesis Organization

This thesis is organized in five chapters, being this one the introduction.

Chapter 2 contains the fundamental concepts,containing the information related to evolutionary algorithms and low noise amplifiers.

Chapter 3 contains the state-of-the-art alternative optimization methods applied to LNAs. Chapter 4 presents the proposed optimization platform in order to fulfill the circuits specifications.

Chapter 5 contains the LNAs architectures which were implemented for the optimization purpose and their respective implementation method.

Chapter 6 presents the obtained results from each method related to the developed optimization platform.

Chapter 7 contains a discussion of the results and the taken conclusions as well as the suggested future work.

Chapter 2

Fundamental Concepts

This chapter will expose, in a first topic,a brief introduction to evolutionary algorithms. The following topic presents the fundamental concepts of the low noise amplifiers, to allow a better comprehension of the developed work.

2.1 Evolutionary Algorithms

Evolutionary computing is an area that is in great development, due to the massive increase of computational systems and to the fact that, this systems, allow the development of algorithms that are capable of solving difficult problems, yet to be solved by "classical"techniques. Evolutionary algorithms are methods based on basic principles of genetic and evolutionary theory [6].

The algorithms work with a group of possible solutions, called population, which try to survive to natural selection processes. The evolution is characterized by the introduction of random variations through the process of reproduction in the population. The reproduction process allows the transmission of genetic heritage which will compose the new generation of individuals.

Having in consideration Darwin's evolution theory, it is possible to propose an evolutionary algorithm pattern which is presented in Fig.2.1 and in Algorithm.1. The algorithm starts from a population of individuals, who are candidates to the problem's solution. Each individual as a structure of data which defines him. Those individuals reproduce themselves, generating new individuals who may obtain some of their parents characteristics. Through these processes genetic mutations may occur, conferring genetic variation to the population. Natural selection is applied through a function, called fitness function, which evaluates the individual and gives him a score according to his fitness. The individuals with the higher fitness tend to survive, and those with the lower fitness

Algorithm 1 General scheme of an evolutionary algorithm in pseudocode adopted from [6].

INITIALIZE population with random candidate solutions
EVALUATE each candidate
while *TERMINATION CONDITION* **do**
  SELECT parents
  RECOMBINE pairs of parents
  MUTATE new candidates
  EVALUATE new candidates
  SELECT individuals for the next generation
end while

tend to be eliminated from the population. At the end of all these processes it is said that a generation has occur. The algorithm only stops if a previously fixed number of generations has been reached or when the reached solution is satisfying.

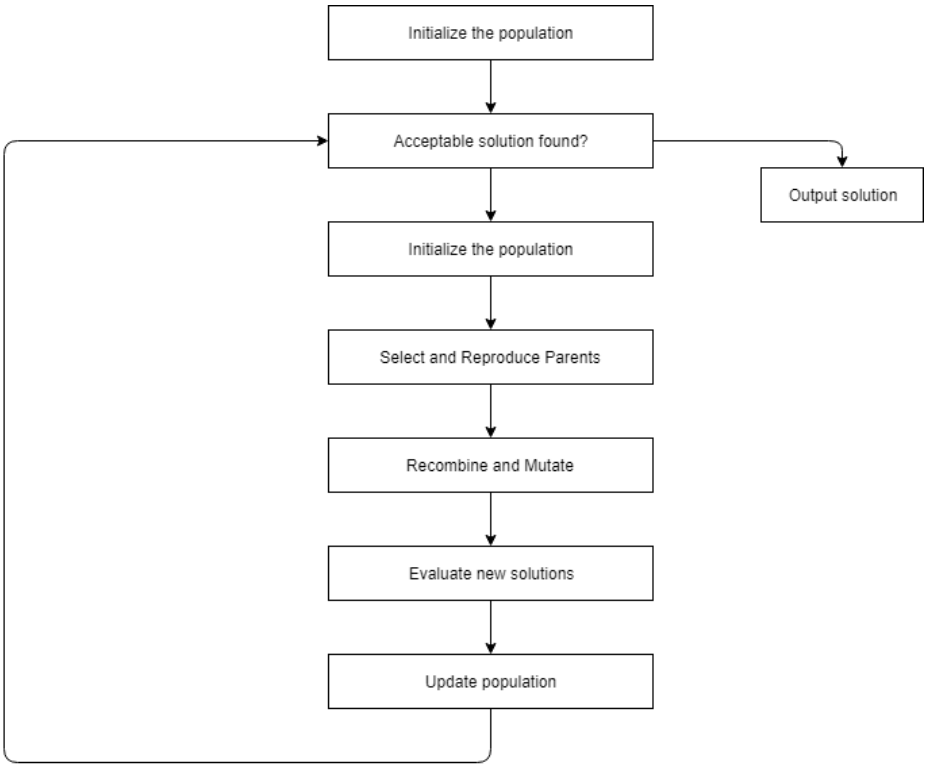

Figure 2.1: Outline of an Evolutionary Algorithm adapted from [17].

2.1.1 Components of Evolutionary Algorithms

To define a particular EA, there are a components that need to be specified. The most important components are [3]:

- representation
- evaluation function

- population
- parent selection mechanism
- recombination and mutation

It is necessary to specify each of the components in order to generate a complete and runnable algorithm.

2.1.1.1 Representation

To create an algorithm it is essential to create a link between the real world and the computational world. Therefore, the first step to have in consideration is to find a way to connect the problem context to the problem-solving space where evolution takes place.

From the point of view of the automated problem-solving it is necessary to decide how possible solutions should be specified and stored in a way that can be manipulated by a computer. The objects that form possible solutions within the original problem context are referred to as phenotypes, while their enconding are called genotypes. This step is called representation since it intends to specify a bridge trough the phenotypes onto a set of genotypes that can represent them. For example, if a binary code representation is chosen the phenotype could have the value 18 and its genotype would be seen as 10010.

The inadequate definition of the representation may lead to solutions that could not be the most suitable to the problem context, which is why this is one of the most critical steps in the implementation of the evolutionary algorithm.

2.1.1.2 Evaluation function

The evaluation function is responsible for setting a measure of quality to the genotype. Normally it is set by a quality function in the phenotype space and than converted to genotype space. Following the previously used example, if the goal is to find and integer x that maximises x^2, then the fitness of the genotype 10010 could be defined by decoding its corresponding phenotype,$10010 \rightarrow 18$, and then applying $18^2 = 324$.

The evaluation function is usually called fitness function in EA.

2.1.1.3 Population

The objective of the population is to hold the possible solutions. In other words the population is characterized as a group of genotypes.

In almost every EA applications the population size is constant during the evolutionary search and the selection operators work at the population level. For instance, the best individual of a population is selected to create the next generation. Another possibility is to replace the worst individual.

2.1.1.4 Parent selection

The role of parent selection is to distinguish individuals based on their quality, to allow the better individuals to become parents increasing the quality of the next generation.

In EA, parent selection is typically probabilistic. Therefore, individuals with a higher fitness are more likely to be selected to become parents than those who have less fitness.

It is important to have in consideration that the less adequate individuals are given a smaller probability, but they can also be selected, which guarantees that the evolutionary search doesn't get stuck in a local solution.

2.1.1.5 Recombination and mutation

Both processes, recombination and mutation, are responsible for generating new individuals from old ones.

Mutation This process is usually the name given to the genetic operators that allows the creation of new individuals from only one parent. This process is stochastic and depends on the previously chosen representation process and on the evolutionary strategy chosen by the designer.

The simplest way to satisfy this process is to allow the mutation operator to jump everywhere, allowing any allele to be mutated into any other, with a nonzero probability. However, many researchers consider this process irrelevant and EA implementations don't often possess this property.

Recombination This process is associated to a binary variation operator. This process merges the information from two parents genotype into one or two new child genotypes.

Like mutation, recombination is a stochastic operator: the choices of what parts of each parent are combined and how this is done depends on random variables. Therefore, this operator is usually dependent on the representation and on the chosen evolutionary architecture.

2.1.2 Genetic Algorithms

Genetic algorithms were developed by Holland in the decade of 1970 at the University of Michigan. Holland studied the natural evolution considering it a simple and powerful process that could be adapted to obtain computational solutions for optimization problems. GAs are commonly known for generating satisfying solutions without being hardly dependent on the initial conditions [5].

The biggest focus of genetic algorithms is the creation of new individuals through recombination operators, combining them with mutation operators which are normally incompatible. GAs do their search for the optimum solution, considering a population of candidates, using transition rules based on probabilistic processes. In Table.2.1 are represented the characteristics of a classic genetic algorithm.

Table 2.1: Genetic Algorithm characteristics adopted from [6].

Representation:	Bit strings
Recombination:	1 Point crossover
Mutation:	Bit flip
Parent Selection:	Fitness proportional- Roulette Wheel

Despite all the features in GAs there are some issues in its implementation. The use of binary strings may lead to non-optimum solutions if applied to a great dimension of numeric problems which require great numeric precision. Therefore, binary strings may be replaced by a different representation, like fluctuating point codification. Some issues may be found during the selection and the reproduction, which may lead to the loss of the best individual of the population, and so other strategies can be used in order to achieve better results.

2.1.2.1 Genetic Algorithm parameters

The use of genetic algorithms in optimization problems has been widely applied. However, there are some parameters that may alter the search for the optimum solution, which is why the need to be adapted to optimization problem in matter.

- Population Size- the population size affects directly the efficiency of the algorithm. A small population may lead to a small search space, which may corrupt the solution. If an elevated number of individuals in a population is applied, the algorithm may take a lot of time and resources, to achieve the optimum solution. Therefore, the values used for the size of the population are normally between 20 and 200 individuals, having in consideration the problem's complexity.

- Number of generations- the number of generations is related to the size of the population and with the available computational time for the algorithm's execution. There are no standard values for this parameter, but is highly recommended to have high numbers of generations.

- Crossover- GAs may suffer from premature convergence when the crossover tax is incorrectly chosen. If the crossover tax is too high the individuals may have great skills an may lead to a non-optimum solution. If the crossover tax is too small the algorithm may be too slow. Typically the chosen value is between 70% and 85%.

- Mutation- The used mutation tax is usually between 1% and 5%. This reduced tax serves to prevent that the algorithms stays locked in a local solution, and may find the optimum solution much faster. If a high mutation tax is selected the algorithms becomes almost random and turns the search for the solution really hard.

2.1.2.2 Additional operators

In GAs there are some additional operators that are added to achieve an higher performance of the algorithm. The most commonly used operators are:

- Elitism- This operator consists in replicating the best individual of a generation, keeping it unaltered through the new generation. With this the designers intend to don't lose the best possible solution through the reproduction process.

- Re-initialization- Initializing the population periodically is an operator that avoids the algorithm's stagnation. After a random number of generations the population converges to a solution that may not be the optimum. Therefore, with this operator, one of the individuals will the taken to another search space which may lead to a better solution.

- Niching- There might be a region of the search space that is undesired by the designer. Therefore, a function is added to the algorithms which intends to avoid that region and makes the population to stay away from it.

2.1.3 Particle Swarm Optimization

Particle Swarm Optimization is a stochastic optimization method developed by Dr.Eberhart and Dr.Kennedy [5]. This method is based on the social behavior of animals like bees, fishes and birds, where each decision is taken always from the leader.

The PSO method is related to evolutionary algorithms, since there is an initial population and each element of the population is evaluated at each iteration of the algorithm, in order to achieve the optimal solution of the problem. Despite the similarities, the PSO algorithm doesn't rely on evolutionary operators such as crossover, or mutation. The possible solution candidates approach to the optimal solution by moving themselves in the search space, following the candidates with the higher rate.

This method has set a good position in solving optimization problems in the engineering area, due to its easy implementation and good results.

2.1.3.1 Algorithm Description

In the PSO each member of the population is called a "particle". Each particle represents a solution in the search space for the problem's solution. All particles have associated to it a fitness score and a velocity which define the movement of the particle in the search space.

The algorithm starts with a random group of particles, creating a population, where each particle is then evaluated in the search for the optimal solution. At each iteration of the algorithm, each of the particles is updated in two values. The first one is given by the best solution found at the moment by the particle, which is defined as p_{best}. The second

value is given by the best value of fitness obtained among the particles, g_{best}. After both values are defined, both speed and direction are updated in the particle according to:

$$v_{i+1} = v + c_1 \times rand() \times (p_{best} - p_{fitness}) + c_2 \times rand() \times (g_{best} - p_{fitness}) \tag{2.1}$$

$$p_{fitness_{i+1}} = p_{fitness} + v_{i+1} \tag{2.2}$$

where v defines the particle's velocity, c_i are learning factors, and rand() a random number generator between zero and one.

Algorithm 2 PSO pseudo-code representation adapted from [5].

```
for each particle do
  Initialize particle randomly
end for
while Stop Criteria is not achieved do
  for each particle do
    Calculate fitness
    if particle fitness> p_best then
      p_best=fitness
    end if
  end for
  Choose the particle with the highest fitness value - g_best
  for each particle do
    Update particle velocity
    Update particle new position
  end for
end while
```

In Algorithm.2 is represented the algorithm's pseudo-code. The velocity in each particle is limited to a certain value which is defined by the user. If the sums of the accelerations results in a higher value than expected the particle's velocity is then settled with the defined value given by the user.

2.1.3.2 PSO Parameters

In the development of the algorithm there are a few parameters to have in consideration:

Number of particles Typically the used number of particles stays between twenty and forty. However the complexity of the problem may alter the value of used particles, for example, a simple problem may only need ten particles, but a more complex problem may need around two-hundred.

Particles' Dimension The dimension of the particles is settled by the complexity of the problem and its representation.

Maximum Velocity Represents the maximum alteration a particle may suffer during an iteration. If a high value is chosen, the algorithm tends to converge fast but may become highly unstable and never converge to the optimal solution. A small value leads to a low speed algorithm, which consumes a lot of time.

Learning factors The learning factors, c_1 and c_2, have typically values between zero and four, ensuring good results.

2.2 Low Noise Amplifiers

A low noise amplifier is a block responsible for amplifying the microwave signal without increasing its noise. This kind of amplifiers require a matching impedance, a low noise factor and a high gain specifications in order to fulfill the proposed requirements.

2.2.1 Impedance Matching

In high frequency circuits, the wavelengths are the same order of magnitude as the physical dimensions of the circuit, therefore, lumped circuit analysis is not appropriate since it assumes instantaneous signal propagation. This way, circuit paths behave like transmission lines.

A segment of a transmission line can be represented by an equivalent lumped circuit, as shown in Fig.2.2, where R, L, C and G are frequency-dependent [13]. The resistance R represents the finite conductivity of the conductors, the inductance L is related to the self-inductance of the wire and the mutual inductance between the conductors, the capacitance C defines the proximity of the the conductors and the conductance G is the electric loss between the conductors.

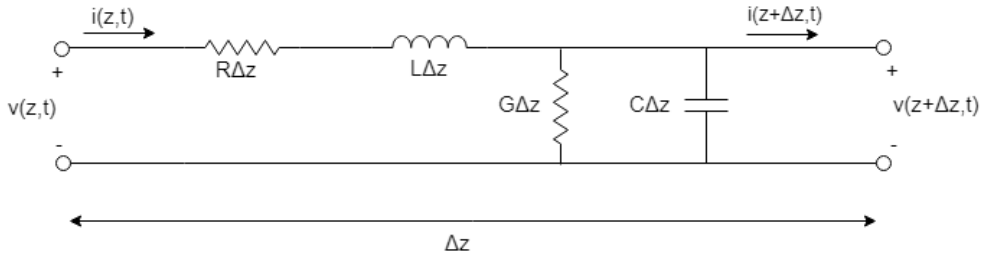

Figure 2.2: Lumped circuit equivalent of a transmission line adopted from [10].

Given that the system can be represented as an equivalent lumped circuit, the Kirchhoff's voltage and current laws can be applied, which lead to

$$v(z,t) - R\Delta z i(z,t) - L\Delta z \frac{\partial i(z,t)}{\partial t} - v(z+\Delta z,t) = 0 \tag{2.3}$$

$$i(z,t) - G\Delta z v(z+\Delta z,t) - C\Delta z \frac{(z+\Delta z,t)}{\partial t} - i(z+\Delta z,t) = 0 \tag{2.4}$$

Dividing both equations by Δz and taking the limit to $\Delta z \to 0$, results in

$$\frac{\partial v(z,t)}{\partial z} = -Ri(z,t) - L\frac{\partial i(z,t)}{\partial t} \tag{2.5}$$

$$\frac{\partial i(z,t)}{\partial z} = -Gv(z,t) - C\frac{\partial v(z,t)}{\partial t} \tag{2.6}$$

These equations can be simplified if sinusoidal steady-state conditions are considered, which lead to

$$\frac{dV(z)}{dz} = -(R + j\omega L)I(z) \tag{2.7}$$

$$\frac{dI(z)}{dz} = -(G + j\omega C)V(z) \tag{2.8}$$

Deriving both terms of Eq.2.7 and Eq.2.8 results in

$$\frac{d^2V(z)}{dz^2} - \gamma^2 V(z) = 0 \tag{2.9}$$

$$\frac{d^2I(z)}{dz^2} - \gamma^2 I(z) = 0 \tag{2.10}$$

where,

$$\gamma = \sqrt{(R + j\omega L)(G + j\omega C)} \tag{2.11}$$

defines the propagation constant, which is dependent on frequency. Solving both previous differential equations it is possible to obtain the voltage and current at any specific point of the transmission line which are presented in the following equations

$$V(z) = V_0^+ e^{-\gamma z} + V_0^- e^{\gamma z} \tag{2.12}$$

$$I(z) = I_0^+ e^{-\gamma z} + I_0^- e^{\gamma z} \tag{2.13}$$

where the wave propagation in the +z and -z is given by the terms $e^{-\gamma z}$ and $e^{\gamma z}$ respectively. Deriving Eq.2.7 and applying to the voltage of Eq.2.12 the current on line is given by

$$I(z) = \frac{\gamma}{R + j\omega L}(V_0^+ e^{-\gamma z} - V_0^- e^{\gamma z}) \tag{2.14}$$

Therefore

$$I_0^+ = V_0^+ \frac{\gamma}{R + j\omega L} \tag{2.15}$$

$$I_0^- = V_0^- \frac{\gamma}{R + j\omega L} \tag{2.16}$$

Transmission line characteristic impedance is given by:

$$Z_0 = \frac{R + j\omega L}{\gamma} \tag{2.17}$$

If the line is terminated by a load Z_L at $z = 0$ and the assuming that the source of the wave is located at $z < 0$, it is verified that

$$Z_L = \frac{V_0^+ + V_0^-}{V_0^+ - V_0^-} Z_0 \tag{2.18}$$

where V_0^+ and V_0^- are the amplitude voltages of the incident and reflect waves, respectively.

Solving the previous equation in order to V_0^-/V_0^+, shows that the voltage reflection coefficient Γ, which represents the amplitude of the reflected voltage wave normalized to the amplitude of the incident voltage wave, is given by:

$$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0} \tag{2.19}$$

To achieve the maximum power transfer to the load, the reflection should be non-existent,$\Gamma = 0$, which only occurs when $Z_L = Z_0$. If this happens, the load is matched to the line's characteristic impedance.

In RF, the characteristic impedance of an antenna is 50Ω, so that the receiver may have a matching input impedance.

2.2.2 Scattering Parameters

The usual system characterization used in low frequencies, through open and short-circuit measurements, is not appropriate since this approach usually involves the the magnitude and phase of traveling or standing waves [12]. Therefore, circuit measurements, at high-frequencies, are made using the average power instead of the traditional open and short-circuit measurements. The scattering parameters, or S-parameters, relate the voltages of the incident and reflected waves,at n-ports, through the scattering matrix,

$$\begin{bmatrix} V_1^- \\ \dots \\ V_n^- \end{bmatrix} = \begin{bmatrix} S_{11} & \dots & S_{1n} \\ \dots & & \dots \\ S_{n1} & \dots & S_{nn} \end{bmatrix} \begin{bmatrix} V_1^+ \\ \dots \\ V_n^+ \end{bmatrix} \tag{2.20}$$

where V_i is the voltage amplitude on port i and the signal, + and -, are related to the incident and reflected waves respectively.

Specific S-parameter can be determined through the equation:

$$S_{ij} = \left. \frac{V_i^-}{V_j^+} \right|_{v_k^+ \neq 0, k \neq j} \tag{2.21}$$

which implies that an S-parameter S_{ij} can be determined as the ratio between the reflected wave voltage, at the port i and the incident wave at the port j, when the other ports are terminated with a matched load so that reflections are avoided. These parameters can be measured by using a network analyzer.

If a two-port network, Fig.2.3, is considered, the S-parameters can be designated [13]:

- S_{11}- Input reflection coefficient
- S_{12}- Reverse voltage gain
- S_{21}- Forward voltage gain

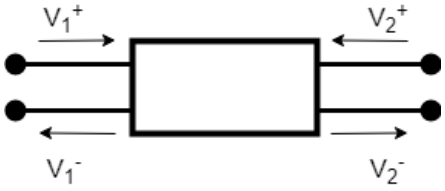

Figure 2.3: Incident and reflected waves in a two-port network adapted from [10].

- S_{22}- Output reflection coefficient

The S-parameters are very important in the LNA design due to the need of input matching, which is associated to the return loss that indicates the fraction of incident power that is reflected back to the source. The input return loss is given by

$$RL = -20log(|S_{11}|) \tag{2.22}$$

where the designers aim to have at least 10dB return loss, which implies a maximum of 10% of power that is reflected back to the source.

2.2.3 Gain

Signals at the receiver's input are usually very weak and need to be amplified in order to be processed. Therefore, the gain is a very important technical specification in an amplifier's design because it expresses the capability of the circuit to increase the amplitude of the signal, ideally without increasing any distortion and noise.

There are, commonly, three types of gain [13]:

- Voltage gain
- Current gain
- Power gain

where

$$A_v = \frac{v_{out}}{v_{in}} \tag{2.23}$$

defines the voltage gain,

$$A_i = \frac{i_{out}}{i_{in}} \tag{2.24}$$

defines the current gain, and

$$A_p = \frac{p_{out}}{p_{in}} \tag{2.25}$$

defines the power gain. Due to simplicity the gain is often expressed in dB where voltage and current gains are expressed as

$$A_{v,i}\big|_{dB} = 20log|A_{v,i}| \tag{2.26}$$

and power gain is expressed as

$$A_p\big|_{dB} = 10log|A_p| \tag{2.27}$$

2.2.4 Noise

In electronic circuits, noise is present as a variable that cannot be predicted at any time which are caused by external interference or by physical phenomena due to the nature of the materials. Having in mind that the presence of noise in circuits is inevitable, it is important to analyze its influence and develop methods to minimize its effect.

2.2.4.1 Noise Figure

The noise factor, F, or the noise figure, NF,when expressed in dB, are both the most common factor to measure the noise generated by a circuit.

Considering a circuit characterized by a 2-port network, the noise factor is defined as the ratio between the total noise power at the 2-port output and the 2-port output noise power due to the input noise sources only and is given by [8]:

$$F = \frac{Totaloutputnoisepower}{Outputnoiseduetothesource} \tag{2.28}$$

and the noise figure can be expressed as:

$$NF = 10logF \tag{2.29}$$

2.2.5 Linearity

Characterization of linearity can be done by the 1dB compression point and by the 3rd-order intermodulation product, and both parameters appear in the system's specifications.

A linear system generates an output proportional to the input signal, however, most devices have a non-linear characteristic, and if they are memory-less and time invariant, the input-output relationship may be described by a Taylor series

$$y = \sum_{i=0}^{n} a_i x^i \tag{2.30}$$

The type of linearity is directly related to the terms used, and its representation becomes more accurate with the increase of used terms.

2.2.5.1 Harmonics

Harmonics are generated by nonlinear devices. A $3^{rd} - order$ polynomial is usually a good approximation, since it allows simplifications to be done in the calculations.

Assuming a sinusoidal input signal where

$$v_i(t) = V_m cos(\omega_f t) \tag{2.31}$$

the output results in

$$y(t) = a_0 + a_1 V_m cos(\omega_f t) + a_2 V_m^2 cos^2(\omega_f t) + a_3 V_m^3 cos^3(\omega_f t) \tag{2.32}$$

or

$$y(t) = a_0 + \frac{a_2 V_m^2}{2} + \left(a_1 V_m + \frac{3a_3 V_m^3}{4}\right)cos(\omega_f t) + \frac{a_2 V_m^2}{2}cos(2\omega_f t) + \frac{a_3 V_m^3}{4}cos(3\omega_f t) \quad (2.33)$$

where

- $a_0 + \frac{a_2 V_m^2}{2} \Rightarrow$ DC component
- $\left(a_1 V_m + \frac{3a_3 V_m^3}{4}\right)cos(\omega_f t) \Rightarrow 1^{st}$ Harmonic
- $\frac{a_2 V_m^2}{2}cos(2\omega_f t) \Rightarrow 2^{nd}$ Harmonic
- $\frac{a_3 V_m^3}{4}cos(3\omega_f t) \Rightarrow 3^{rd}$ Harmonic

An n-order nonlinearity will generate n harmonics, each one with a frequency that is multiple of the fundamental frequency,ω_f.

2.2.5.2 Intermodulation product

Assuming that,instead of applying at the input a single sinusoidal signal, two signals with two different frequencies are applied, the input signal can be described as

$$v_i(t) = V_1 cos(\omega 1t) + V_2 cos(\omega 2t) \quad (2.34)$$

the output intermodulation product results in

$$y(t) = a_0 + a_1(V_1 cos(\omega_1 t)V_2 cos(\omega_2 t)) +$$

$$a_2\left[\frac{V_1^2}{2}(1 + cos(2\omega_1 t)) + \frac{V_2^2}{2}(1 + cos(2\omega_2 t)) + V_1 V_2(cos((\omega_1 + \omega_2)t) + cos((\omega_1 - \omega_2)t)\right] +$$

$$a_3\begin{bmatrix} (\frac{3}{4}V_1^3 + \frac{3}{2}V_1 V_2^2)cos(\omega_1 t) + (\frac{3}{4}V_2^3 + \frac{3}{2}V_2 V_1^2)cos(\omega_2 t) + \\ \frac{3}{4}V_1^2 V_2(cos((2\omega_1 + \omega_2)t) + cos((2\omega_1 - \omega_2)t)) + \\ \frac{3}{4}V_2^2 V_1(cos((2\omega_2 + \omega_1)t) + cos((2\omega_2 - \omega_1)t)) + \\ \frac{3}{4}(V_1^3 cos(3\omega_1 t) + V_2^3 cos(3\omega_2 t)) \end{bmatrix} \quad (2.35)$$

and the appearance of the different frequencies are presented in Fig.2.4 for the particular case of a device with a 3^{rd}order non-linearity.

2.2.5.3 1dB Compression Point

The 1 dB compression point is a linearity measure of a circuit which is defined as the output signal power that corresponds to 1dB difference from the nominal output of the ideal (linear) circuit and it is represented in Fig.2.5. Past this point, the signal consequently starts to degrade.

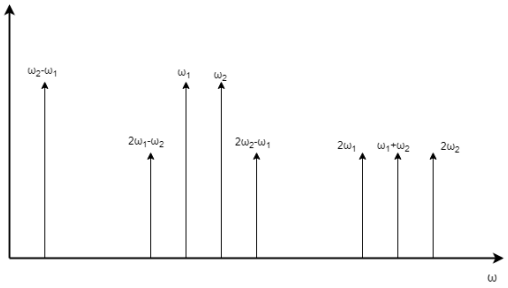

Figure 2.4: Frequency spectrum of a $3^{rd} order$ nonlinear device adapted from [10].

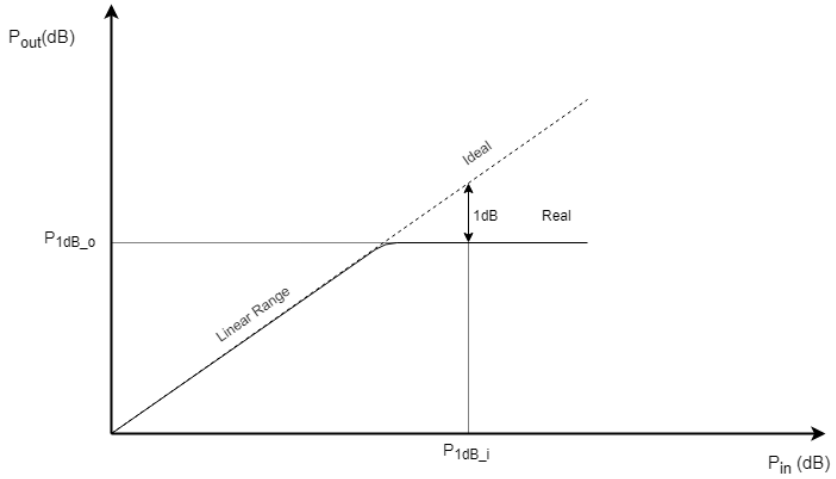

Figure 2.5: 1dB Compression Point adapted from [10].

2.2.5.4 Third-order Intercept Point

The third-order intercept point, IP3, is defined as the interception point between the linear curves of the power output of the fundamental frequency and the third-order intermodulation product, IM3. The specification of the IP3 is usually input-referred, IIP3, but it can also be output-referred, OIP3, as it is shown in Fig.2.6.

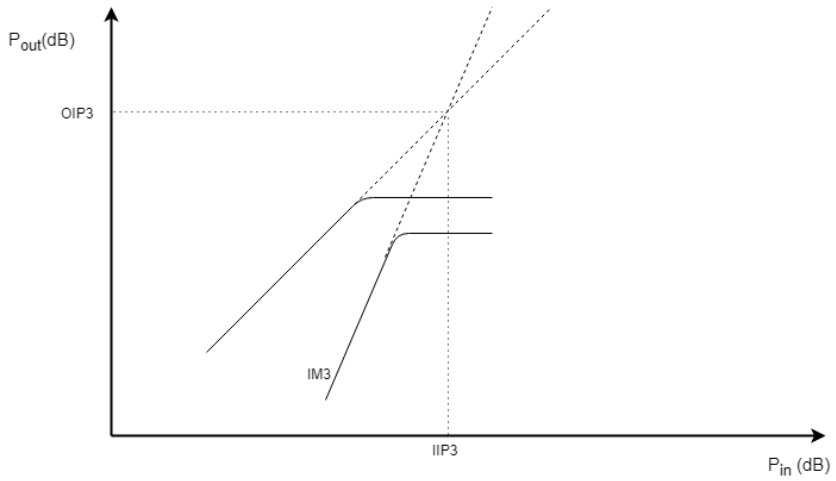

Figure 2.6: Definition of the IP3 adapted from [10].

Chapter 3

Related Work

This chapter is devoted to present a review of state-of-the-art optimization processes and different approaches to wideband LNAs.

Considering the development of the processing units that are present in our days, some new approaches related to the optimization of circuits are being presented. Choosing a new approach to replace a traditional optimization process is a huge issue, when considering the multiple requirements related to high-frequency architectures. Therefore, simple and easy implementation alternative methods are chosen to allow the developers to understand and be able to make adjustments to the developed optimization software.

3.1 LNA Parametric Simulation-based Optimization

In [4] an alternative optimization method is proposed where the design space was explored with parametric simulations in spectreRF. This method was set to optimize the circuit presented in Fig.3.1, where all the variables are represented through a vector,

$$\overrightarrow{x} = [W_{M1n}, W_{M1p}, W_{M2}, W_{M3}, R_F, R_B, V_C, I_{FB}] \tag{3.1}$$

where W_{M1n}, W_{M1p}, W_{M2} and W_{M3} represent the widths of the transistors. R_F and R_B represent the feedback and bias resistors. V_C represents the control voltage from the current-reuse stage and I_{FB} is the bias current from the feedback stage. The design space was limited by,

$$1\mu m \leqslant W_{Mi} \leqslant 300\mu m \tag{3.2}$$

$$0 < V_C \leqslant 1.5V \tag{3.3}$$

$$0 < I_{FB} \leqslant 10mA \tag{3.4}$$

the length of the transistors was set to 130µm which is the smallest dimension value from the used technology. The frequencies with interest were also put in a vector,

$$\overrightarrow{f} = [f_{std1}, f_{std1}, ..., f_{stdn}] \tag{3.5}$$

where f_{stdi} represents the work frequency for each optimization goal.

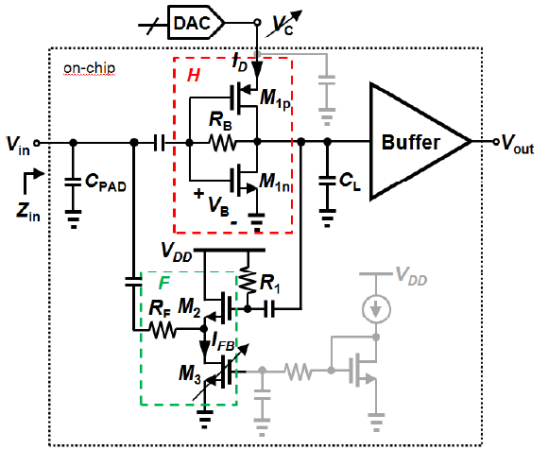

Figure 3.1: Proposed LNA Architecture from [4].

A brief analyze to the proposed circuit concludes that the gain is given by,

$$A_V \approx -G_{m1} Z_2 \tag{3.6}$$

where,

$$Z_2 = \frac{1}{g_{ds1}} \| \frac{1}{s(C_L + C_{gd2} + C_{db1})} \tag{3.7}$$

the bandwidth is represented by,

$$BW \approx \frac{1}{2\pi R_S (C_{gs1} + C_{gd1} + (1 + |A_V|))} \tag{3.8}$$

if ideal conditions and low frequency are considered, the input impedance is,

$$Z_{inLF} = \frac{1}{g_{m2} g_{m1} r_{o1}} \tag{3.9}$$

The circuits' noise can be considered, when $|A_V| >> 1$, by the equation,

$$F \approx 1 + \frac{\gamma_{1N} g_{m1N}}{G_{m1}^2 R_S} + \frac{\gamma_{1P} g_{m1P}}{G_{m1}^2 R_S} + \frac{R_F}{R_S A_V^2} + \frac{\gamma_2}{|A_V|}\left[1 - \frac{R_F}{R_S |A_V|}\right] +$$

$$+ \gamma_3 g_{m3} R_S \left[1 - \frac{R_F}{R_S |A_V|}\right]^2 + \frac{R_S}{R_B}\left[1 - \frac{R_F}{R_S |A_V|}\right]^2 \tag{3.10}$$

the linearity can be expressed using the equation,

$$IIP3 = \sqrt{\frac{4}{3}\frac{A_1(j\omega_1)}{A_3(-j\omega_1, j\omega_2, j\omega_2)}} \tag{3.11}$$

where $A_i(s)$ represent the transfer functions from the Taylor series.

According to the simulation, the power consumption of the circuit decreases with the size of the transistors, while the linearity increases with the size of the components. Therefore, a tradeoff between W_{M1n} and W_{M2} needs to be found in order to fulfill the circuits' specifications. To get the best set of results a figure of merit, FOM, is considered

$$FOM = \frac{G_{max}.IIP3_{max}.BW}{(F_{min}-1).P_{DC}} \tag{3.12}$$

where the G_{max} represents the maximum voltage gain expressed in V/V, $IIP3_{max}$ is the maximum input-referred third-order intercept point expressed in mW, BW is the is the -3 dB bandwidth in GHz, F_{min} expresses the minimum noise factor in linear units and P_{DC} goes for the the static power consumption of the LNA in mW.

Considering the complexity in finding a possible solution to the circuits' parameters, an algorithm is proposed in order to find the maximum FOM value. The algorithm is represented in Fig.3.2.

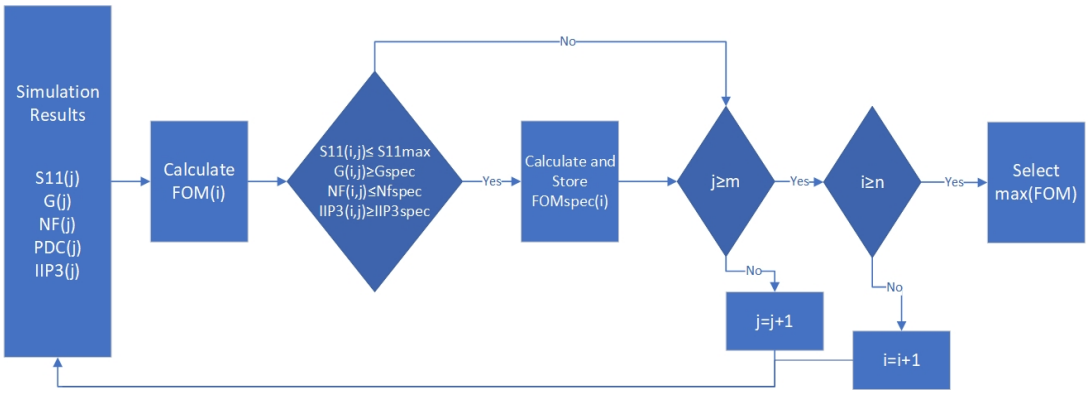

Figure 3.2: Proposed Algorithm from [4].

The presented algorithm calculates the figure of merit from the simulation results obtained from the parametric simulations in spectreRF. If the circuits specifications are fulfilled, the FOM is stored and the next set of simulation results are select. The process is repeated until all the simulations are analyzed. Finally the highest rated FOM is selected, which may guarantee the best set of results. This algorithm was implemented because the chosen FOM is more sensible to P_{DC} and IIP3 than to G, NF and BW. Therefore, some regions of values which contain high rated FOM don't fulfill the specifications. To guarantee the proper results from the algorithm, there was set a FOM_{spec} which contain the expected results which are,

$$\overrightarrow{x_{opt}} = [150\mu m, 150\mu m, 25\mu m, 10\mu m, 300\Omega, 14k\Omega, 1.3V, 650mV] \tag{3.13}$$

By the provided information it isn't possible to settle a comparison between the expected results and the obtained ones since there were only presented the results referent to the FOM parameters and not to the components dimension. Despite this, is possible to conclude that the obtained results are way above average when compared to other LNA architectures which are mentioned in the work.

3.2 LNA matching network sizing using a Genetic Algorithm

In [2] a genetic algorithm, GA, is proposed to optimize the input and output matching of a basic LNA architecture. This architecture is composed only by a RF transistor as represented in Fig.3.3.

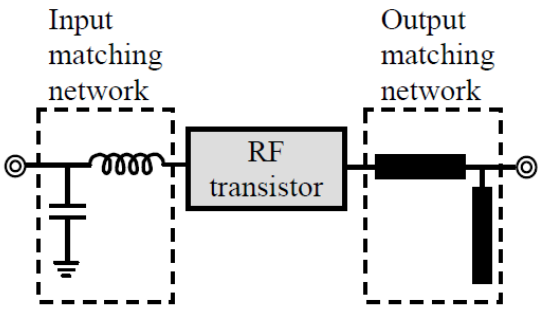

Figure 3.3: Proposed LNA architecture from [2].

The input matching network, as well as the output matching network are both composed by two components. The IMN is composed by an inductor, L, and a capacitor, C, while the OMN is composed by a transmission line and a stub. Therefore, the chromosome's is represented by a matrix which contains four rows and three columns. Each row represents a component from the matching network, while the first column represents the topology type, the second and third represent the electrical parameter for each parameter as shown in Fig.3.4. After the matching networks are represented, they are both integrated with the RF transistor to obtain the circuits' simulation results.

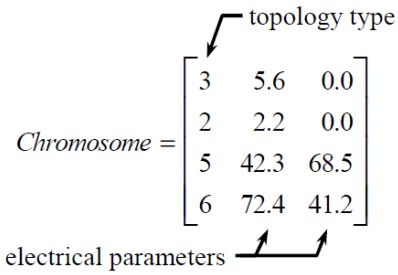

Figure 3.4: Matrix representation from [2].

To achieve the matching requirements the function to be optimized contains the matching conditions of the LNA, as well as the performance gain and noise figure,

$$F = \sum_{i=1}^{N} w_{i,S_{11}} f_{i,S_{11}} + w_{i,S_{22}} f_{i,S_{22}} + w_{i,S_{21}} f_{i,S_{21}} + w_{i,NF} f_{i,NF} \tag{3.14}$$

where $f_{i,\alpha}$ represents the difference between the obtained and the expected results. The $w_{i,\alpha}$ represents the weight considered to each simulation result. It is possible to identify that the two first parameters of the Eq.3.14 are responsible for the circuits' matching conditions, while the third and fourth parameters represent, respectively, the gain and noise figure results.

The algorithm can be described through:

- A population is initialized composed by the chromosomes with the matching network parameters;
- Each chromosomes is evaluated through the function described in Eq.3.14;
- The evolution of the population is insured by selection, crossover and mutation processes.

It is important to mention that the crossover and mutation processes where settled with a probability of 0.7 and 0.1, respectively. The convergency of the circuit was acquired in about seventy iterations which are obtained after three minutes of simulation.

The simulation requirements were settled to obtain:

- 2.4GHz LNA:
  - Input and output return loss < -10 dB
  - Power gain > 20 dB
  - Noise figure < 3 dB
  - 2.2 GHz < Frequency band < 2.5 GHz
- 5.2GHz LNA:
  - Input and output return loss < -10 dB
  - Power gain > 20 dB
  - Noise figure < 3 dB
  - 5.0 GHz < Frequency band < 5.3 GHz

Both tests were done by the Agilent ADS simulator to get the circuits' simulation results.

3.3 LNA matching network sizing using a Particle Swarm Optimizer

Another approach to set optimization goals is presented in [15] where a particle swarm algorithm is proposed to establish the circuits' design. The LNA architecture can be seen in Fig.3.5.

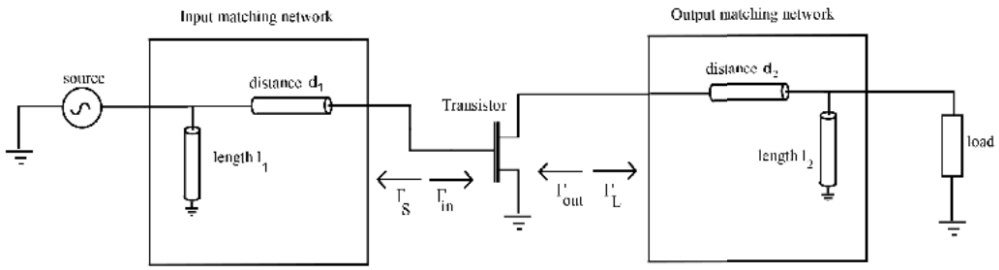

Figure 3.5: Proposed LNA architecture from [15].

To increase the knowledge of the algorithm's implementation is necessary to understand the circuits relative equations which were used to guarantee the convergence of the algorithm. The transducer power gain is given by,

$$G_T = \frac{|S_{21}|^2(1-|\Gamma_S|^2)(1-|\Gamma_L|^2)}{|(1-S_{11}\Gamma_S)(1-S_{22}\Gamma_L)-(S_{12}S_{21}\Gamma_S\Gamma_L)|^2} \tag{3.15}$$

where,

$$\Gamma_S = \frac{Z_{Source}-1}{Z_{Source}+1} \tag{3.16}$$

$$\Gamma_L = \frac{Z_{Load}-1}{Z_{Load}+1} \tag{3.17}$$

The noise figure of the circuit is given by,

$$F = F_{min} + \frac{R_N}{G_S}|Y_S - Y_{opt}|^2 \tag{3.18}$$

where,

$$Y_S = \frac{1}{Z_O}\frac{1-\Gamma_S}{1+\Gamma_S} \tag{3.19}$$

$$Y_{opt} = \frac{1}{Z_O}\frac{1-\Gamma_{opt}}{1+\Gamma_{opt}} \tag{3.20}$$

F_{min} represents the minimum noise figure from the transistor, R_N is the noise resistance from the transistor, G_S is the real part of the source admittance,Y_S is the source admittance and Y_{opt} goes for the admittance which results in a minimum noise.

The evaluation of the particles in the algorithm are done trough the gain, Eq.3.15, of the circuit, when considering a single frequency in the design. Since there are a lot

of requirements the circuit's noise figure was used as a fitness criteria, including also the Γ_{in} and Γ_{out}. The developed Algorithm.3, stands for a particle swarm optimization algorithm, PSO, where a group of particles are randomly initialized. For each particle, at every iteration, the fitness values, which represents how close the particles are to the desired solution, are calculated. There is a best local solution in each iteration, p_{best}, that is achieved. Among all particles, the best is labeled as g_{best}.

Algorithm 3 PSO scheme from [15].

INITIALIZE particles
CALCULATE fitness in each *particle*
while *MINIMUM ERROR CRITERIA* **do**
  CHECK fitness value in each *particle*, replace *pbest* if fitness is better than current *pbest* of the particle
  CHOOSE best *particle* among all *particles* with the best *pbest* and label *gbest*
  UPDATE each *particle* velocity and position according to the equations
end while

The equations to update the particles velocity and position are respectively:

$$v = w * v + c * rand() * (pbest - present) + c * rand() * (gbest - present) \tag{3.21}$$

$$present = present + v \tag{3.22}$$

where v stands for the particles' velocity and present for the position. Inertia weight, w, is set to 0.4 and the learning parameter, c, is set to 2.

For the algorithm's design a few considerations were taken:

- 15 particles were randomly generated with the input and output matching networks components parameters;
- Minimum required gain was set to 20 dB;
- Noise figure was set to be below 1 dB;
- 3000 iterations were considered for all the particles to converge.

The expected results were achieved where after five-hundred iterations the algorithms as more than twelve particles near the optimized solution, as presented in Fig.3.6.

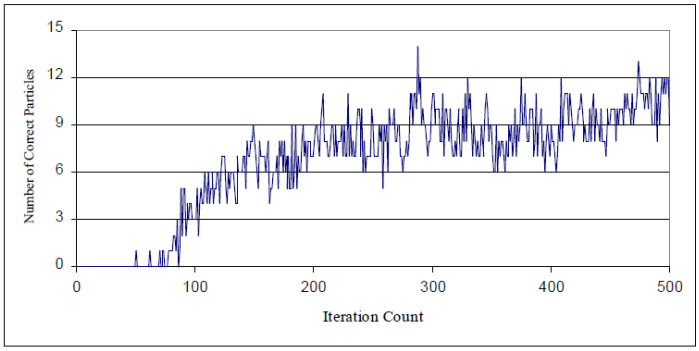

Figure 3.6: Obtained results from [15].

3.4 LNA optimization using a NSGA-II

A non-dominated sorting genetic algorithm, NSGA-II, in [7] is presented in order to optimize the circuit presented in Fig.3.7, which represents a LNA with a image rejection filter.

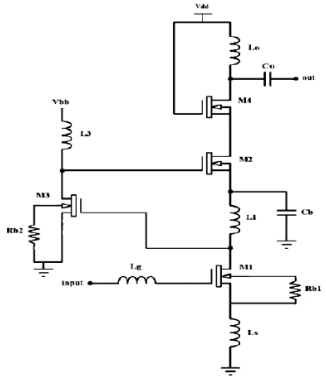

Figure 3.7: Proposed LNA architecture from [7].

This kind of optimizer are usually used in multiple objective problems, where, in this case, all of the objectives have the same weight when considered the relevance of each one. Initially a N individuals population is randomly initialized, and by using the usual genetic algorithm processes another population with the same size is created. Then the two populations are merged using a non-dominated sort algorithm which will generate a 2N population. The non-dominated sort generates a set of nondominated fronts. The solutions in the first non-dominated front are better than those in the second non-dominated front and so on. After the non-dominated sort process is done, new fronts are added to the N population starting with best selected front. After this process is done another sort based on crowding distances is applied guaranteeing the diversity of the individuals.The crowding distance method is very important to achieve solutions that set a tradeoff between the multiple objective functions to be optimized.The used expressions

are,

$$cdk(x_{[i,k]}) = \frac{z_k(x_{[i+1,k]}) - z_k(x_{[i-1,k]})}{z_k^{max} - z_k^{min}} \tag{3.23}$$

$$Cd(x) = \sum_k Cd_k(x) \tag{3.24}$$

where z_k is the function goal, while z_k^{max} and z_k^{min} are the maximum and minimum of this function, respectively. A graphic representation of this process is represented in Fig.3.8.

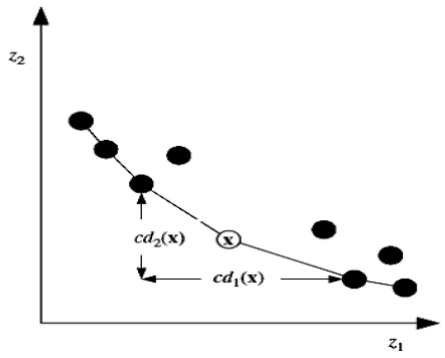

Figure 3.8: Crowding method from [7].

The implementation of the algorithm was done in MATLAB and the simulation results were obtain trough the HSPICE RF, where each object is evaluated, as presented in Fig.3.9.

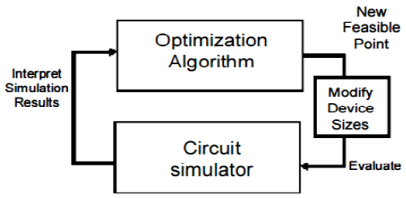

Figure 3.9: Proposed Algorithm's architecture from [7].

It is important to refer that due to the complexity of the circuit the circuit's optimization was done only to computer processes. The applied algorithm took about six-hundred iterations to achieve the expected results which are presented in Table.3.1, and were then compared with the ones from [9].

Table 3.1: Obtained results from [7].

Ref	Test 1	Test 2	Test 3
F0(GHz)	5.7	5.7	5.7
NF(dB)	1.68	1.63	1.68
S21(dB)	39.89	36.74	39.36
S11(dB)	-17.77	-15.88	-26.64
S22(dB)	-22.79	-28.14	-27.36
Vdd(V)	0.61	9.61	0.61
FOM1	23.74	22.53	23.42
fOM2	626.39	440.91	589.31

3.5 LNA sizing using an Evolutionary Algorithm with cost functions

In [16] another approach of an evolutionary algorithm, based on a GA is proposed. The EA is represented in Algorithm.4.

Algorithm 4 EA scheme from [16].

START a genetic optimization strategy
FIT the cost function after each population.Include only points within a defined radius from the active optimum
while *stop criteria isn't fulfilled* **do**
  COMPUTE the next evolutionary population using the circuit but calculate the value predicted by the model as well
  UPDATE the fit for each point within a radius from the active optimum.
  START a new evolutionary iteration using the fitted function.
  CONTINUE an adaptive number of evolutionary populations depending on the quality of the fit
end while

To achieve the optimum solution, this algorithm presents cost functions to attribute the respective fitness to each individual that is evaluated. The P_i represents each optimization parameter, and the cost functions are represented in Fig.3.10.

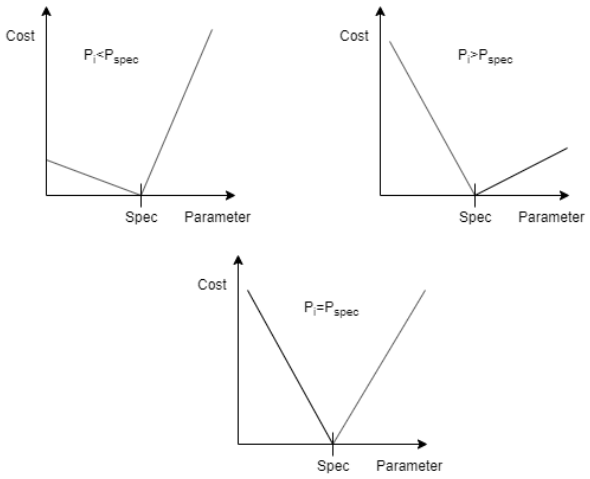

Figure 3.10: Cost functions from [16].

The circuit, presented in Fig.3.11, was optimized to the approached circuit's electrical model and only the parameters that were required to be extracted from SPICE are simulated. This way, the algorithm runs almost as two independent parts, which increases the total optimization speed.

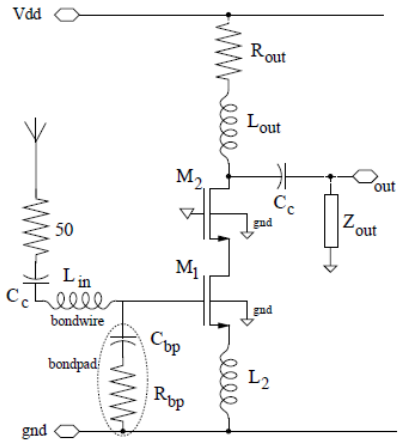

Figure 3.11: LNA architecture from [16].

A brief analyze to circuit ensures that the input third order intercept-point is given by,

$$IIP3 = 11.25 + 10log_{10}\left(\frac{V_{gst,M1}(1+r^2)(2+r)}{\theta}\right) \tag{3.25}$$

the input reflection coefficient is represented by,

$$S_{11} = \frac{Z_{in} - R_S}{Z_{in} + R_S} \tag{3.26}$$

the noise figure is expressed by,

$$NF = 10log_{10}\left(1 + \frac{R_l}{R_S} + R_{ch}R_S + \gamma g_{d0}R_S\left(\frac{\omega_0}{\omega_T}\right)^2\right) \tag{3.27}$$

finally the gain is given by,

$$Gain = G_M R_{out} \tag{3.28}$$

where G_M equals to,

$$G_M = gm_{M1} Q R_{out} \tag{3.29}$$

and Q represents the matching network quality factor. The algorithm was designed with the following specifications:

- S_{11} < 10 dB
- IIP3 > -10 dBm
- Gain > 20 dB
- NF < 2.5 dB
- Ibias = 5 mA
- Q < 2.5
- frequency = 1.8 GHz

The results were settle at a stable value in about six-hundred iterations.

3.6 LNA optimization using a Genetic Algorithm

A genetic algorithm based optimizer is proposed in [11]. This algorithm is set to achieve a possible solution in the optimization of the circuit presented in Fig.3.12.

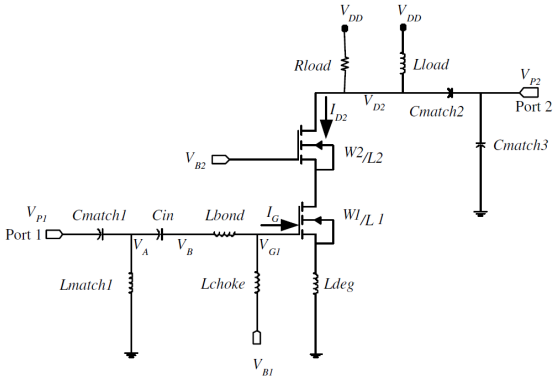

Figure 3.12: LNA architecture from [11].

To achieve the best solution the optimizer considers ten parameters of the LNA architecture which are presented in Table.3.2.

Table 3.2: Parameters to be optimized from [11].

Element	Range
C_{match1}	300-800 fF
C_{match2}	1-10 pF
C_{match3}	1-10 pF
L_{bond}	1-10 nH
L_{deg}	0.1-5 nH
L_{match1}	1-10 nH
R_{load}	1.5-5.5 Ω
L_{load}	1.5-5.5 H
V_{B1}	0.5-1.5 V
V_{B2}	0.5-5 V

Algorithm 5 GA scheme from [11].

INITIALIZE parameters extraction environment
INITIALIZE the GA
while *fitness score of the best chromosome > tolerated score* **do**
  SEARCH for better solution calculate the value predicted by the model as well
  if the evolution seems to be saturated **then**
    SEARCH best local solution
  end if
end while

The proposed genetic algorithm is represented in Algorithm.5. The problem was defined, in this work, as,

$$F(S_P, V_{in}, \overrightarrow{p}) = O_{result} \tag{3.30}$$

where S_p represents a netlist required by the circuit simulator. V_{in} is the input bias while $\overrightarrow{p}$ represents the parameters that need to be extracted. O_{result} represents the output of the simulation (S_{11}, S_{12}, S_{21}, S_{22},NF,...).

The gene encoding was done by transforming real numbers into discrete steps which represent each gene in a chromosome. This process was implement using the equation,

$$P_{value} = P_{min} + \frac{P_{max} - P_{min}}{Resolution} \tag{3.31}$$

The fitness function associated to the algorithm is:

$$F = \sum_{alltarget} (W(sim - spe)) \tag{3.32}$$

where the W represents a weight function which increases when there is a problem associated with the solution. The sim parameter represents the simulated result solution, while the spe is for the simulation's target to achieve.

The selection process is done after the fitness are attributed to each chromosome. This process is responsible for selection the best individuals, where in this work is done by a tournament method where two competitors are randomly chosen and the one with the

best fitness score is selected. This process repeats itself for a specified number of times. In Algorithm.6 is possible to observe the selection method.

Algorithm 6 Selection process scheme from [11].

SET number of contests according to the selection rate
while *contest isn't done* **do**
  PICK 2 competitors, randomly
  SELECT the competitor with the better fitness score
end while

A mutation process was also included in the algorithm which is responsible for not letting the algorithm getting stuck into non-optimal solutions. After the mutation process it was also implemented another method capable of verifying if the algorithm is obtaining a region of values that have a monotonic variation property, and if so the step size may be increased in order to speed up the optimization. This is also very useful when the optimization process founds a sensitive region, where the steps should be shortened to avoid skipping the possible optimum solution. This process was adapted from the Levenberg-Marquardt method.

In order for the algorithm to find a possible solution the following specifications were proposed:

- S_{11} < -10 dB
- S_{22} < -10 dB
- S_{12} < -25 dB
- S_{21} as large as possible
- NF < 2
- IIP3 < -10

The obtained results from the optimization are presented in Table.3.3 and Table.3.4.

Considering the experiment the application of the algorithm shows good convergency behavior when the population size is set to 50 individuals and the mutation rate is set to 0.5 which ensures the diversity of the evolution of the population. After 60 generations the population starts to converge to the optimal solution.

Table 3.3: Optimized parameters from [11].

Element	Result
C_{match1}	512.132 fF
C_{match2}	4.6104 pF
C_{match3}	4.5511 pF
L_{bond}	1.0782 nH
L_{deg}	1.145 nH
L_{match1}	6.202 nH
R_{load}	3.5 Ω
L_{load}	3.5 H
V_{B1}	0.75 V
V_{B2}	2.7 V

Table 3.4: Achieved results from [11].

Specification	Result
S_{11}	-14.1 dB
S_{22}	-22.6 dB
S_{12}	-39.3 dB
S_{21}	12.7 dB
NF	0.979
$IIP3$	-1.3

3.7 LNA sizing using a Particle Swarm Optimizer

In [] another PSO is proposed to fulfill the LNA circuit requirements presented in Fig.3.14. A flowchart of a PSO can be seen in Fig.3.13 where the equations to update the particles position and see are represented in Eq.3.21 and Eq.3.22.

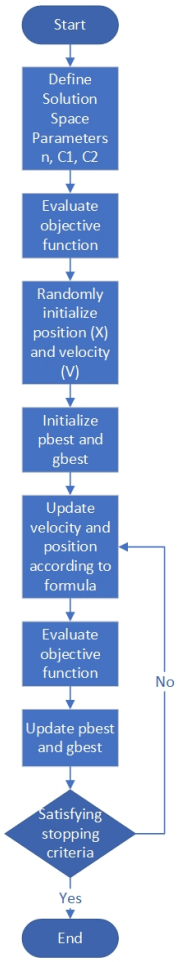

Figure 3.13: PSO flowchart from [].

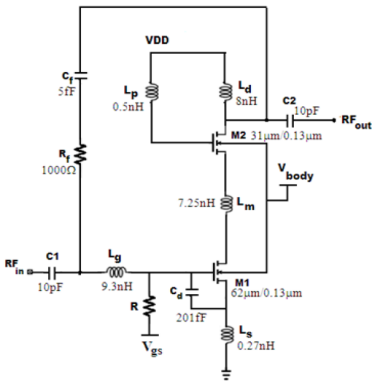

Figure 3.14: LNA architecture from [].

The algorithm is set to maximize the equation,

$$FOM = \frac{G}{(NF-1)P_{DC}} \tag{3.33}$$

where G represents the gain, NF the noise factor and P_{DC} the power consumption.

The gain equation can be expressed by,

$$A_V = Q_{in} g_{m1} Z_L \tag{3.34}$$

where Q_{in} is the input matching network quality factor, g_{m1} represents the transconductance of the main transistor, and Z_L is the load impedance.

The noise figure of the circuit is expressed as,

$$F = \frac{R}{R_s}\left(1 + R\frac{\gamma \omega_0^2 C_t^2}{\alpha^2 g_{d0}} X\right) \tag{3.35}$$

where,

$$X = \frac{\delta \alpha^2}{5\gamma}[1 + Q_{in}^2]\frac{C_{gs}^2}{C_t^2} + 1 - 2|c|\frac{C_{gs}}{Ct}\sqrt{\frac{\delta \alpha^2}{5\gamma}} \tag{3.36}$$

The power consumption equation is expressed as,

$$P_{DC} = V_{DD} \times I_{DC} \tag{3.37}$$

where the circuit's current is given by,

$$I_{DC} = W C_{ox} V_{sat} \frac{V_{od}^2}{V_{od}^2 + L E_{sat}} \tag{3.38}$$

The PSO was implemented using Matlab and the Advanced Design System (ADS) was used to simulate the circuit. To set the link between the algorithm and the optimizer, constraints were added to the design parameters:

- $10\mu m \leqslant W_i \leqslant 100\mu m$;
- L_i close to minimum feature size of the technology;
- $10mV \leqslant V_{odi} \leqslant 400mV$.

The initial population of the algorithm was developed using a matrix of seventy rows and three columns, 70x3, where seventy represents the number of particles in the population, while the columns' number is responsible for the vector with the simulation's results, $[A_V, NF, P_{DC}]$.

The executed test using the algorithm was then compared with a simulation made through traditional methods of optimization. The results are expressed in Table.3.5, where it is possible to conclude the PSO was efficient in finding an optimization solution for the LNA circuit.

Table 3.5: PSO optimization results adopted from [].

Specifications	Traditional Method Optimization	PSO optimization
P_{DC}	957 mW	869 mW
A_V	10.7 dB	12.6 dB
NF	3.27 dB	3.20 dB
FOM	4.92	6.6

3.8 Brief Comparison

Having in consideration the previously presented works there are a few points to have in mind when a brief comparison is settled. In the Table.3.6 are presented the pros and cons between the algorithms.

Table 3.6: Comparison between the algorithms.

Authors	Algorithm	Spent Time	Complexity
De Souza et al.[4]	Parametric Simulations	Average	Medium
Chen et al.[2]	GA	Low	Medium
Ülker [15]	PSO	High	Medium
Fallahpour et al.[7]	NSGA-II	High	High
Vancorenland et al.[16]	GA	Medium	High
Li[11]	GA	Low	Medium
Manjula and Selvathi []	PSO	Medium	Low

Having in mind the presented works, it is possible to notice that the main target of the algorithms is to get possible solutions according to a FOM of the circuit. The applied FOMs, despite their differences, all contemplate the gain and noise figure of the LNA architecture, since it is a simple solution to combine the imposed specifications. Another point to have in consideration, is the fact that the extracted optimization values are based in a certificated program. This method ensures the obtained results are much more accurate than the ones based on simple the circuit equations because they consider a lot more variables.

Since the circuits' specifications don't stand on linear equation models, computer based algorithms are being considered as a reliable solution to overcome traditional optimization methods. A big issue of the algorithms relates to the need of a structure capable of not getting stuck in local solutions, which increase the complexity of the problem.

Despite the difference between the works, there are some considerations which all have in common. Therefore, setting a link between them, is possible to conclude:

- The application of algorithms to optimize circuits is a possible solution.
- Evolutionary computing is a proper alternative to traditional methods.

- The diversity of algorithms applied to different architectures guarantee it is a proper alternative optimization method.
- The optimization time depends on how many parameters are set to be optimized.

Chapter 4

Optimization Platform

Circuits' sizing are becoming a region of interest for the application of self-learning mechanisms since it requires a lot of time and cost. Many of the applied approaches are based on circuits' equivalent equations which provide good and fast results.

The proposed optimization platform consists in using the provided simulations from Cadence for a more accurate result which feed an algorithm capable of finding a good sizing for the components in the circuits. To establish the link between the algorithm and the Cadence software, the optimization platform uses the SOCAD documentation which provides the necessary resources to ensure the communication with the server.

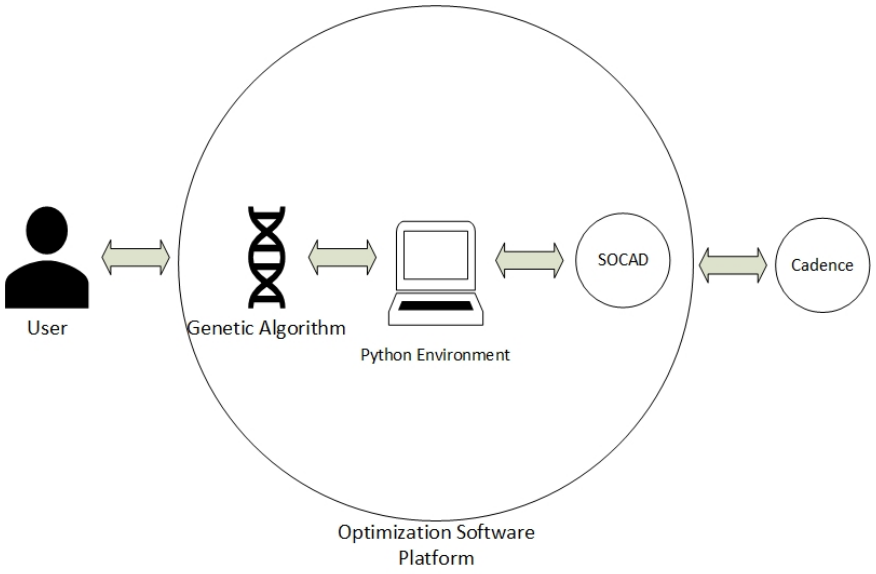

Figure 4.1: High level graphic representation of the developed system.

The optimization platform, represented in Fig.4.1, is implemented in Python which

provides multiple tools to ensure the development of the project. The platform runs in a created environment containing the DEAP documentation package, as well as the SOCAD documentation. After both packages are installed a connection with the Cadence server needs to be initialized providing the communication with the algorithm and the simulations results.

The optimizer can be describe in two main topics which are managed by the main platform allowing the connection between all two of them:

- Genetic Algorithm
- SOCAD Configurations

To establish a more friendly link with the user, the platform presents a list of options, presented in Fig.4.2, with different types of actions:

- 1-Load Simulator for LNA with Resistors
- 2-Update variables and run simulation with 2.4Ghz Optimization
- 3-Update variables and run simulation with 4 points Optimization
- 4-Wideband Optimization
- 5-Wideband Optimization with n=4 factor
- 6- Wideband Balun Optimization
- 7-Load Simulator for LNA with Active Loads
- 8-Wideband Balun Optimization for LNA with Active Loads
- 9-Wideband Balun Optimization for LNA with Active Loads with self-polarization
- 0-Exit

```
############### Optimization Software Platform ###############
1 - ************Load simulator for LNA with Resistor Loads***********
2 - Update variables and run simulation with 2.4G Optimization
3 - Update variables and run simulation with 4 points Optimization
4 - Wideband Optimization
5 - Wideband Optimization with n factor
6 - Wideband  Balun Optimization
7 - ***********Load simulator for LNA with Active Loads**********
8 - Wideband Balun Optimization for LNA with Active Loads
9 - Wideband Balun Optimization for LNA with Active Loads with polarization optimization
0 - Exit.

-> Please Select:
```

Figure 4.2: Platform Optimization options.

Initially the user needs to perform the loading of the circuit characteristics and simulations using the Load Simulator options, option 1 and 6, where a message of success will be sent if the simulator is properly loaded as presented in Fig.4.3.

```
Simulator loaded with success!!!
Key: r1 - Val:267.0
Key: r2 - Val:193.0
Key: r3 - Val:200.0
Key: L - Val:1.2e-07
Key: VB2 - Val:0.45
Key: VB1 - Val:0.8
Key: VDD - Val:1.2
Key: W1 - Val:4.1e-05
Key: IDC - Val:0.002
Key: W2 - Val:8.7e-05
```

Figure 4.3: Load Simulator successfully.

After the loading requirements are fulfilled the platform is ready to execute the optimization of the circuit's components. From this point the used optimization method will be dependent on the user's choice among the presented options in the platform.

After the optimization is selected the genetic algorithm will start its processes and the SOCAD software will send the provided information from the GA to the Cadence software, which will provide the necessary simulations, and send their results back to the platform and GA. This process will occur in a specific number of generations which are defined by the user. After the number of generations are concluded the GA will then provide the best individual among the final population.

4.1 System Modulation

In the system modulation there will be presented the use-case and sequence UML (*Unified Modeling Language*) diagrams, for a better understanding of the system's interaction.

4.1.1 Use Case Diagram

Analyzing the presented use-case diagram in Fig.4.4, it is possible to identify the actors in the system, as well as the interaction between the scenarios of the system. In this diagram are identified four actors: user, genetic algorithm, SOCAD and the Cadence. In the use case "Select Menu Option"the user is responsible to choose the desired option, and the SOCAD sets the link between the provided information and the Cadence server and program. This use case also includes the possibility to "Load Simulator"and "Start Optimization"which are the provided options in the menu.

Another use case is "Run optimization"where the Cadence is responsible to run the necessary simulations to feed the algorithm. In order to feed the algorithm another use case called "Get analysis results"is responsible to save the results from the simulations which will be used to calculate the fitness of each individual. The genetic algorithm is then responsible to "select the best individuals", another use case, which includes the applied

processes in the GA. It is also possible to observe that the "mutation"and "crossover"only occur when the individuals are selected for each process.

Finally the GA is responsible to get the best individual among the population after a reasonable number of generations. This use-case is represented by "Get best Individual".

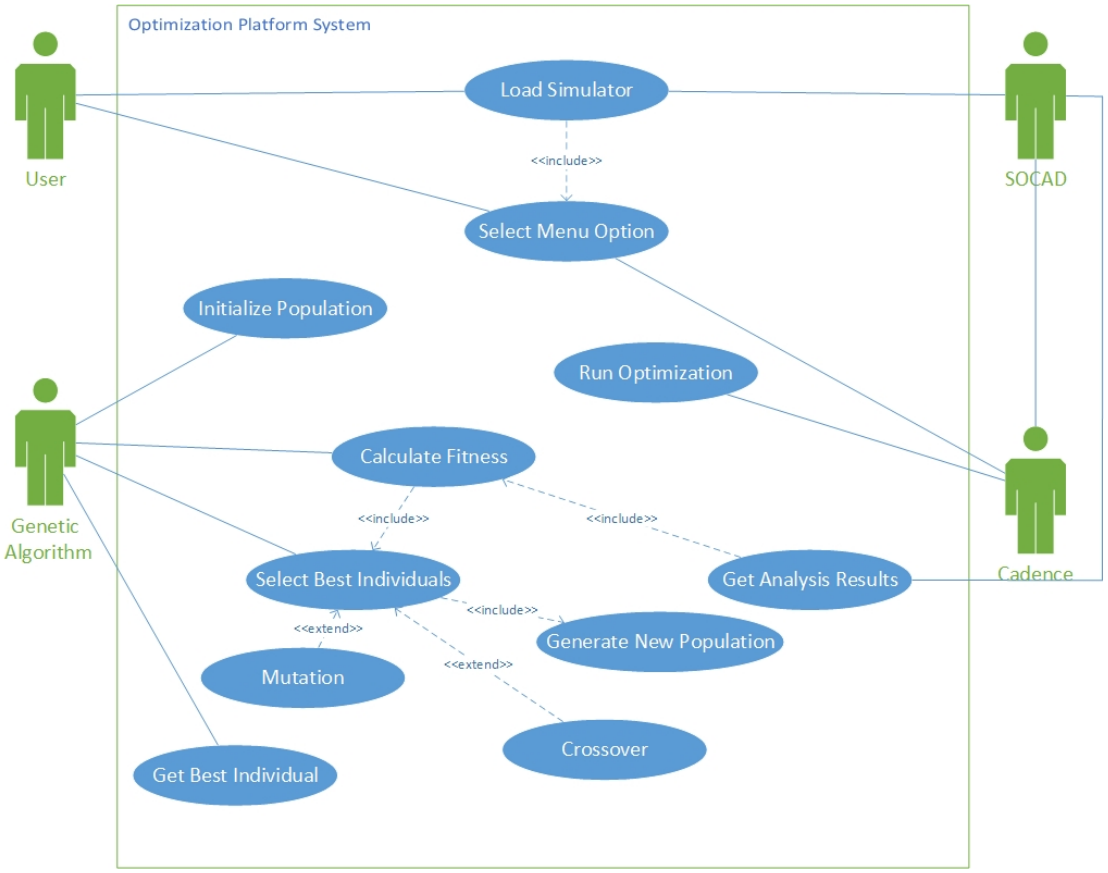

Figure 4.4: UML use case diagram.

4.1.2 Sequence Diagram

The UML sequence diagrams allow a temporal and behavioral perception of how the system interacts with the different actors and objects in the operation scenarios. To represent each element of the diagram it is necessary to identify the actors and objects. The actors are element which communicate autonomously in the system, while the objects don't have that ability, but have some interference in the processes.

In Fig.4.5 is represented the sequence diagram developed for this project.

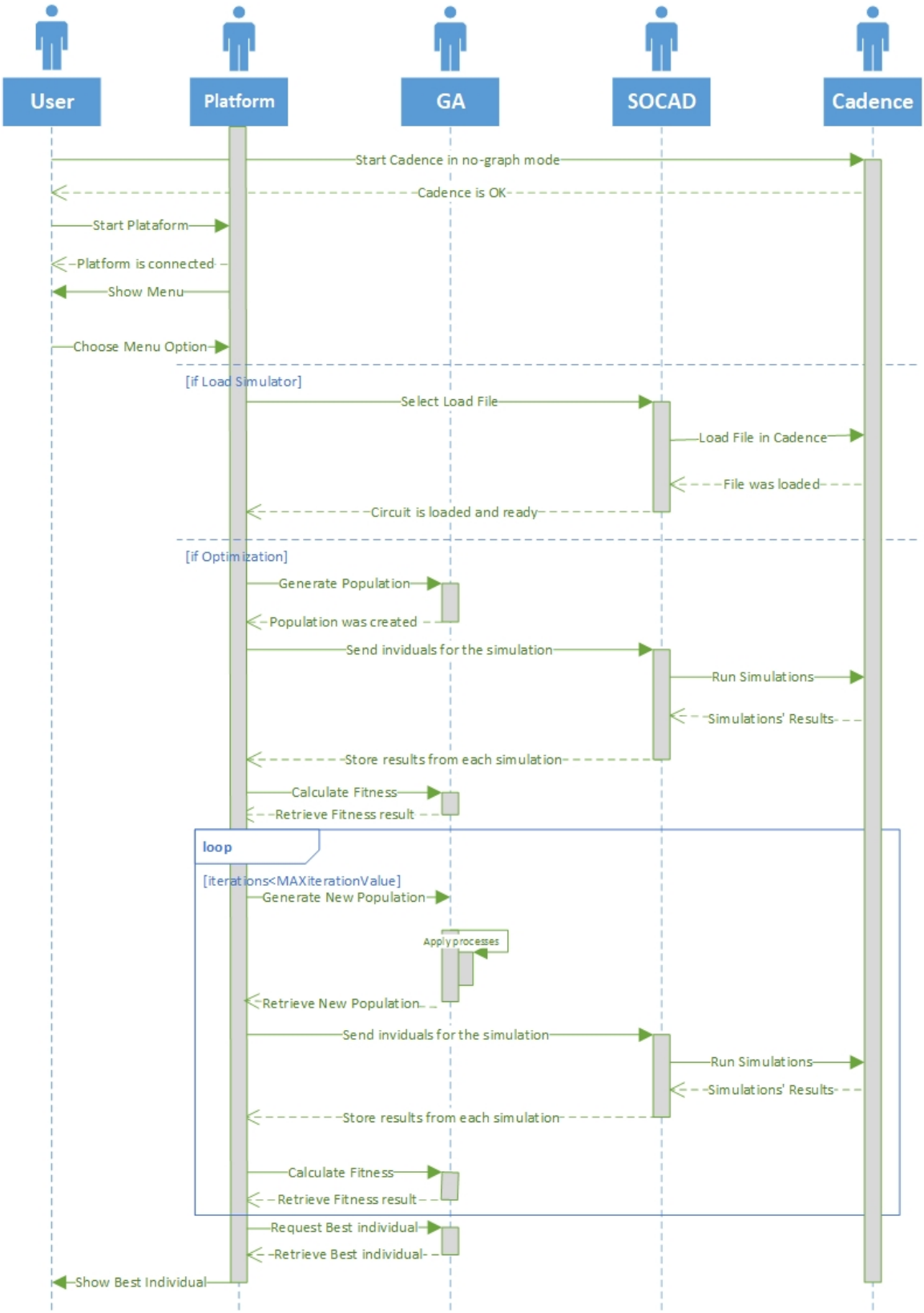

Figure 4.5: UML sequence diagram.

4.2 Genetic Algorithm

The implemented algorithm is based on the DEAP documentation where examples are provided, as well as libraries containing implemented processes related to evolutionary algorithms[6].

To be able to find an optimal sizing for the implemented circuits the chosen algorithm approach was based on a genetic algorithm with certain modifications to suit as well as possible the presented work. This modifications stand on the interactions between the main operators in the platform and, having that in mind, a classic genetic algorithm wouldn't be able to achieve good results in a reasonable time.

Establishing a link between the developed and modified genetic algorithm with the classic one, it is possible to identify five similar processes:

- Representation
- Selection
- Crossover
- Mutation
- Fitness calculation

4.2.1 Representation

To start the algorithm a population needs to be initialized. This process is responsible for generating the circuits' components sizing, to be further optimized by the platform, and is based on the DEAP libraries provided examples[]. Each component size is placed in a position of a list containing all the components to be optimized, creating an individual. The individuals set will form the population as shown in Fig.4.6.

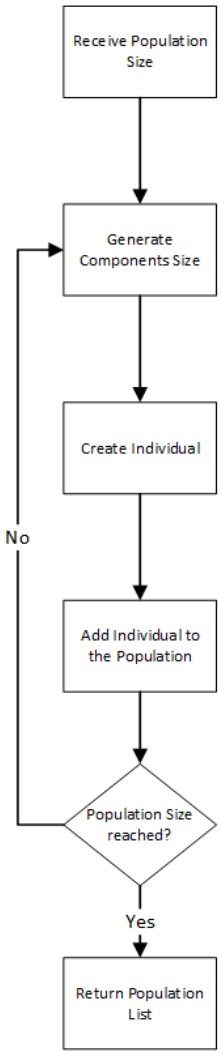

Figure 4.6: Representation process flowchart.

4.2.1.1 Individuals' Representation

As it was previously referred, the circuits' components are represented by a list containing the size of the components according to the selected circuit for the optimization purpose. In Fig.4.7 is possible to observe the chosen representation for each individual, having in consideration the characteristics of the circuits. The first vector corresponds to the circuit presented in Fig.5.3, where each position of the vector corresponds to a generated dimension for each correspondent component of the circuit, where V_{DD}, V_{B1}, VB_2 are the circuit's voltage sources, L represents the length of the transistors, I_{DC} represents the bias current, W_1 and W_2 correspond to the widths of the transistors M_1 and M_2, and R_1, R_2 and R_3 to the resistors. In the second vector there are a few changes, since the vector corresponds to the components from the circuit shown in Fig.5.5. The resistors are replaced by W_3 and W_4 representing the width of the transistors M_3 and M_4 respectively, and V_{Tune} is a voltage source which is also added to the list.

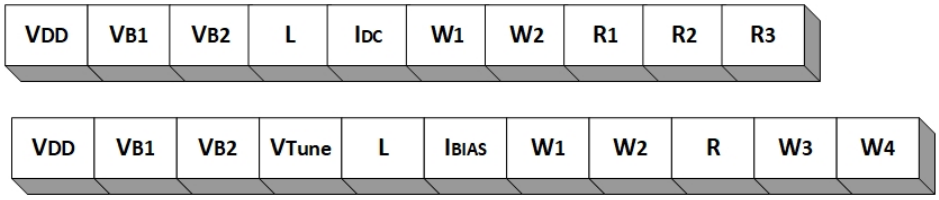

Figure 4.7: Individuals' representation.

4.2.1.2 Components restrictions

Components are restricted according to the selected optimization method in the provided menu of the platform. After the optimization type is defined by the user, each component is then selected and a random number is generated giving the chosen dimension for the component. After the random number is generated a round function is then applied to set a valid size for the component's dimension. In Algorithm.7 is possible to observe the applied process.

Algorithm 7 Restricted components creator.

Receive chosen optimization option;
Receive component to generate size;
Receive minimum and maximum component size values;
Generate random number for the component size according to the received values;
Round the previously generated number to a valid one;

4.2.1.3 Representation operator Modulation

A further analysis to this process is presented in Fig.4.8 and Fig.4.9. To be able to generate the population of individuals, the GA contains three toolboxes:

- Population Creator- This toolbox receives the size of the desired initial population, as well as the chosen circuit to be optimized. It is also responsible to save each individual into a position of the population's list;
- Individual Creator- This toolbox is responsible to create each individual with the components size and fitness in a list. It is also responsible for returning each individual into the population creator list;
- Restricted Component Creator- This function is responsible for generating the size of each component for each individual in the population. It returns the components description of each circuit back to the Individual creator.

The representation operator is executed in the beginning of the optimization. The GA contains the size of the population to be generated, and will provide it to the Population Creator. This creator will ensure the Individual Creator is executed the same amount of times as the population size provided by the GA. After all the iterations are concluded,

the population's list is retrieved back to the GA which will contain the individuals, with the components size provided by the Component Creator, for the optimization.

To improve the algorithm's convergence rate, high rated individuals are also added to the population in certain optimization options. This individuals are also created by the same toolboxes but with specific sizing for the components.

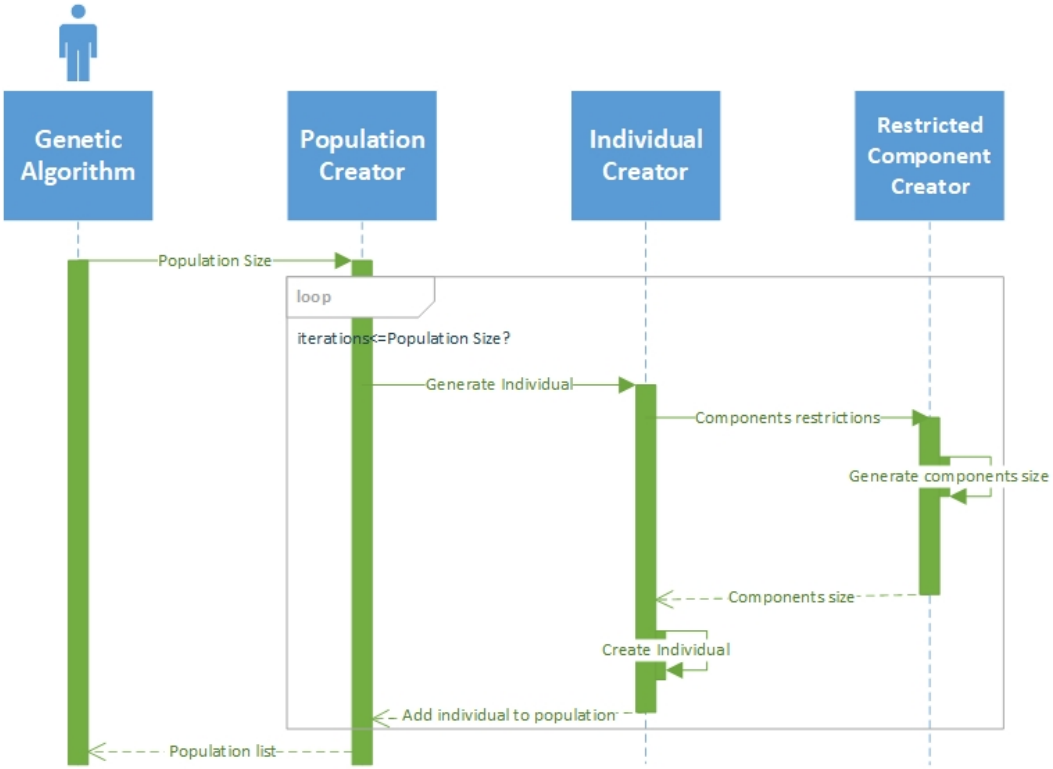

Figure 4.8: Representation-UML sequence diagram.

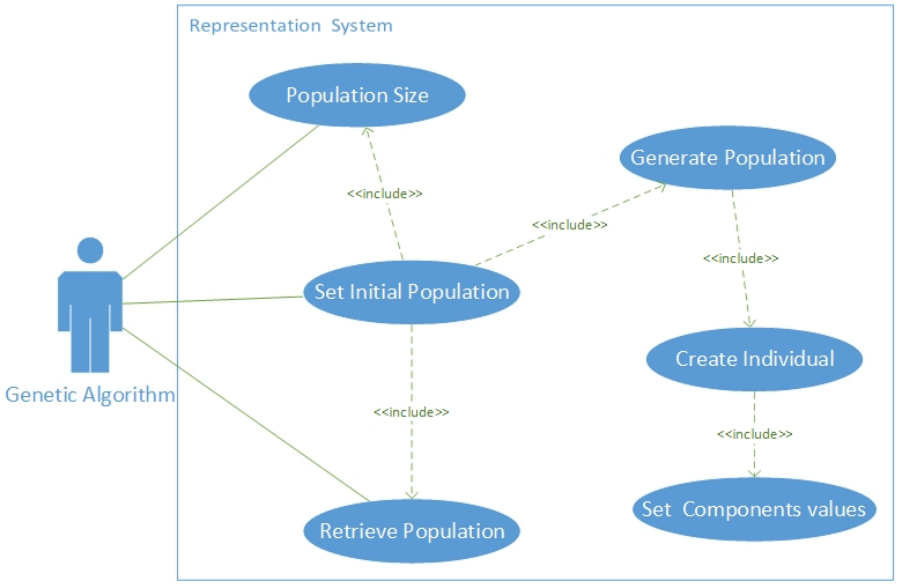

Figure 4.9: Representation-UML use case diagram.

4.2.2 Selection

The selection process is responsible for selecting the best individuals among the population in order to generate the new population. This is a very useful process to guarantee the convergence of the algorithm.

In this process the best individuals are chosen based on their respective fitness, where a tournament selection method is implemented. The tournament method selects three individuals and compares them with each other. The individual with the highest fitness value among the three is then selected. This process is repeated N times, where N represents the size of the actual population.

In order to guarantee a higher convergence rate, the best individual in the population is always chosen in the selection process.

Another taken consideration relates to the speed of the algorithm, where all the individuals selected from the tournament may result in an inadequate individual, $fitness = 0$, and if so the individual is deleted, which decreases the size of the chosen individuals to the crossing process, thus decreasing the size of the population. In Fig.4.10 is a flowchart related to the process.

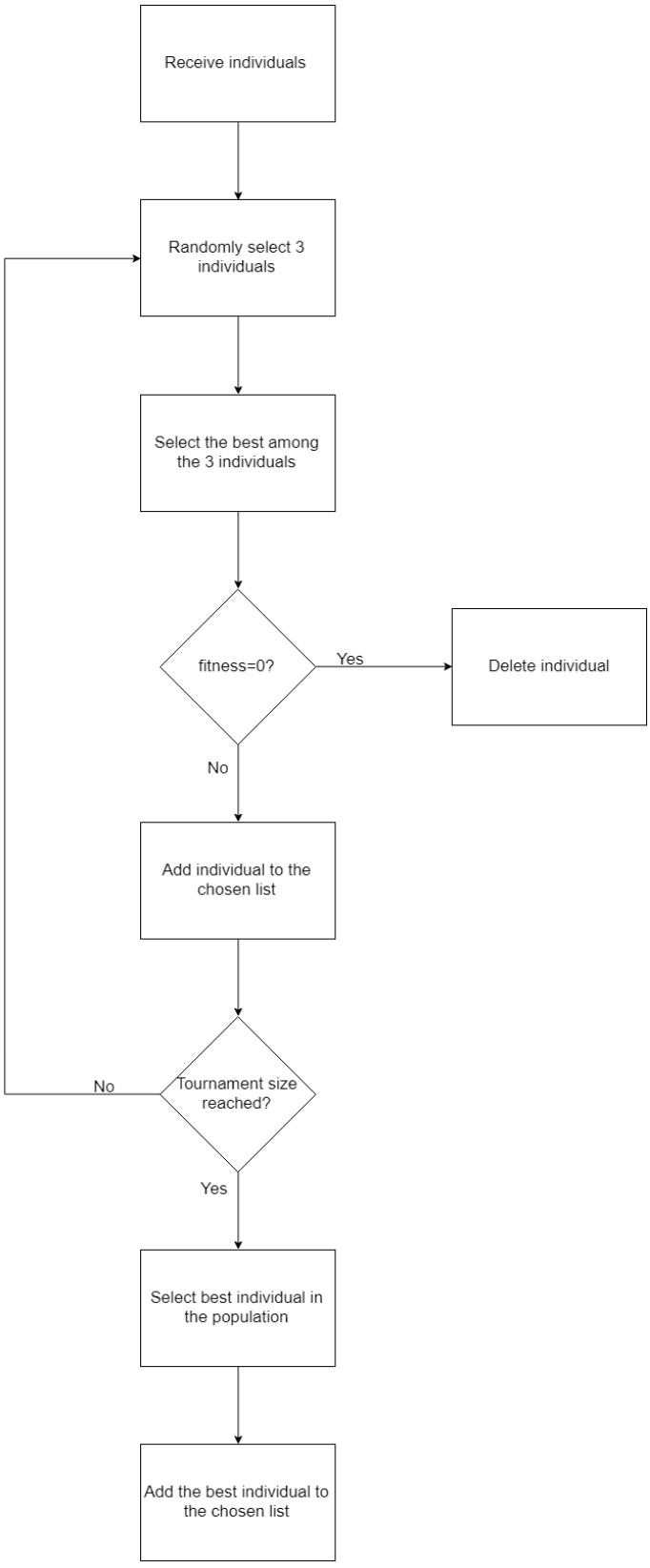

Figure 4.10: Selection process flowchart.

4.2.3 Selection operator Modulation

To establish a more closer to implemented operator, a closer representation of the process can be seen in Fig.4.11 and Fig.4.12. The selection process is implemented using four functions:

- Selection Toolbox- This toolbox receives the population and manages the other functions related to the selection operator. It is also responsible for creating a list with the selected individuals which may be used in the generation of the new population.
- Tournament- This function contains the implemented tournament method. In this case the tournament is done by selecting the highest rated individual among three randomly selected candidates. This tournament occurs N times, where N equals the size of the population.
- Best individual Selector- This function contains the necessary processes to select the most fitted individual among the population.
- Fitness Evaluator- This function is responsible for retrieving the fitness associated to each individual.

As it is possible to observe, the Selection Toolbox will manage the selected individuals from the other functions. Initially the selected individuals from the tournament will be added to the aspirants list. After the tournament is concluded the individuals with a non adequate fitness, $fitness = 0$, are deleted from the aspirants. After the list is composed, only with adequate individuals, the highest rated individual in the population is also added to the aspirants list. After the selection operator is completed the individuals in the aspirants will be further selected for the crossover and mutation operators, setting up the new generation of individuals.

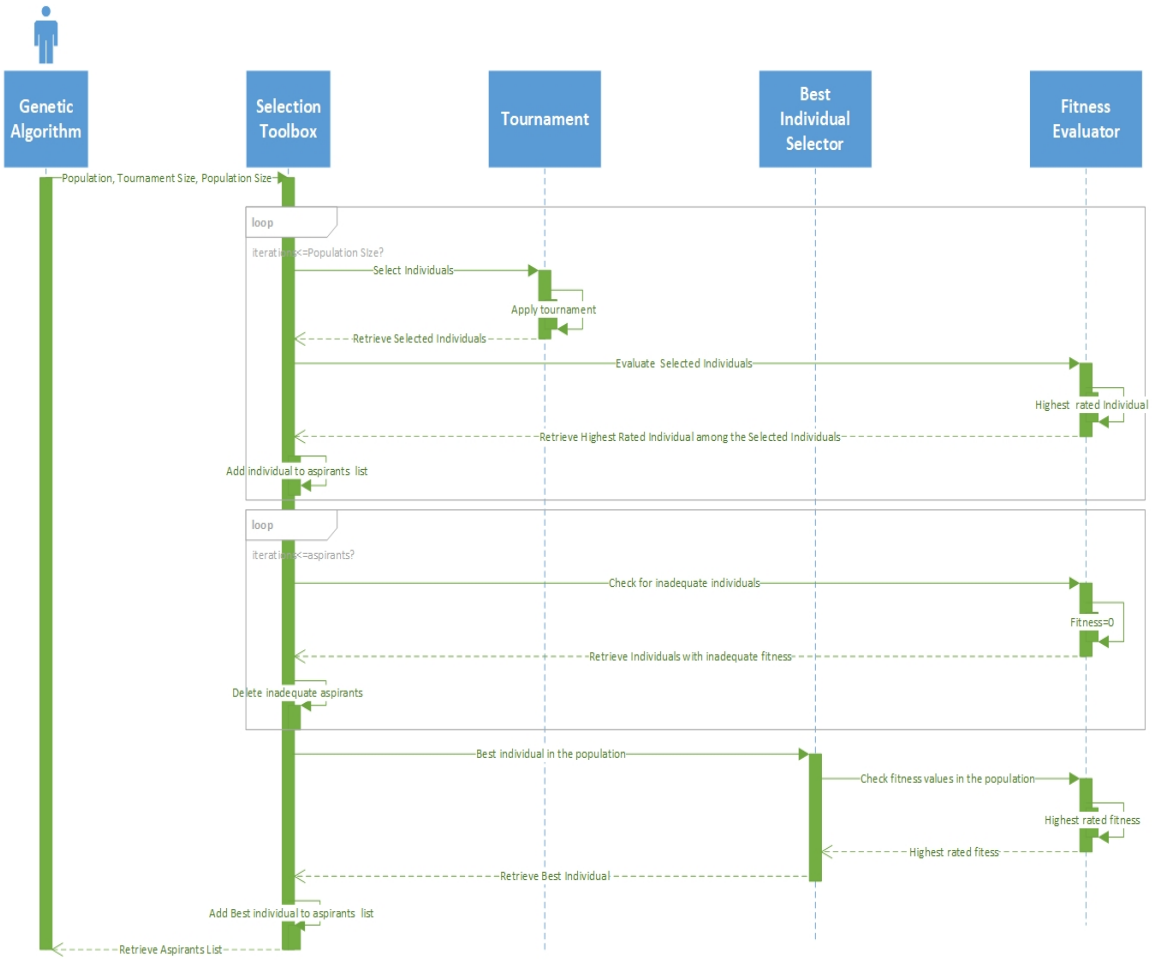

Figure 4.11: Selection-UML sequence diagram.

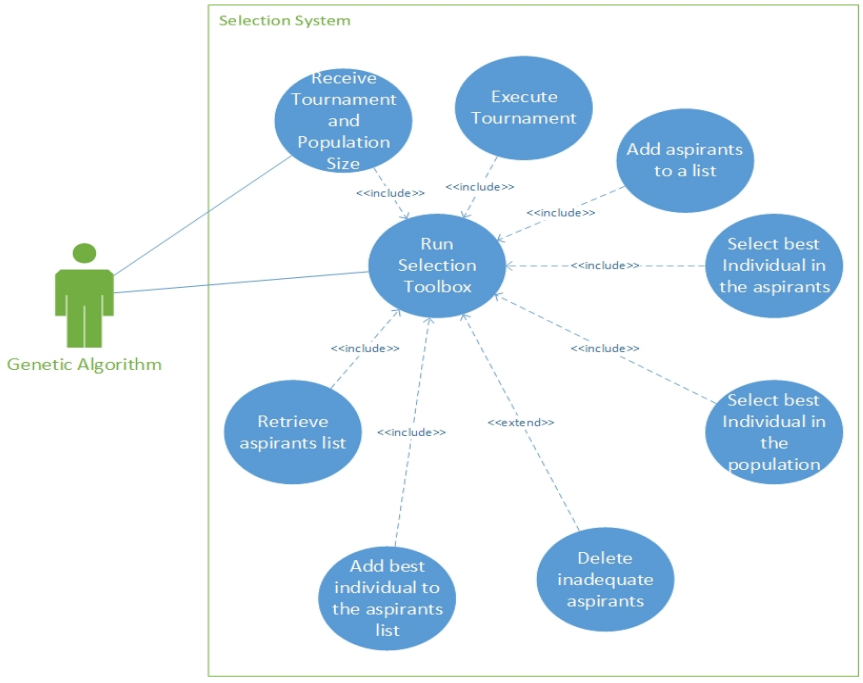

Figure 4.12: Selection-UML use case diagram.

4.2.4 Crossover

The crossover process consists in creating a new generation of individuals based on the previously selected ones. These selected individuals are called parents, and when crossed, create new individuals, called children.

Initially the aspirants list is cloned resulting in two similar lists, and one individual from each list is randomly selected. After both individuals are selected, both individuals are crossed, which consists in changing the dimensions of the components, if a certain component from the individuals' list is selected. Each position of the individuals' list is associated with a certain probability of being selected to be crossed with the other individual respective position, as illustrated in Fig.4.13. From this process two children are created, containing the characteristics of the two parents.

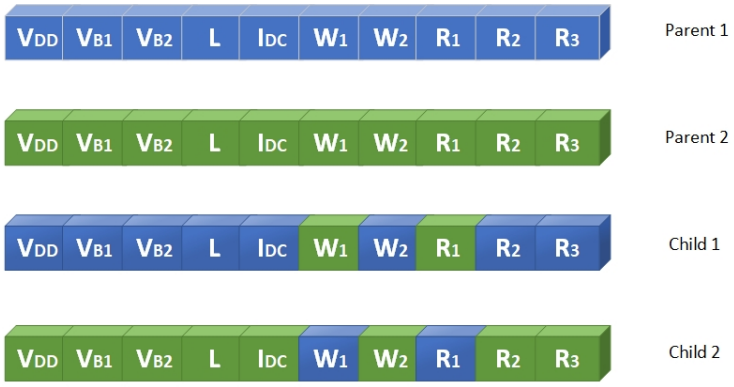

Figure 4.13: Crossover illustration.

Associated to the crossover operator is a probability which manages if the individuals are selected for crossing , in order to ensure the GA doesn't get lost in the search space. The crossover operator flowchart can be seen in Fig.4.14.

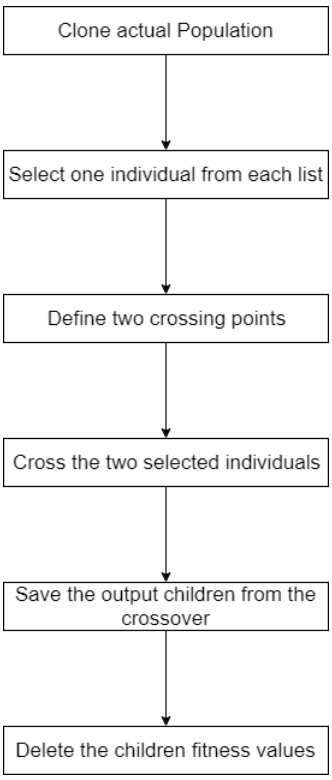

Figure 4.14: Crossover flowchart.

4.2.4.1 Crossover operator Modulation

To a further understanding of the crossover operator the UML sequence and use case diagrams are presented in Fig.4.15 and Fig4.16, respectively. This operator is composed by five functions:

- Crossover Toolbox- This toolbox manages the crossover operator by calling the other processes and storing the alterations in the individuals.
- Population Clone- This function clones the aspirants list, retrieving the equal ones.
- Individual Selector- This process is responsible for selecting one individual from each list, if the probabilistic restrictions are fulfilled.
- Individuals Cross- This process consists in crossing the two selected individuals and selecting the components to be crossed, retrieving two new individuals, called children.
- Fitness Evaluator- Function responsible for retrieving the fitness associated to an individual.

The crossover process is associated to a probability that selects if an individual will enter or not the mate process. If the individual in the aspirants list is select, the implemented toolbox will start executing the functions, previously referred, in order to generate the

new individuals after the mate process. In this project, considering the restrictions applied to each component, the selected crossing method is done point to point. In other words, each component can only be mate with another component of the same type. After the children is generate, each child will have its fitness deleted to be further updated have the simulations are executed.

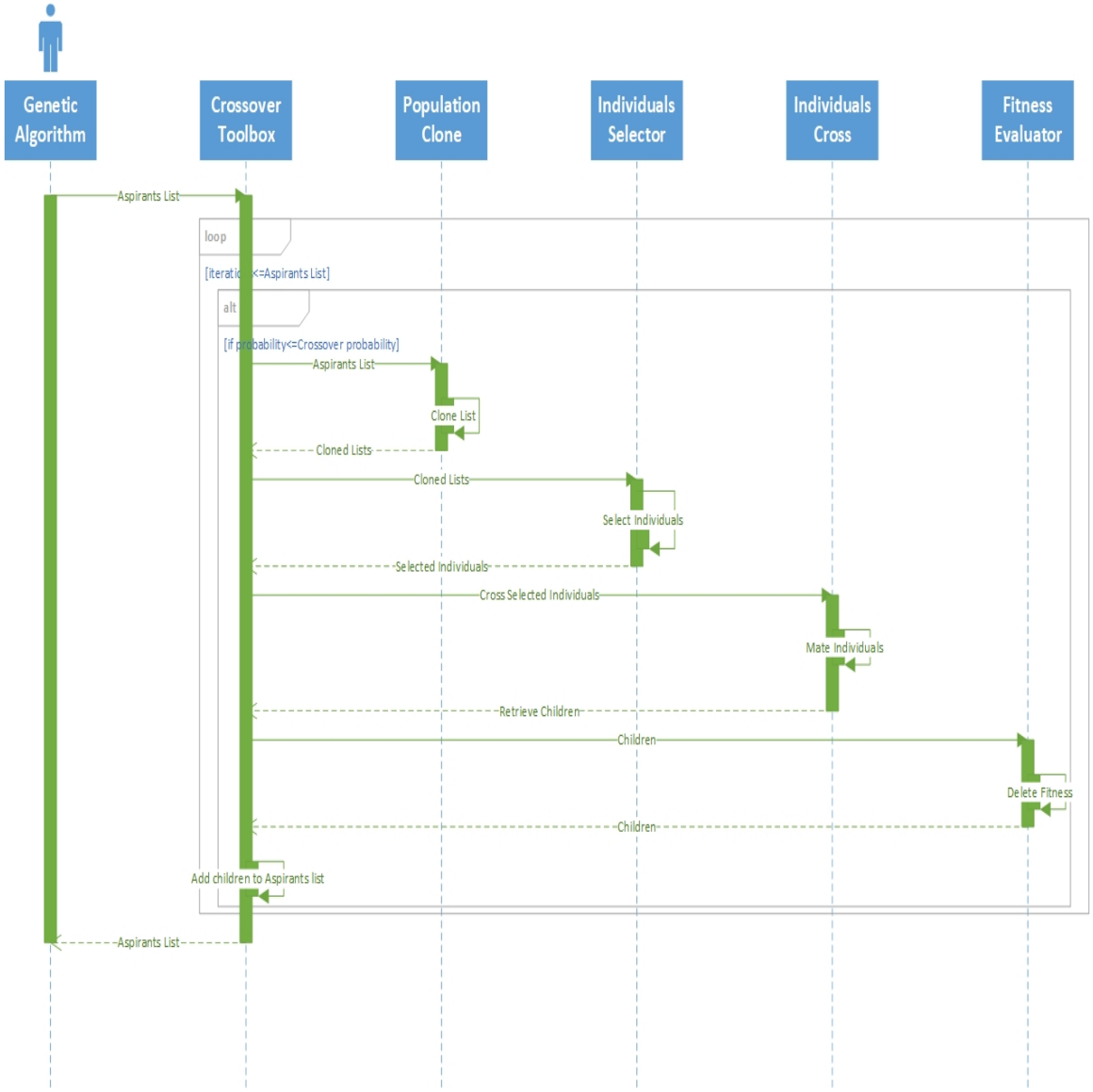

Figure 4.15: Crossover Sequence diagram.

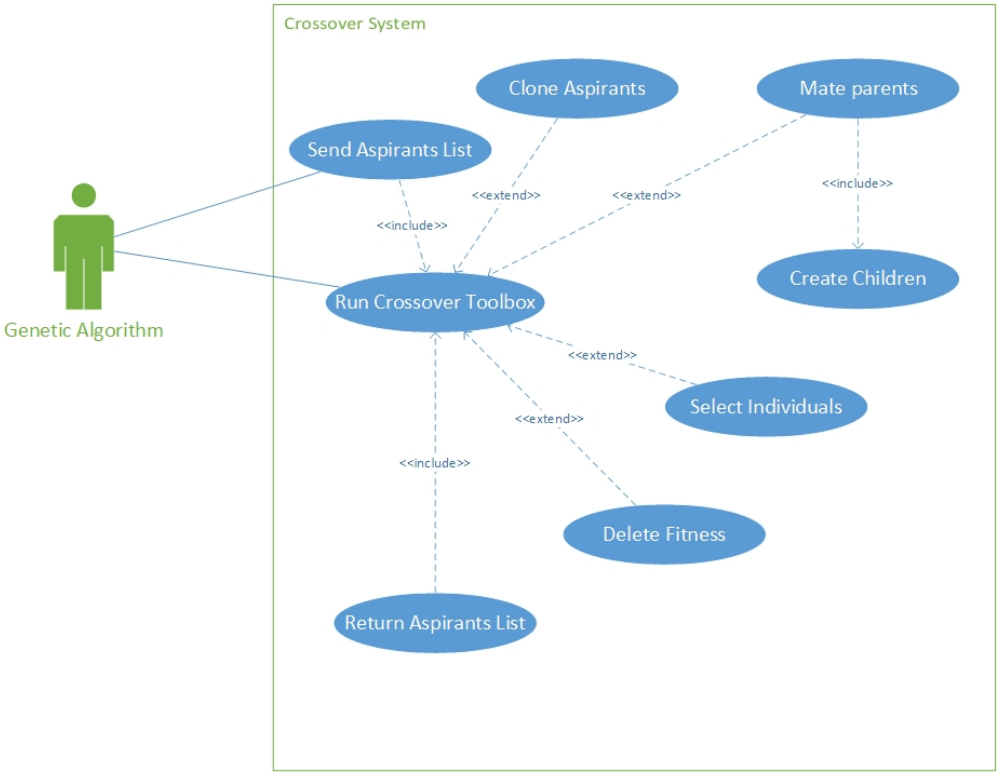

Figure 4.16: Crossover use case diagram.

4.2.5 Mutation

The mutation process, which is illustrated in Fig.4.17, is very useful to guarantee the diversity of the solutions. The process consists on:

1. Randomly choose individuals from the aspirants list for the mutation process with a certain probability;
2. If the individual is selected each component has a probability of being mutated;
3. The individual is mutated according to the restrictions applied to each component in the Representation process;
4. The mutated individuals' fitness is then deleted to be further calculated.

To increase the search speed in the space search for the optimum solution, a flag was additionally implemented which increases the mutation probabilities when the population tends to converge to a solution.

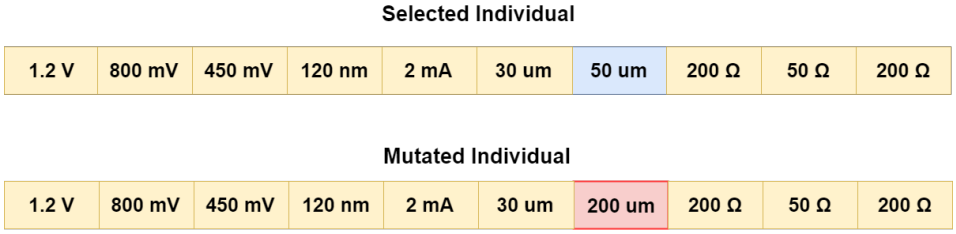

Figure 4.17: Mutation process illustration.

4.2.5.1 Mutation operator Modulation

Through a more detailed analysis is possible to observe in Fig.4.18 and Fig.4.19 that the mutation operator is implemented by four main objects:

- Mutation Toolbox- The toolbox manages the mutation process in the algorithm. It is also responsible for selecting if the individuals in the aspirants list are mutated or not.
- Component Selector- Function which identifies the components to be mutated.
- Restricted Component Creator- Creator to generate the new component size value.
- Fitness Evaluator- Function used to delete the fitness value from the mutated individual.

The mutation operator is, as previously referred, mostly managed by the Mutation Toolbox. Initially the toolbox receives the aspirants list and selects which individuals are going to enter the mutation process. If the individual is selected, there is a probability associated to its components of being mutated, where a new size value for the selected components will be then created. The mutated individuals' fitness is deleted to be further calculated, after the simulations are executed.

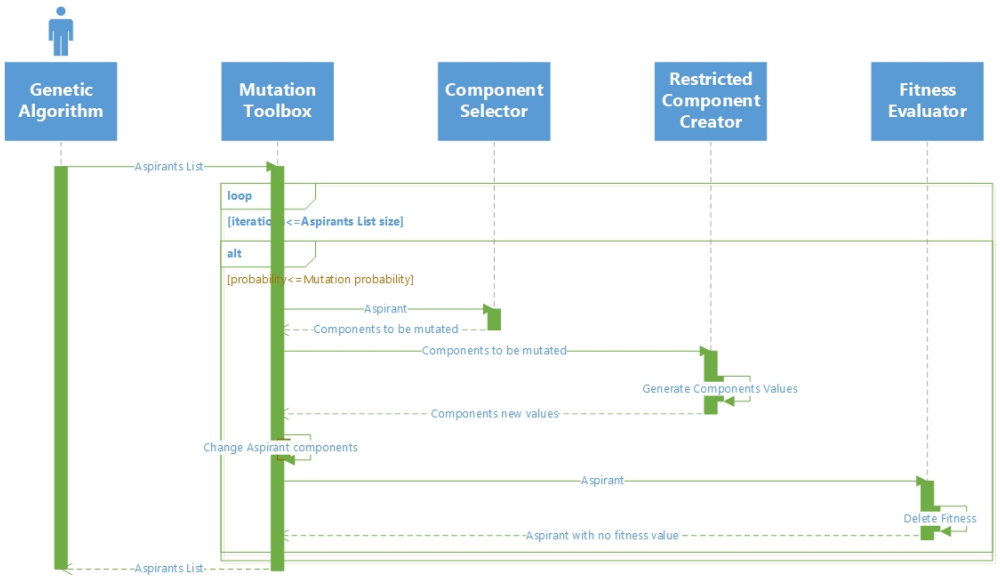

Figure 4.18: Mutation sequence diagram.

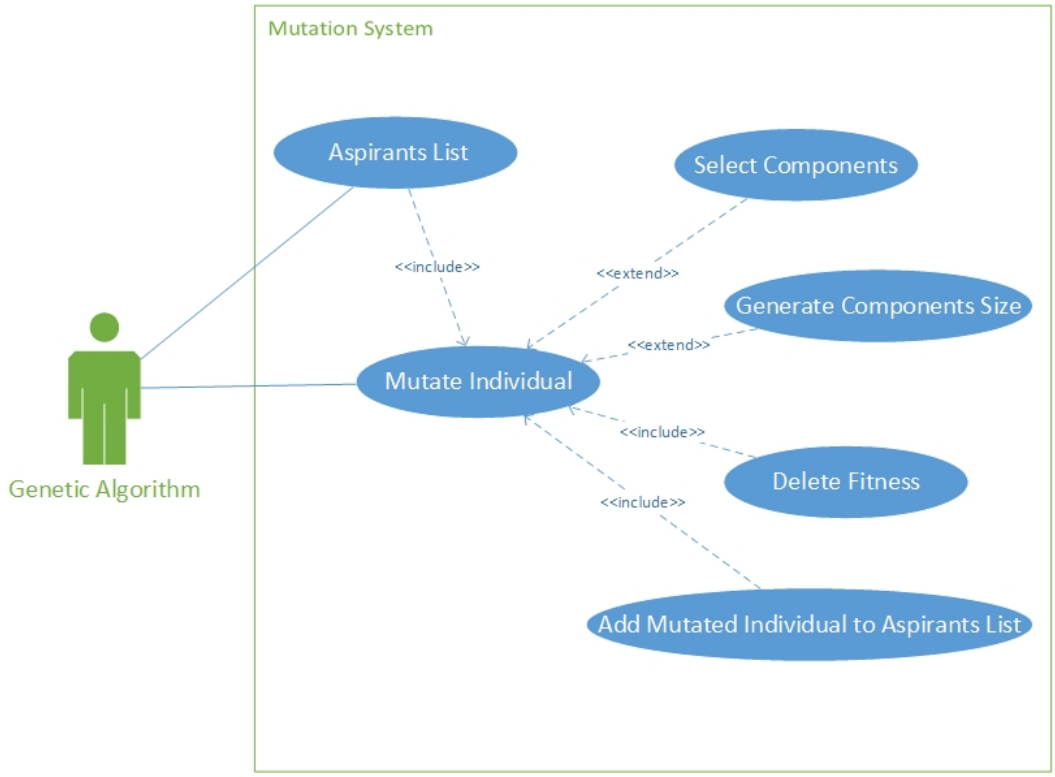

Figure 4.19: Mutation use case diagram.

4.2.6 Fitness calculation

To associate the simulation results with the GA it is necessary to create a managing operator capable of providing a rating association to each individual. This rating is given by the simulation output results provided by the Cadence Spectre implemented scripts. In order to establish the fitness association between the individuals among the population and the simulation results, *Figures of Merit* (FOMs), are used to distinguish a well adjusted sizing of the circuit to a not well adjusted one.

The fitness operator uses a list to store the simulation results. After the list is composed, the circuits sizing needs to fulfill certain defined restrictions. If the restrictions are not fulfilled, then the individuals' fitness is set to zero, considering it an invalid individual. On the other hand, if the circuits sizing passes the imposed restrictions, then the FOM is calculated an the result is associated to the respective individual. In Fig.4.20 is represented a flowchart of the fitness calculation operator.

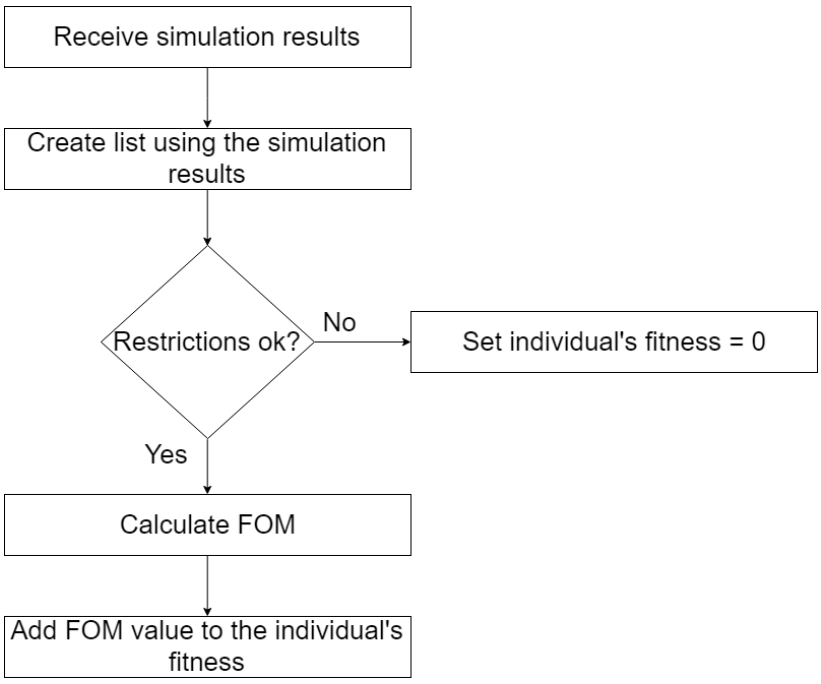

Figure 4.20: Simulation results management flowchart.

4.2.6.1 Restrictions manager

As previously referred, to check the sizing of the circuits, some minimum requirements are implemented in order to limit the search space of the GA. Therefore, the restrictions are based on the noise figure, impedance matching and the operating region of the transistors. These restrictions are also altered according to each simulation option, to match each circuit's characteristics.

In Algorithm.8 is represented an high level representation of the applied procedure to manage the restrictions.

Algorithm 8 Restrictions manager.

```
Receive chosen optimization option
Check restrictions
if restrictions ok then
   Calculate FOM;
end if
if restrictions failed then
   Set individual as invalid;
end if
```

4.2.6.2 Fitness calculation operator Modulation

Considering the diagrams in Fig.4.21 and Fig.4.22 it is possible to observe the fitness calculation operator contains three main procedures:

- Simulations List Creator- This creator manages the simulations' results and stores them into a list.
- Restrictions Manager- This functions is responsible for checking the main restrictions related to the circuit.
- Fitness Calculator- Function responsible for calculating the fitness of the individuals and associate it with the respective individual.

The fitness calculation operator is essential for the comparison of the individuals among the population. Therefore, the implementation of this operator needs to be simple and fast, since it is run numerous amounts of times.

Initially the simulations' results are stored in a list. After this process, restrictions are checked and the fitness is calculated and added to the respective individual. If an individual doesn't fulfill the necessary conditions, the fitness value will be equaled to zero, even if the fitness rating of the individual could result in a reasonable individual. This operator is executed for every individual in the population, and in every new individuals' generation.

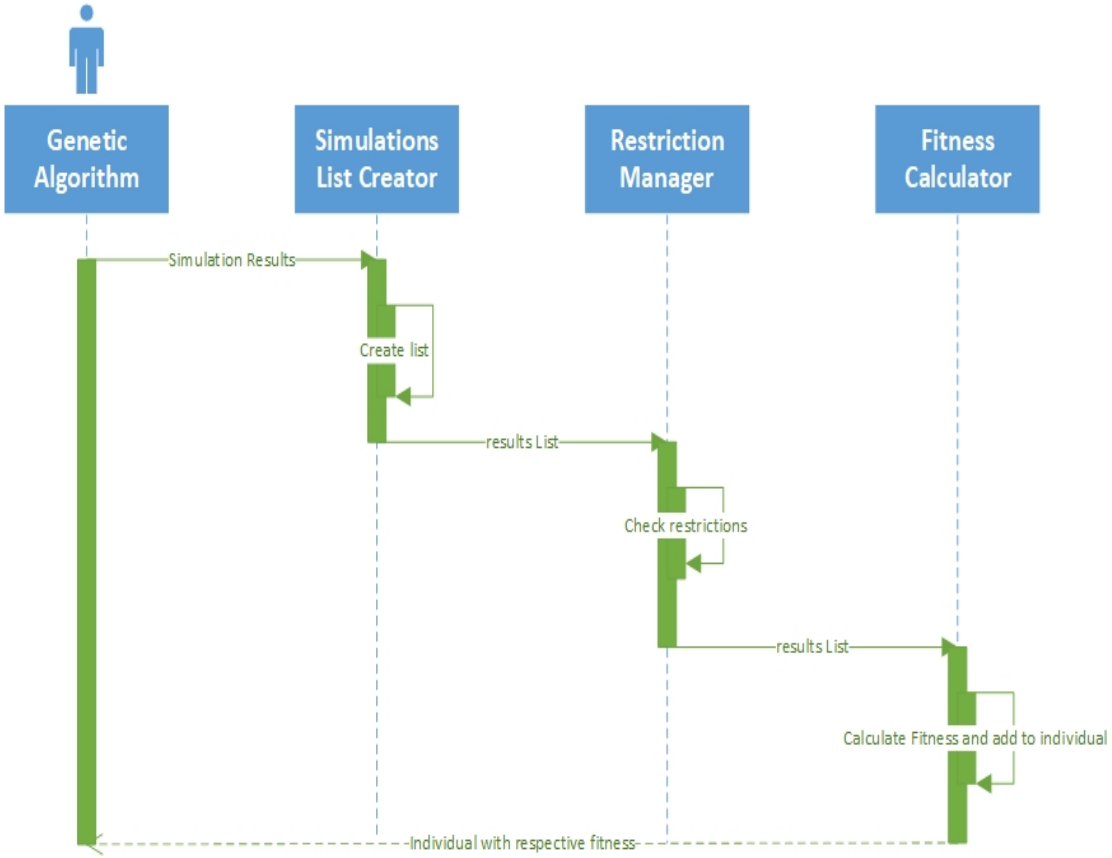

Figure 4.21: Fitness operator sequence diagram.

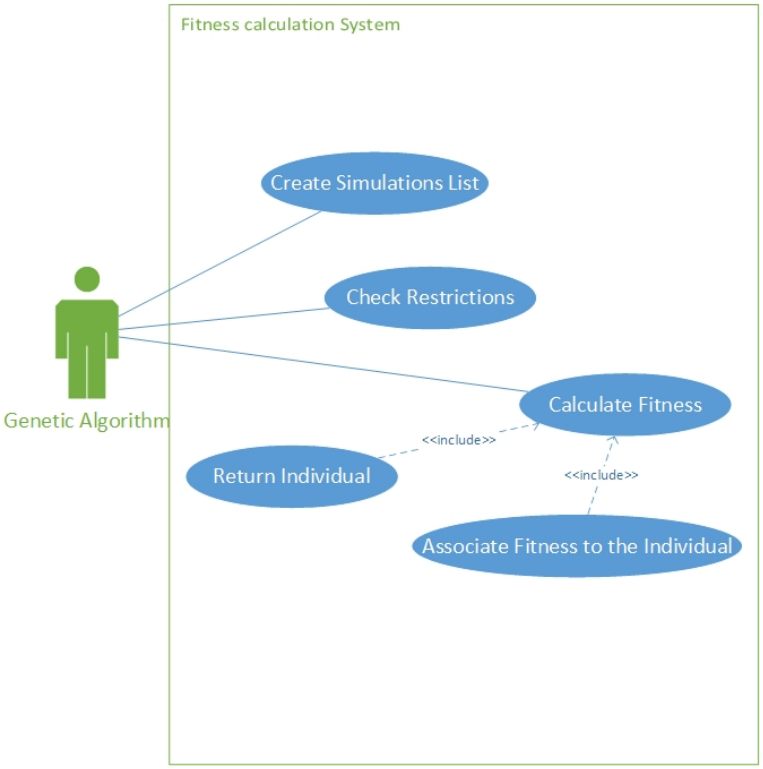

Figure 4.22: Fitness operator use case diagram.

4.3 Socket Communication for Cadence program configurations

The SOCAD documentation is responsible for establishing the connection between the developed algorithm and the Cadence program and server. To be able to guarantee the connection a client is implemented in order to allow an easier installation of the necessary software.

Considering the developed architecture three files need to be generated considering the circuit's characteristics when designed using Cadence. This files are provided from the Cadence design tools which allow the user to download an Ocean Script file containing the previously design architecture.

After the circuit is implemented and the Ocean Script file is then downloaded, the file needs to be split in three files for each architecture:

- Load Simulator file
- Variables file
- Run file

4.3.1 Load Simulator file

The "LoadSimulator.ocn"and "LoadSimulatorActiveLoads.ocn"files are responsible for loading the circuits' characteristics where each component is described and loaded according to the available libraries in Cadence.

The referred files contain the designed circuits' components and each characteristics as well as the netlist of the circuits.After the circuits are loaded it is also needed to load the simulations which will provide the results.

These files need to be loaded in the CIW the first time the optimization platform is initialized.

4.3.2 Variables file

The "vars.ocn"and "varsActiveLoads.ocn"files contain the dimensions of the components present in the circuits. These files are feeded from the optimization platform and change the variables values according with the selected individual from the GA.

Each time an individual is selected, each position from the individual contains the dimension of each component which will update the variables file. After this process is completed the simulations are ready to be executed.

4.3.3 Run file

The "run.ocn"and "runActiveLoads.ocn"files are responsible for giving the instructions for the simulations to be executed. After the simulations are loaded through the Load

simulator files, the simulations no longer need to be loaded. The simulations only need to be executed for each individual for each generation.

These files also contain the implemented functions which will allow to retrieve the necessary results from each simulation, to apply to the fitness of each individual which were previously presented.

4.3.4 Simulation Results file

To save the results of the simulations in a simple way a file called "sim_res.ocn"is overwritten every time a simulation is executed. This file contains the results from the functions and equations implemented in the Run file. Each result is written in each line of the file.

This results will then be sent to the GA which will calculate the fitness value from each respective individual.

Chapter 5

Used Low Noise Amplifiers

In this chapter will be presented the implemented LNAs for the optimization purpose, containing the circuits' study as well as the taken steps for their implementation.

When choosing the LNA topology important decisions need to be made. The first decision to take is, if the topologies are narrowband or wideband, which will influence the used optimization method. Another decision is related to single-ended and differential input, where the single-ended input simplifies the connection to most filters and antennas, and avoids the need of other stages responsible for the conversions from single to differential.

In this thesis, a well known wideband LNA will be implemented, where the main goal is to achieve a good sizing provided from the optimization platform which will be further presented.

5.1 Balun Low Noise Amplifier with Noise and Distortion cancellation

The presented circuit in Fig.5.1 consists in a well known inductorless LNA[1]. This circuit combines a common-gate stage and common-source stage, performing a balun operation allowing noise and distortion cancellation.

Considering the study through all the many years applied to this topology many ways to provide a better sizing trade-off in order to maximize the circuit's performance have been proposed.

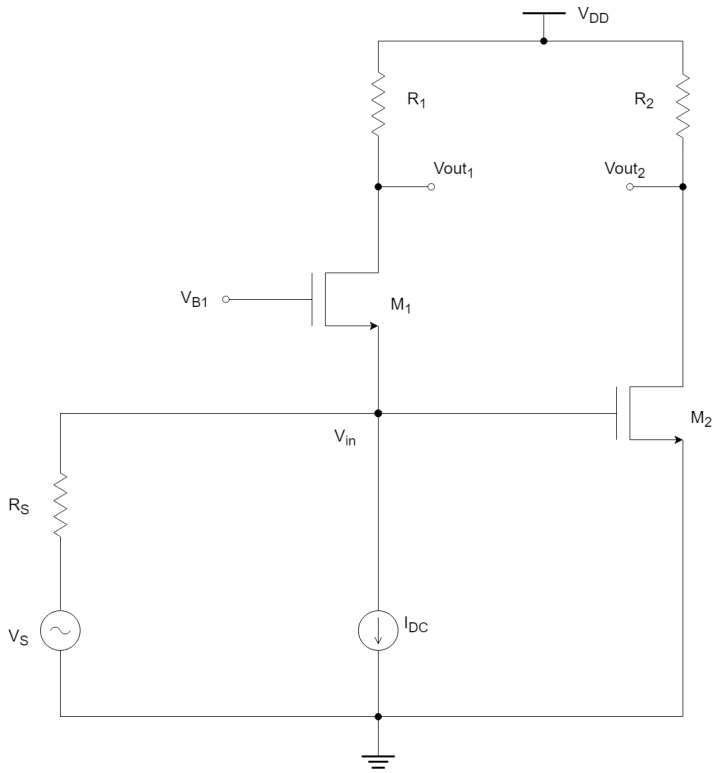

Figure 5.1: LNA with Noise and Distortion cancellation.

5.1.1 Balun Operation (Balancing)

Assuming ideal transistors with infinite output resistance, the common-gate stage biased with a current source, establishes a relation with its voltage gain, A_{VCG}, and its input impedance, Z_{in}. All the input current flows into the CG stage through R_1 originating the output signal, V_{out1}, with the same phase as the input signal v_{in}. Therefore the common-gate gain can be given by,

$$A_{VCG} = g_{m1} R_1 \tag{5.1}$$

and its input impedance given by,

$$Z_{in} = \frac{1}{g_{m1}} \tag{5.2}$$

where Z_{in} must be equal to R_S for ideal input matching conditions.

In order to obtain a proper balancing between the both stages of the circuit, they must have equal gains with different signs to achieve differential operation which results in,

$$A_{VCS} = -g_{m2} R_2 \tag{5.3}$$

which leads to,

$$A_{VCG} = A_{VCS} = \frac{R_1}{R_S} \tag{5.4}$$

5.1.2 Noise and Distortion Cancellation

The thermal noise generated by the CG stage can be represented by a current source, I_{n1}, which produces an input noise at V_{in}. The CS stage noise, when gain matching is considered with the CG stage, stands in opposition which allows the noise contribution of the CG transistor to be cancelled.

The presented architecture also allows the cancellation of the distortion produced by the CG stage. The CG transistor produces a nonlinear current, i_{ds}, dependent on the variations of V_{gs} and V_{ds}, which originates a nonlinear voltage at V_{in}.

The voltage V_{in} can be represented by a Taylor expansion dependent of v_s where,

$$V_{in} = \alpha_1 v_s + \alpha_2 v_s^2 + \alpha_3 v_s^3 + .. + \alpha_n v_s^n = \alpha_1 v_s + v_{NL} \tag{5.5}$$

where α_i represents the Taylor coefficients and v_{NL} the nonlinear voltages. The CG stage output voltage,V_{out1} can be expressed by,

$$V_{out1} = \frac{v_s - V_{in}}{R_S} R_1 = ((1-\alpha_1)v_s - v_{NL})\frac{R_1}{R_S} \tag{5.6}$$

and the CS stage output voltage can be expressed by,

$$V_{out2} = -V_{in}\frac{R_1}{R_S} = -(\alpha_1 v_s + v_{NL})\frac{R_1}{R_S} \tag{5.7}$$

This way all nonlinear voltages are cancelled by the differential operation where it only remains a linear voltage:

$$V_{out} = V_{out1} - V_{out2} = v_s\frac{R_1}{R_S} \tag{5.8}$$

Considering the transistors with an infinite output impedance, an ideal current source I_{DC}, and considering only the thermal noise of the resistors and transistors, the noise factor can be expressed by [1]:

$$F = 1 + \frac{\gamma g_{m1}(R_1 - R_S g_{m2} R_2)^2 + \gamma g_{m2} R_2^2 (1 + g_{m1} R_S)^2 + (R_1 + R_2)(1 + g_{m1} R_S)^2}{R_S A_V^2} \tag{5.9}$$

where $\gamma = 2/3$, the second part is the contribution from the CG stage, the third part is from the CS stage, the last part corresponds to the load resistors and A_V equals:

$$A_V = g_{m1} R_1 + g_{m2} R_2 \tag{5.10}$$

5.1.3 Linearity and Distortion of the CS-stage

As it was previously presented the CG-stage distortion can be canceled in the parallel of the CG-stage and CS-stage amplifier, therefore, the distortion of the total amplifier is given by the CS-stage behavior[1]. Since the modern CMOS are becoming more and more smaller, the output conductance cannot be neglected[][]. The drain current, i_{ds}, can be represented as function of V_{gs} and V_{ds}:

$$i_{ds} = g_{m1} V_{gs} + g_{ds1} V_{ds} + g_{m2} V_{gs}^2 + g_{ds2} V_{ds}^2 + g_{11} V_{gs} V_{ds} + g_{m3} V_{gs}^3 + g_{ds3} V_{ds}^3 + g_{12} V_{gs} V_{ds}^2 + g_{21} V_{gs}^2 V_{ds} + ... \tag{5.11}$$

where,

$$g_{mk} = \frac{1}{k!}\frac{\partial^k i_{ds}}{\partial V_{gs}^k}; g_{dsk} = \frac{1}{k!}\frac{\partial^k i_{ds}}{\partial V_{ds}^k}; g_{pq} = \frac{1}{p!q!}\frac{\partial^{p+q} i_{ds}}{\partial V_{gs}^p \partial V_{ds}^q} \tag{5.12}$$

Considering the drain source voltage, V_{ds}, depends on the current i_{ds} and on the load resistor R_2, it can be express by:

$$v_{ds} = c_1 V_{gs} + c_2 V_{gs}^2 + c_3 V_{gs}^3 + ... \tag{5.13}$$

with,

$$c_1 = -g_{m1}(R_2//(1/g_{ds1}))$$

$$c_2 = -(g_{m2} + g_{ds2}c_1^2 + g_{11}c_1)(R_2//(1/g_{ds1}))$$

$$c_3 = -(g_{m3} + g_{ds3}c_1^3 + 2g_{ds2}c_1c_2 + g_{11}c_2 + g_{12}c_1^2 + g_{21}c_1)(R_2//(1/g_{ds1})) \tag{5.14}$$

Applying the Taylor coefficients it is possible to define the IIP2 and IIP3 equations:

$$IIP2 = 20log_{10}\left(\left|\frac{c_1}{c_2}\right|\right) + 10dB$$

$$IIP3 = 20log_{10}\left(\sqrt{\left|\frac{4c_1}{3c_3}\right|}\right) + 10dB \tag{5.15}$$

Considering the IIP2 and IIP3 equations, a graph providing the variations through V_{gs} is possible to obtain as shown in Fig.5.2.Therefore, it is possible to conclude that the CS-stage can ensure good IIP2 and IIP3 values, +20dBm and +2dBm respectively, when V_{gs} is near 500mV.

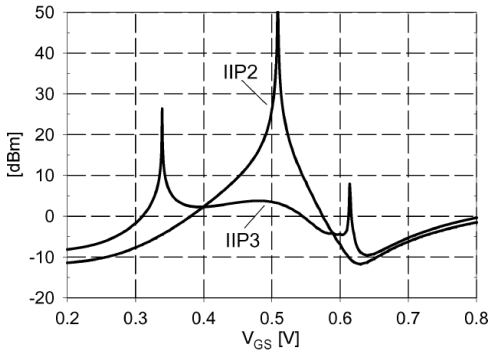

Figure 5.2: IIP2 and IIP3 versus V_{gs} for CS-stage from [1].

5.1.4 Circuit Implementation

To be able to achieve reasonable gain and low NF, small modifications were applied to the proposed architecture, as visible in Fig.5.3.

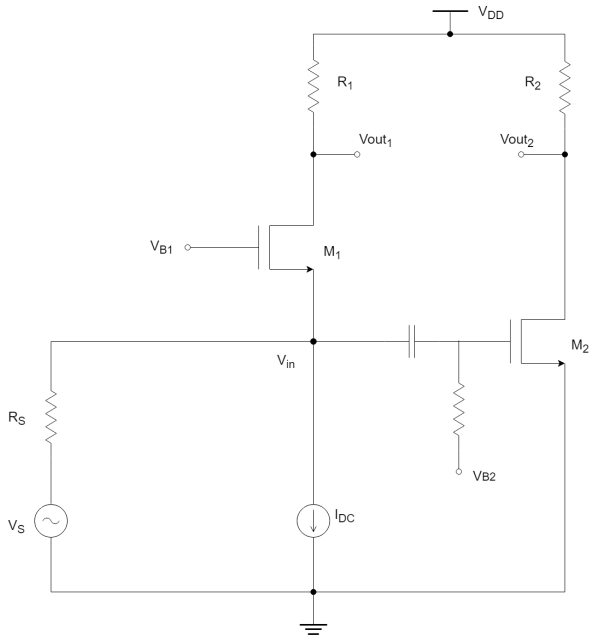

Figure 5.3: LNA with Noise and Distortion cancellation used in the simulation.

This changes allow a more flexible adjustment of the bias currents on the circuit,provided by V_{B1}, V_{B2} and a RC high pass filter, reducing the impact of the nonlinearities. Therefore, since the sizing of the components is adjusted by the optimization platform, it provides a bigger search space for the algorithm to provide a reasonable result.

To obtain a more realistic optimization approach, a converter from differential to single-ended architecture is also applied to the LNA during the simulations to ensure the output matching, and combining the differential outputs. The architecture is presented in Fig.5.4.

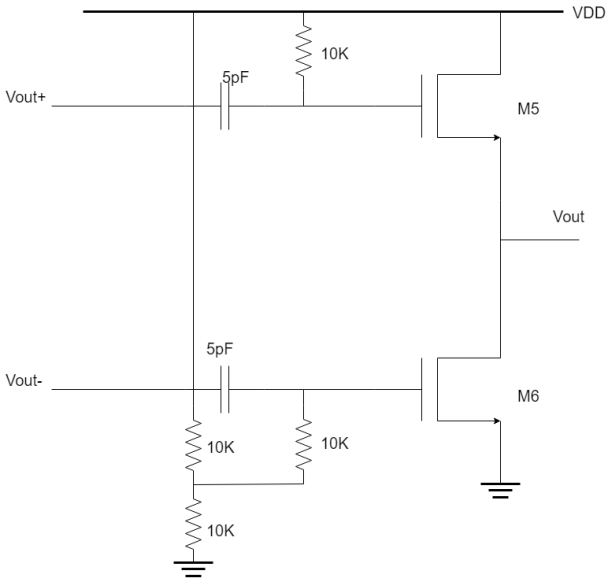

Figure 5.4: Differential to Single-Ended Buffer.

5.2 Low Noise Amplifier with Active Loads

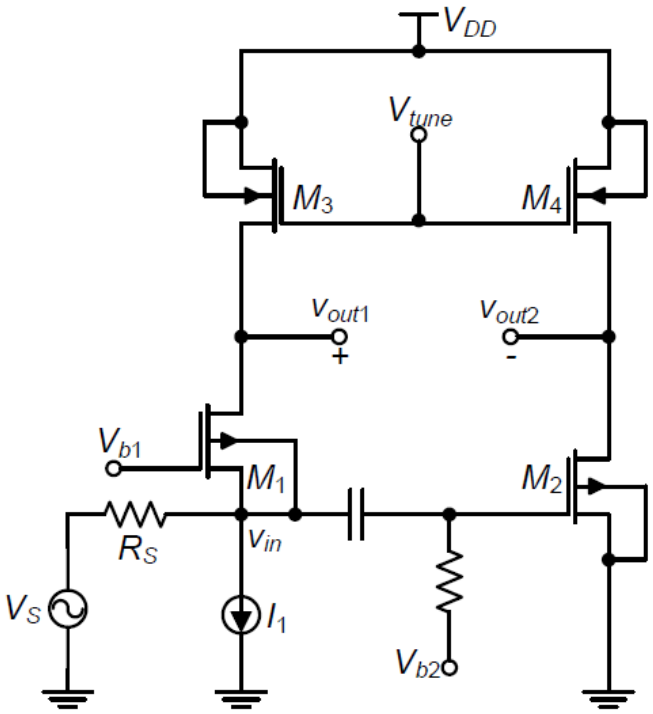

Figure 5.5: LNA with Active Loads adopted from [1].

In Fig.5.5 a LNA with Active Loads is proposed[10]. This architecture replaces the resistors from the circuit in Fig.5.3 with the transistors M_3 and M_4. By performing this small modification both gain and NF can be improved. In this architecture the voltage gain is controlled by the voltage source, V_{tune}, and the bias currents are defined by the voltage sources, V_{b1} and V_{b2}. The bulk and source of the CG-stage are also connected to reduce the body effect.

5.2.1 Circuit Implementation

After the resistor loads are replaced with the active loads it is possible to obtain better results from this architecture. To allow this better results the transistors M_3 and M_4 need to be in the triode region, without degrading the input matching of the circuit which, but also close to the saturation region to ensure that both differential voltage gains are matched for noise and distortion cancellation.

Considering the previously analysis it is possible to conclude that the low frequency gain of the circuit is given by:

$$A_{VLNA} = g_{m1}(r_{ds1}//r_{ds3}) + g_{m2}(r_{ds2}//r_{ds4}) \tag{5.16}$$

which will lead to a higher gain when compared with the load resistor circuit, if r_{ds3} and r_{ds4} values are bigger than R_1 and R_2 from Eq.5.10.

Considering the differential output gains are both matched in both circuits, it is possible to compare the noise factor in the circuits. Assuming $R_1 = R_2 = R_L$ and $g_{m1} = g_{m2} = g_m$ from the circuit in Fig.5.3, and if the flicker noise is taken into account, the noise factor from Eq.5.9 can be rearranged as[10]:

$$F = 1 + \frac{k_f}{8kTR_S C_{ox} f^{\alpha f}}\left(\frac{1}{W_1 L_1} + \frac{1}{W_2 L_2}\right) + \frac{\gamma}{2R_S g_m} + \frac{1}{R_S R_L g_m^2} \tag{5.17}$$

where k is the Boltzmann's constant, C_{ox} is the oxide gate capacitance per unite area, W and L are the transistor dimensions, T is the absolute temperature, γ is the transistors' excess noise factor, k_f and α_f are intrinsic process parameters dependent on the transistors size[]. Otherwise, the active load LNA also needs to take into account the flicker noise from the transistors M_3 and M_4. Once again if $r_{ds3} = r_{ds4} = r_{ds}$, the noise factor is given by:

$$F = 1 + \frac{k_f}{8kTR_S C_{ox} f^{\alpha f}}\left(\frac{1}{W_1 L_1} + \frac{1}{W_2 L_2} + \frac{1}{W_3 L_3} + \frac{1}{W_4 L_4}\right) + \frac{\gamma}{2R_S g_m} + \frac{1}{R_S r_{ds} g_m^2} \tag{5.18}$$

which proves the noise factor can get lower thanks to the bigger r_{ds} value when compared with R_L in the last part of the equation. This way, the noise figure may become lower with changing the value of g_m.

Another point to discuss relates to the distortion introduced by the transistor M_4. Since the distortion is introduced mainly by the CS-stage, the transistor M_4 would be also an issue, but its influence is negligible because it is in the triode region. Therefore, the main concern between gain and linearity trade-off relates to the V_{gs} of the transistor M_2, which needs to be biased from 400mV to 430mV for an ideal biased. This value may depend on the chosen application since it might be important to increase the gain while decreasing the IIP2 and IIP3 and vice-versa.

The previously analysis are very useful since the develop optimization platform will be used, mainly, to boost the figure of merit of the circuit considering only the CG-stage and CS-stage.

5.3 Cadence Spectre

In order to ensure more accurate results, the circuit is simulated using Cadence Spectre tools which runs in a no-graph option for a higher speed rate in each simulation.

Initially the circuit needs to be designed with the components variables to be optimized as shown in Fig.5.6.After the circuit is designed, the dimension of the components need to be set as variables in order for them to be replaced for the values given by the GA.

The following step is to set the simulations which will provide the results to feed the fitness calculation in the GA, using the ADE-L provided by the program. In this project two simulations are implemented:

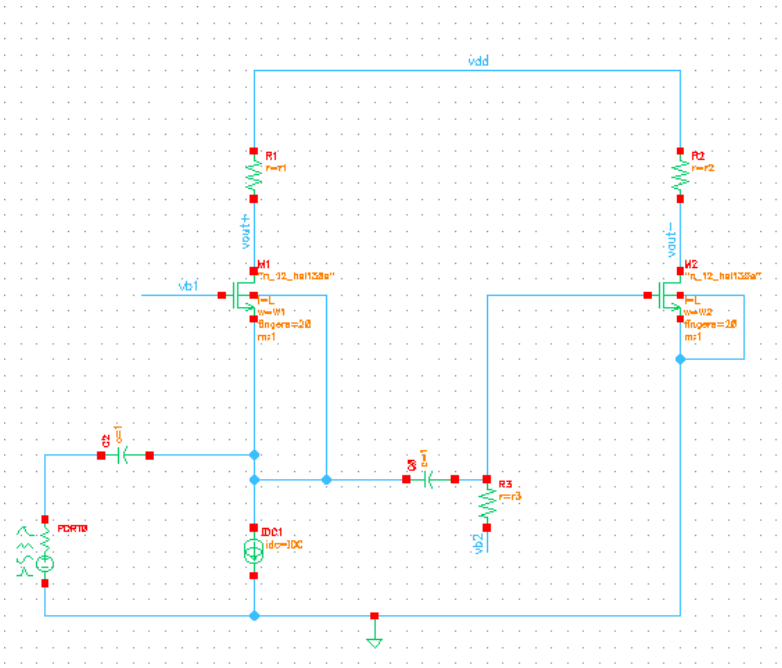

Figure 5.6: LNA circuit designed in Cadence.

- DC simulation
- S-Parameters and Noise Figure

5.3.1 DC simulation

The DC simulation is responsible for getting the consumption power from the circuit as well as the transistors' region. Cadence Spectre returns values according to the region the transistor is in. In this project the goal is for the transistors M_1 and M_2 to be in saturation which will be represented by the value 2. If the active load circuit is selected to be optimized the transistors M_3 and M_4 need to be in the triode region which is represented by the value 1.

The balun operation in the circuit is obtained through the difference between the trans-conductance of the transistors M_1 and M_2 multiplied by the ouput impedance of the circuit which is represented by R_1 and R_2, or by the trans-impedance of the transistors M_3 and M_4, as shown in,

$$\Delta_V = gm_1 R_1 - gm_2 R_2 \tag{5.19}$$

or,

$$\Delta_V = gm_1 rds_1 - gm_2 rds_2 \tag{5.20}$$

5.3.2 S-Parameters and Noise Figure

The S-Parameters includes the Noise Figure analysis and provide the most part of the results to feed the GA. These analysis are executed from 20MHz to 20GHz to minimize

the errors and provide:

- S_{11}
- Z_{11}
- S_{21}
- NF
- $Gain$
- BW_{min}

These analysis results are very important to achieve a reasonable fitness value for each individual. To obtain such results an Ocean Script file is implemented where Skill functions are used which are represented in Algorithm.9.

Algorithm 9 Used Skill functions.

S_{11min}=ymin(db(spm('sp 1 1)))
Z_{11max}=ymax(real(zpm('sp 1 1)))
S_{21max}=ymax(db(spm('sp 2 1)))
NF_{min}=ymin(db10(getData("F"?result "sp_noise")))
$Gain(f)$ == value(db(spm('sp 2 1)) f)
$Gain_{3dB}$=S_{21max} − 3
$BW_{Gain_{3dB}}$= cross(db(spm('sp 2 1)) $Gain_{3dB}$)
$BW_{S_{11}}$= cross(db(spm('sp 1 1)) -10)
$NoiseWave$=cross(db10(getData("F"?result "sp_noise")) 3.5 1 "either"t "cycle")
$NoiseVector$= drGetWaveformYVec($NoiseWave$)
BW_{Noise}= drGetElem(NoiseVector 1)

In order to get the circuits' bandwidth, the GA will then choose the minimum value between the $BW_{Gain_{3dB}}$, $BW_{S_{11}}$ and BW_{Noise}. It is important to refer that the Ocean Script file also contains some applied restrictions which ensure the provided results are reasonable, despite increasing its complexity.

Chapter 6

Simulations and Results

This chapter contains the description of the implemented simulations provided in the menu of the platform and also the results from the each optimization.

Each option contains a different approach to obtain a viable sizing for the circuit's components, and are organized according to the related issues found in each one. With this being said, it means that each presented optimization tries to solve the issues related to the previously developed ones, in order to improve the circuits' performance.

6.1 Low Noise Amplifier with resistor loads Optimization

This section contains the executed optimization to achieve an optimal sizing for the circuit presented in Fig.5.3. In[1] the authors conclude that there is a relationship between the transistors and resistors in the circuit that increases the gain and decreases the noise factor without affecting the linearity of the architecture.

The implemented optimizations are settle to obtain similar conclusions, using different simulation methods which are provided by the Cadence Spectre software, allowing a more accurate result, and combining them with the optimization platform to increase the search space.

To execute the necessary analysis, an output stage, Fig.5.4, combining both differential gains is also added to the circuit.

6.1.1 N-Points Optimization

Since the chosen circuit's architecture is a wideband-LNA, it is assumed that if the sizing fulfills the requirements for specific frequency values, then it will also fulfill then in the other frequencies. Therefore, the N-point optimization uses four frequencies (400MHz,900MHz,1.7GHz,2.4GHz).

Considering the main goal is to provide the simulations results from each frequency the used FOM, to provide the rating of the sizing, is given by:

$$FOM = \frac{1}{N}\sum_{i=1}^{N}\frac{S_{21}(f_i)[dB]}{(NF(f_i)[dB]-1)P_{DC}[W]} \tag{6.1}$$

where $f_i = [400M, 900M, 1.7G, 2.4G]$.

6.1.1.1 Restrictions

To limit the search space of the GA, restrictions are implemented relating to the simulations' results as well as the components size. In this specific optimization option the restrictions related to the size of the components are presented in Table.6.1 and Table.6.2, where it is possible to observe that only the transistors sizes are not specific values.

Table 6.1: Constant components size restrictions in N-Points Optimization.

V_{DD}	V_{B1}	V_{B2}	L	I_{DC}	R_1	R_2	R_3
1.2V	800mV	450mV	120nm	2mA	200Ω	50Ω	200Ω

Table 6.2: Variable components size restrictions in N-Points Optimization.

W_1	W_2
10-100µm	100-200µm

In Table.6.3 it is possible to observe the restrictions applied to the simulations' results which will ensure that inadequate individuals are deleted from the population. This restrictions are related to the operating point of the transistors and with the S-parameters analysis as well as the noise simulation provided by the Cadence.

Table 6.3: Analysis results restrictions in N-Points Optimization.

Analysis	**Restrictions**
S_{21}	$S_{21}(f_i)$>0dB
S_{11}	$S_{11}(f_i)$<-10dBΩ
Z_{11}	40Ω < $Z_{11}(f_i)$<60Ω
NF	$NF(f_i)$<3dB
M_1	M_{1OP}=2
M_2	M_{2OP}=2

6.1.1.2 Results

Considering the previously presented considerations and method applied to this optimization option, an initial population with 30 individuals is generated to provide a good amount of individuals capable of finding an adequate sizing for the circuit. The presented results are obtained after 200 generations and the best individual is presented in Table.6.4.

Table 6.4: Best individual in N-Points Optimization.

V_{DD}	V_{B1}	V_{B2}	L	I_{DC}	W_1	W_2	R_1	R_2	R_3
1.2V	800mV	450mV	120nm	2mA	48µm	157µm	200Ω	50Ω	200Ω

Relating the obtained sizing for the transistors in the best individual among the population it is visible that their sizes are not the same, $W_1 \neq W_2$. Establishing a ratio between the transistors size, $N_W = W_2/W_1$, and also between the previously defined resistors size, $N_R = R_1/R_2$, it is noticeable that the obtained dimensions ratio in the optimized components are way different, as shown in Table.6.5.

Table 6.5: Transistors and Resistors relationship in N-points Optimization.

N_W	N_R
3.3	4.0

Due to the imposed restrictions the search space of the GA is limited, since it is only responsible for providing the transistors sizes, but it is also very useful to provide the necessary conclusions regarding the N_W ratio. Since the restrictions don't allow the W_1 and W_2 to be equally sized, if $N_W = 1$ were beneficial the W_2 dimension would reduce to the lower imposed bound, which is not verified, since the W_2 keeps increasing to a size, leading to a N_W much closer to N_R.

Using the best individual sizing the analysis results used in the optimization process are presented in Table.6.6. In a more detailed perspective the full analysis is shown in Fig.6.1.

Table 6.6: Simulations results from N-points Optimization.

Analysis	**Results**
S_{21max}	11.58dB
S_{11min}	-25.9dB
IIP3	-3.7dBm
P_{DC}	6.5mW
NF_{min}	2.83dB
BW	7.16GHz
Δ_V	1.12dB
A_{VCG}	6.17dB
A_{VCS}	7.29dB
FOM	974

Considering the obtained results it is visible that both differential gains are not entirely matched, but are relatively close, due to the ratio between the resistors and transistors. Another noticeable aspect is related to the gain and noise improvement. In this optimization method, the linearity and bandwidth are not consider in the individuals' fitness calculation. Therefore, main goal of the GA is to boost the gain and decrease the

noise figure and power consumption. The power consumption being considered in the fitness calculation is really important to establish a more reliable sizing since it is directly associated with the noise figure, and it is probably why $N_W \neq N_R$.

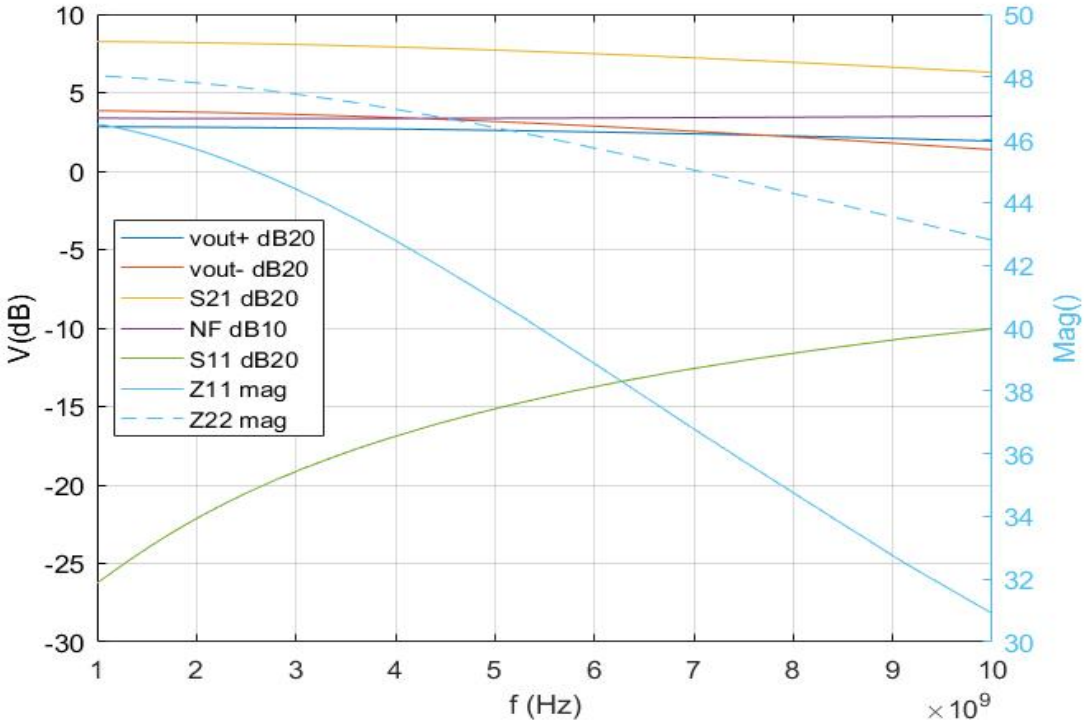

Figure 6.1: Simulation results for N-points Optimization.

6.1.1.3 Conclusions

The presented optimization method is based on the analysis results provided by N-points which are defined by the user. In the platform menu there are two presented options, where one only a single point is considered, which is more adequate to narrow-band LNAs, and the other considers four frequencies placed in a list.

Considering the selected topology to be optimized, the presented optimization method does not follow the typical processes used in conventional methods. Having this fact in mind, the N-points method, is not as reliable as the further described methods. Another issue in the previously provided results, are related to the low freedom given to the GA, since only the transistors size are being generated, and considering only 200 generations.

Although the issues related to the N-points method, it was still possible to obtain a functional sizing which fulfills the imposed requirements.

6.1.2 Wideband Optimization

The wideband optimization method is based on a more the typical approach to set a good fulfillment of the predefined requirements for the circuit's sizing, when compared with

the previous method. To implement this optimization method a few changes related to the FOM are elaborate where:

$$FOM = \frac{S_{21max}[dB] \times BW[GHz]}{(NF_{min}[dB] - 1)P_{DC}[W]} \tag{6.2}$$

which now considers the bandwidth of the circuit in the rating of the size of the components.

6.1.2.1 Restrictions

In this method the applied restrictions also take in consideration the previously provided study[1], providing a good polarization for good linearity results.

Considering the circuit's polarization, this time the main goal is to optimize the transistors and resistors size to find a good relationship between the components. Therefore, the applied restrictions are shown in Table.6.7 and Table.6.8.

Table 6.7: Constant components size restrictions in Wideband Optimization.

V_{DD}	V_{B1}	V_{B2}	L	I_{DC}	R_3
1.2V	800mV	450mV	120nm	2mA	200Ω

A more closer look to the components size restrictions shows that the transistors and resistors have larger bounds ensuring the GA a good increase of the search space. Applying such freedom, may lead the GA to find a not optimal solution. To go around this issue a flag system to increase the probabilities in the mutation and crossover processes are also added to the optimization.

Table 6.8: Variable components size restrictions in Wideband Optimization.

W_1	W_2	R_1	R_2
10-200µm	10-200µm	50-500Ω	50-500Ω

Considering the changes applied to the components, the considered analysis restrictions were maintained the same as the previous method. This restrictions will ensure the space search is reduced, especially due to the restrictions applied to the transistors operation point, that when combined with the imposed polarization for the components, will ensure the generated individuals sizing which aren't viable are further deleted from the population. In Table.6.9 there are presented the restrictions applied to the analysis provided from the Cadence Spectre.

Table 6.9: Simulation results restrictions in Wideband Optimization.

Analysis	**Restrictions**
S_{21}	S_{21max}>0dB
S_{11}	S_{11max}<-10dBΩ
Z_{11}	40Ω < Z_{11max}<60Ω
NF	NF_{min}<3dB
M_1	M_{1OP}=2
M_2	M_{2OP}=2

6.1.2.2 Results

To initiate the optimization, a population with 60 individuals is generated to provide an adequate sizing for the circuit. This optimization method was executed for 200 generations where the highest rated individual is presented in Table.6.10.

Table 6.10: Best individual in Wideband Optimization.

V_{DD}	V_{B1}	V_{B2}	L	I_{DC}	W_1	W_2	R_1	R_2	R_3
1.2V	800mV	450mV	120nm	2mA	46µm	85µm	321Ω	256Ω	200Ω

From the obtained best individual it is possible to observe that both transistors, M_1 and M_2, have now much closer sizing. This is due to the bigger bounds imposed in the restrictions allowing the M_2 width to be smaller. Despite the changes in the restrictions, it was expected to obtain a higher W_2 size, since the A_{VCS} depends on g_{m2}. This can be explained by the changes applied to the FOM which provides the fitness to each individual. By adding the circuit's bandwidth to the FOM, another trade-off between size and bandwidth is also taken into account for the circuit's performance.

In Table.6.11 there are presented the obtained relationship from transistors and resistors. Taking a closer look to the N_R value it is also noticeable that the relationship between R_1 and R_2 has also decreased when compared with the previous method. Despite this decrease in the N_R it is visible that both components increased in size providing a higher gain in the circuit, since A_{VCG} and A_{VCS} are directly dependent on the resistors sizing.

Table 6.11: Transistors and Resistors relationship in Wideband Optimization.

N_W	N_R
1.25	1.85

In Table.6.12 there are provided the obtained results from the executed analysis on the best individual among the population. From this analysis is visible that an improvement in the power consumption is achieved, as well as a slightly higher gain, despite the worse result in the IIP3 analysis, which it is expected considering the optimization doesn't not have in account the linearity effects.

Another visible issue relates to both differential gains, since the balancing operation between A_{VCG} and A_{VCS} does not occur, which explains the higher distortion effects. For a more detailed analysis of the circuit's performance, the executed simulations are shown in Fig.6.2.

Table 6.12: Simulations results from Wideband Optimization.

Analysis	**Results**
S_{21max}	12.67dB
S_{11min}	-26.5dB
IIP3	-4.32dBm
P_{DC}	5.43mW
NF_{min}	2.95dB
BW	7.73GHz
Δ_V	1.54dB
A_{VCG}	7dB
A_{VCS}	8.54dB
FOM	5795

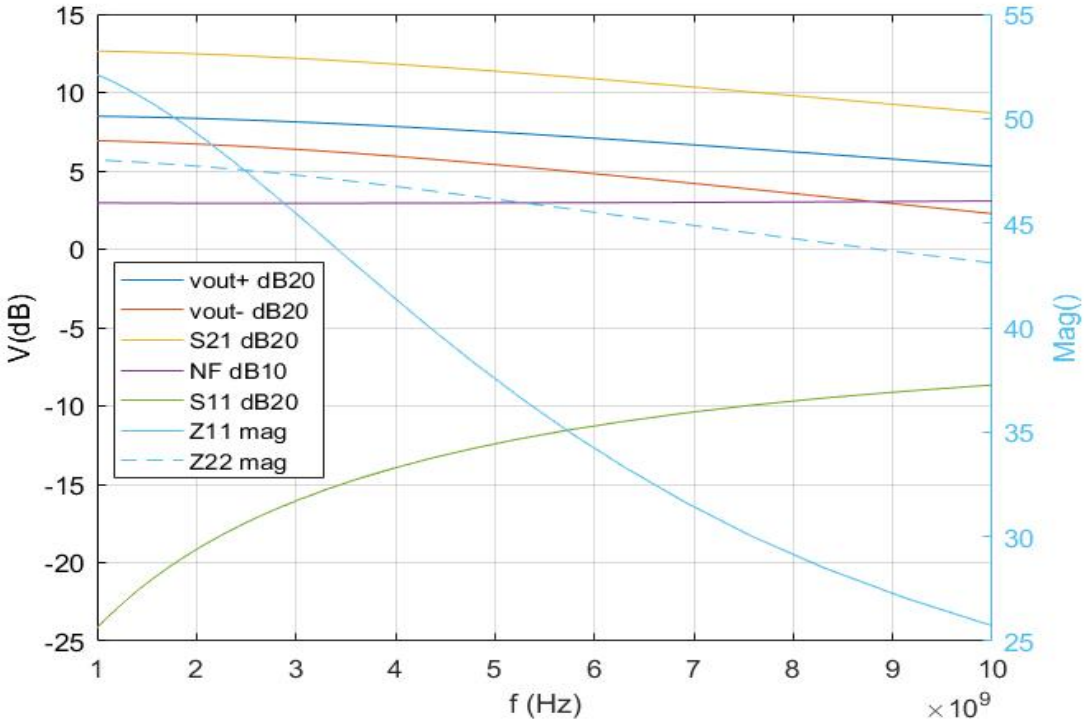

Figure 6.2: Simulation results for Wideband Optimization.

6.1.2.3 Conclusions

The wideband optimization method provides a good sizing for the circuit's components as it is shown previously. The obtained relationship between the transistors and resistors

size also provide a further knowledge and understanding of the circuit. It is noticeable that with the decreasing of the components relationship the power consumption decreases and the bandwidth increases, without affecting the circuit's gain and noise figure.

Despite the good results, the same issue regarding the balancing of the A_{VCG} and A_{VCS} continues to affect the results, since the balancing operation is not entirely fulfilled.

In conclusion, and having in mind the low number of generations applied to the test, the obtained sizing for the circuit's transistors and resistors shows good results, leading to the idea of a successful applied method.

6.1.3 Wideband N-factor Optimzation

The N-factor optimization method is based in the study provided in [1] where the size of the components are not randomly generated, and depend on a previously defined relationship. This relationship is defined by the N-factor and intends to achieve the balancing operation between the differential output gains.

Considering that the algorithm generates the R_1, W_1 and W_2 sizes, the size of the resistor R_2 is then defined by the N-factor relationship, where $N = W_2/W_1$. This means that $R_2 = R_1/N$.

To guarantee that this optimization method is viable, it is necessary to apply some changes to the crossover and mutation processes. If an individual is selected for both processes, then the selected components will always generate the other depended components based on the N-factor.

The applied FOM in this method is the same one applied in Eq.6.2.

6.1.3.1 Restrictions

The applied restrictions to the components size in this optimization method can be seen in Table.6.13 and Table.6.14.

Table 6.13: Constant components size restrictions in Wideband N-Factor Optimization.

V_{DD}	V_{B1}	V_{B2}	L	I_{DC}	R_1	R_3
1.2V	800mV	450mV	120nm	2mA	200Ω	200Ω

The GA will only be able to generate values for the width of the transistors M_1, M_2 and for the resistor R_2, between their respective bounds.

Table 6.14: Variable components size restrictions in N-Factor Optimization.

W_1	W_2	R_2
10-100µm	100-200µm	10-500Ω

The applied simulation restrictions are shown in Table.6.15 where an increase in the noise figure restriction, allowing more freedom in the search space, is applied when compared with the previous methods. The restrictions ensuring the circuit's input impedance

matching are also changed, in this case, for more restricted bounds, limiting the search space.

Table 6.15: Simulation results restrictions in Wideband N-factor Optimization.

Analysis	**Restrictions**
S_{21}	S_{21max}>0dB
S_{11}	S_{11max}<-10dBΩ
Z_{11}	45Ω < Z_{11max}<55Ω
NF	NF_{min}<3.5dB
M_1	M_{1OP}=2
M_2	M_{2OP}=2

6.1.3.2 Results

The obtained results from 200 generations with an initial population of 30 individuals are presented in Table.6.16.

Table 6.16: Best individual in Wideband N-factor Optimization.

V_{DD}	V_{B1}	V_{B2}	L	I_{DC}	W_1	W_2	R_1	R_2	R_3
1.2V	800mV	450mV	120nm	2mA	37µm	88µm	200Ω	84Ω	200Ω

Considering the obtained individual, it is possible to get the transistors and resistors relationship considering their respective sizing. It is noticeable that $N_W = N_R = N$ which is visible in Table.6.17.

In this optimization, and having in mind the few generations applied in the process, the N-relationship between the components stood in a relatively small value, when considering the numerous presented possibilities from[1]. This is due to the fact that, in this optimization, the FOM considers not only the circuit's gain and noise figure, but also the power consumption and bandwidth, which will lead the GA to find smaller components capable of fulfilling also these requirements.

Table 6.17: Transistors and Resistors relationship in Wideband N-factor Optimization.

N_W	N_R
2.38	2.38

Considering the best achieved individual in the optimization process, it is possible to get the analysis results, presented in Table.6.18.

Having in mind the presented analysis results, it is possible to visualize that the results suffered a huge decrease when compared with the previous Wideband optimization method. It is visible once again an increase in the power consumption, and worse results when considering both gain and noise figure. Once again the both differential outputs are not balanced which is not viable for the topology to eliminate the distortion effects. In Fig.6.3, there are presented the analysis results from Cadence Spectre.

Table 6.18: Simulations results from Wideband N-factor Optimization.

Analysis	**Results**
S_{21max}	8.28dB
S_{11min}	-29.50dB
IIP3	-2.50dBm
P_{DC}	6.73mW
NF_{min}	3.35dB
BW	9.99GHz
Δ_V	1.2dB
A_{VCG}	4.0dB
A_{VCS}	2.8dB
FOM	3175

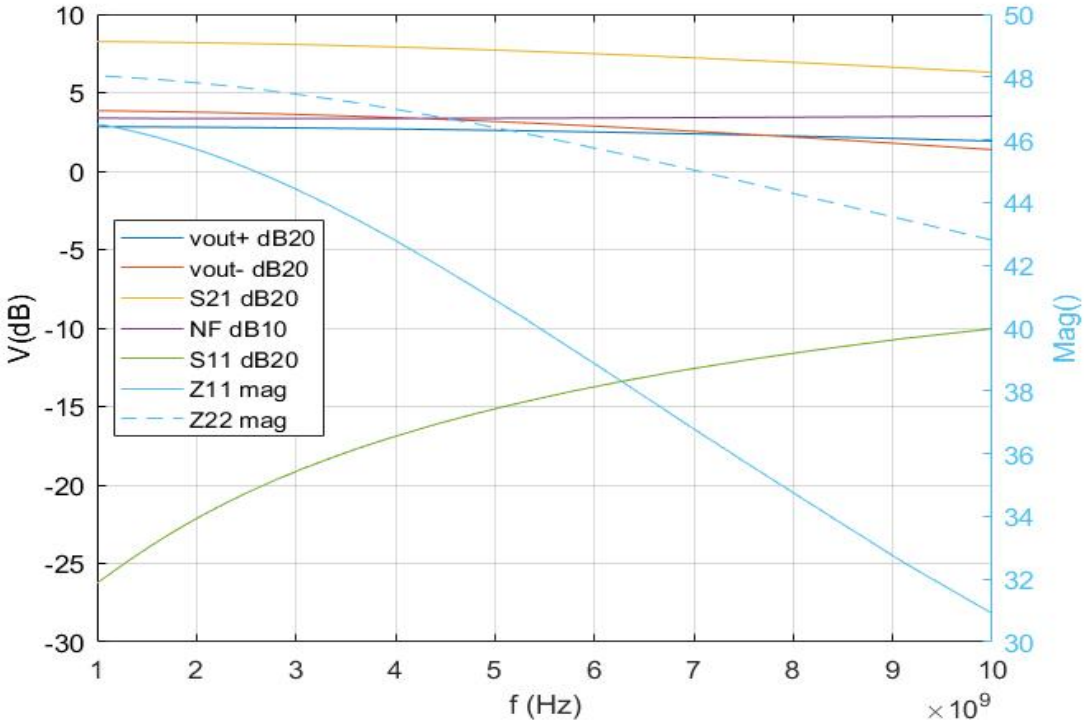

Figure 6.3: Simulation results for Wideband N-factor Optimization.

6.1.3.3 Conclusions

Considering the applied restrictions to the optimization method, the space search is very limited since the R_1 is also fixed. Therefore, the low rated result was expected as well as the gain decrease and noise increase.

Having in mind that the Wideband optimization method provided a good sizing for the topology, the objective for the N-Factor optimization method, was to provide a more reliable balancing operation between the differential outputs, which couldn't be achieved.

This occur because of the low generations used in the optimization process, which are not entirely conclusive.

Despite the bad results, conclusions regarding the components sizing can be taken according to the transistors dimensions, where it is visible the GA keeps searching for smaller W_1 and W_2, to increase the bandwidth and decrease the power consumption.

6.1.4 Wideband Balancing Optimization

The wideband balancing optimization method is similar to the wideband optimization method previously presented. The main of goal of this method is to ensure that both differential output are matched to each other, guaranteeing that the balancing operation in this topology is fulfilled.

To be able to guarantee the balancing operation in this optimization method, it is added to the FOM, responsible for the fitness calculation of each individual, the difference between both A_{VCG} and A_{VCS}. Therefore the new FOM is given by:

$$FOM = \frac{1}{1+|\Delta_V[dB]|}\frac{Gain_{max}[dB]\times BW[GHz]}{(NF_{min}[dB]-1)P_{DC}[W]} \tag{6.3}$$

where Δ_V is given by:

$$\Delta_V = A_{VCG} - A_{VCS} \tag{6.4}$$

6.1.4.1 Restrictions

Considering the similarities from this method to the wideband optimization method previously presented, the applied restrictions to this method are also the same ones as in the previous method. Once again the main goal is to optimize the transistors and resistors sizes, Table.6.20, applying the polarization from Table.6.19.

Table 6.19: Constant components size restrictions in Wideband balancing Optimization.

V_{DD}	V_{B1}	V_{B2}	L	I_{DC}	R_3
1.2V	800mV	450mV	120nm	2mA	200Ω

Table 6.20: Variable components size restrictions in Wideband balancing Optimization.

W_1	W_2	R_1	R_2
10-200µm	10-200µm	50-500Ω	50-500Ω

In Table.6.21 there are shown the applied restrictions to the analysis results and requirements, where this time a slightly greater noise figure is allowed, to match the obtained results from[1].

Table 6.21: Simulation results restrictions in Wideband balancing Optimization.

Analysis	**Restrictions**
S_{21}	S_{21max}>0dB
S_{11}	S_{11max}<-10dBΩ
Z_{11}	45Ω < Z_{11max}<55Ω
NF	NF_{min}<3.5dB
M_1	M_{1OP}=2
M_2	M_{2OP}=2

6.1.4.2 Results

The wideband balancing optimization method is initialized with a population of 60 individuals and the best individual is achieved after 200 generations, which is presented in Table.6.22.

Table 6.22: Best individual in Wideband balancing Optimization.

V_{DD}	V_{B1}	V_{B2}	L	I_{DC}	W_1	W_2	R_1	R_2	R_3
1.2V	800mV	450mV	120nm	2mA	39µm	63µm	284Ω	203Ω	200Ω

Once again is visible that the GA tends to search for solutions where the W_2 generated size doesn't stand to far from W_1. Since M_1 is responsible for the input matching it is normal to visualize that the obtained size for this transistors are normally around the same values.

In Table.6.23 there are presented both obtained relationship, N_W and N_R, where it once again shown that by adding the circuit's bandwidth and power consumption into the fitness calculation, the difference between M_1 and M_2 doesn't converge to a huge relationship.

Table 6.23: Results relationship in Wideband balancing Optimization.

N_W	N_R
1.40	1.62

Considering the best obtained sizing for the components, represented in Table.6.24 and Fig.6.4, it is possible to observe that both differential gains are now balanced. Despite the changes improved this aspect, it is also important to notice that the circuit's gain and noise figure performances have decreased. This may be related to the circuit's limitations according to the resistors loads. In a more detailed analysis it is visible that M_1 is limited for certain bounds related to the input matching, M_2 needs to be sized according to the balancing operation of the topology, and when considered the power consumption and bandwidth of the circuit it also needs to keep its bounds into small sizes. This leaves the loads with a very important role, by ensuring that the balancing operation does not affect the gain in the topology. The main issue with the resistors loads is related to their

voltage drop which limit the loads to get impedance values superior to approximately 350Ω when considering the used polarization or a low power consumption one.

Table 6.24: Simulations results from Wideband balancing Optimization.

Analysis	**Results**
S_{21max}	11.05dB
S_{11min}	-36.2dB
IIP3	-4.09dBm
P_{DC}	5.04mW
NF_{min}	3.25dB
BW	10.40GHz
Δ_V	0.06dB
A_{VCG}	6.18dB
A_{VCS}	6.12dB
FOM	6580

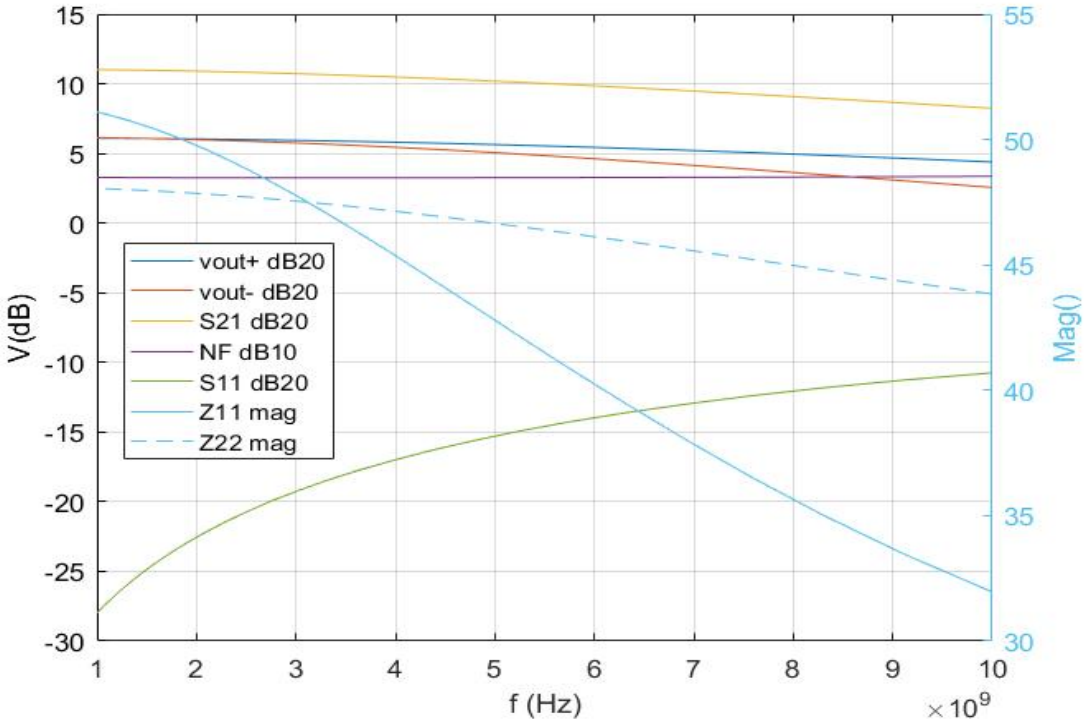

Figure 6.4: Simulation results for Wideband balancing Optimization.

6.1.4.3 Conclusions

Considering the previously presented results acquired from this optimization method it is visible that a well adjusted sizing is achieved.

Considering the applied changes to the FOM when establishing a comparison with the other optimization methods previously presented, it is visible that both differential

gains are now matched, thus reducing the distortion effect present in the circuit.

Another noticeable point is related to the relationship between the transistors and resistors sizing, where it is visible that M_1 and M_2, as well as R_1 and R_2 are much closer in terms of their dimensions, due to the balancing operation. This leads to a lower gain which can be overcome by providing some changes to the topology, which will be further shown.

6.2 Low Noise Amplifier with active loads Optimization

The presented section contains the related optimization methods applied to obtain a reasonable sizing for the circuit presented in Fig.5.5. Considering the previously study provided in[10], the main goal is to obtain the optimal sizing provided by the developed optimization platform.

To execute the necessary simulations in the optimization process, the output stage, presented in Fig.5.4, is also added to the circuit, combining the differential output into a single-ended output. Therefore, the simulations also consider the distortion and noise effects imposed by the output stage.

6.2.1 Wideband Optimization

The wideband optimization method follows the same ideology as the previously presented one in the LNA with resistor loads and uses the same FOM from Eq.6.3 to establish the balancing operation between the differential gains.

Considering the changes applied to the topology it is necessary to update some of the platform processes. In the Cadence management files it is necessary to update the circuit's characteristics file, which replace the resistors for active loads, where R_1 and R_2 are replaced for the transistors M_3 and M_4. Also the variables file to be generated are also altered in both GA and Cadence files which will provide the individuals and their respective simulations. All these changes are done through the menu option "Load Simulator for LNA with Active Loads".

6.2.1.1 Restrictions

The applied restrictions to the circuit's components are presented in Table.6.25 and Table.6.26, where it is visible that the transistors M_3 and M_4 are restricted to the the triode region, but also close to the saturation, as it was previously described in the circuit's description section.

The applied restrictions also consider the used polarization in the work developed from[10] to improve the linearity of the circuit.

Table 6.25: Constant components size restrictions in Wideband Optimization with Active Loads.

V_{DD}	V_{B1}	V_{B2}	V_{Tune}	L	I_{DC}	R
1.2V	800mV	450mV	400mV	120nm	2mA	200Ω

Table 6.26: Variable components size restrictions in Wideband Optimization with Active Loads.

W_1	W_2	W_3	W_4
10-200µm	10-200µm	18-35µm	18-35µm

Despite the bounds applied to the active loads, the Cadence analysis also verify is the transistors operation point fulfills the requirements, Table.6.27, for a proper sizing. The other analysis restrictions are maintained the same as the previous presented ones.

Table 6.27: Simulation results restrictions in Wideband Optimization with Active Loads.

Analysis	**Restrictions**
S_{21}	S_{21max}>0dB
S_{11}	S_{11max}<-10dBΩ
Z_{11}	45Ω < Z_{11max}<55Ω
NF	NF_{min}<3.5dB
M_1	M_{1OP}=2
M_2	M_{2OP}=2
M_3	M_{3OP}=1
M_4	M_{4OP}=1

6.2.1.2 Results

The results from this optimization method were obtained from 200 generations with an initial population of 60 individuals. In Table.6.28 is presented the highest rated individual from the optimization process.

Table 6.28: Best individual in Wideband Optimization with Active Loads.

V_{DD}	V_{B1}	V_{B2}	V_{Tune}	L	I_{DC}	W_1	W_2	W_3	W_4
1.2V	800mV	450mV	400mV	120nm	2mA	41µm	60µm	27µm	35µm

Analysing the best individual in the population it is noticeable, once again, that both M_1 and M_2 aren't that different in terms of their respective size. Through Table.6.29 it is possible to observe that the obtained sizing for the active loads, conducts to an impedance that could be easily obtained with the previous topology which uses resistors.

Table 6.29: Components relationship in Wideband Optimization with Active Loads.

R_{DS3}	R_{DS4}	N_W	N_{RDS}
286	205	1.39	1.46

Considering the best individual among the population, the simulation results from the applied simulations are presented in Table.6.30, where a great improvement in the FOM result is visible, when compared to the resistor loads circuit's results. It is also possible to observe that the output stage reduces the circuit's gain by 1.5dB. On the other hand a huge improvement in the linearity is achieved.

Table 6.30: Simulations results from Wideband Optimization with Active Loads.

Analysis	**Results**
S_{21max}	11.3dB
S_{11min}	-31.5dB
IIP3	1.93dBm
P_{DC}	5.4mW
NF_{min}	3.28dB
BW	7.86GHz
Δ_V	0.5dB
A_{VCG}	12.8dB
A_{VCS}	12.3dB
FOM	6965

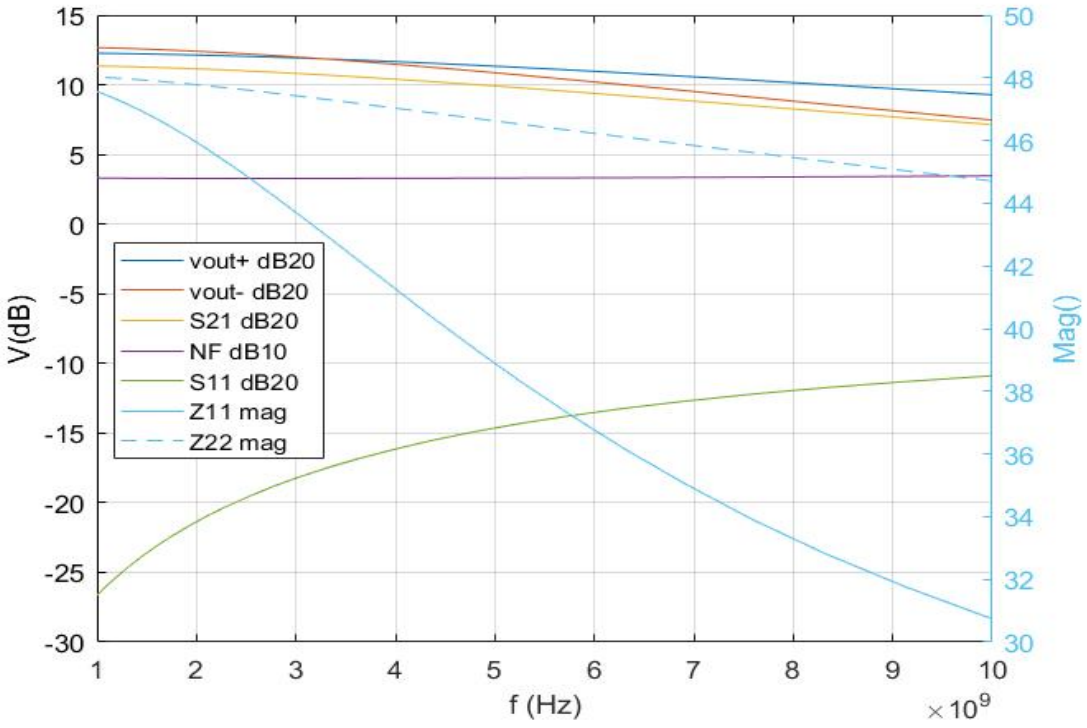

Figure 6.5: Simulation results for Wideband Optimization with Active Loads.

6.2.1.3 Conclusions

Considering the optimization method uses the same polarization provided from[10], where $V_{Tune} = 400mV$ which improves the gain and decreases the noise factor in the circuit, the obtained sizing for the components in the circuit are reasonable. With this polarization, the circuit's bandwidth and linearity is supposed to decrease, which confirms the results.

Having in mind the matching conditions restrictions it is shown that with a $N_{RDS} \neq 1$ there is also an improvement in the results. The balancing condition is also satisfied which limits the A_{VCS}, limiting the search space of the algorithm. Despite this considerations it is possible to conclude that the obtained sizing outdoes the requirements when compared with the LNA with resistors loads obtained sizing from the previous optimization methods.

6.2.2 Wideband Optimization with self-polarization

Considering the previously implemented optimization method, it is possible to conclude that the achieved sizing does not provide the highest performance from the optimized topology. Having in mind the higher potential from this circuit when compared with the presented topology from[1], a more exhaustive optimization was implemented, to be able to search for other sizing options.

The implemented Self-Polarization method follows the previously presented optimization method, but also tries to improve the circuit's polarization to achieve a higher performance rate.

6.2.2.1 Restrictions

In this optimization method, the GA has the liberty to change all the components to a limited restricted size, presented in Table.6.32, according to each component itself. There are also some constant restrictions which are presented in Table.6.31.

Table 6.31: Constant components size restrictions in Wideband Optimization with Polarization.

V_{DD}	L	R
1.2V	120nm	200Ω

Table 6.32: Variable components size restrictions in Wideband Optimization with Polarization.

V_{B1}	V_{B2}	V_{Tune}	I_{DC}	W_1	W_2	W_3	W_4
500-950mV	100-450mV	0-400mV	1-3mA	10-200µm	10-200µm	15-35µm	15-35µm

The applied restrictions to the simulations results applied in the optimization process are presented in Table.6.33, where it is visible that the used restrictions are the same as in the previous optimization method.

Table 6.33: Simulation results restrictions in Wideband Optimization with Polarization.

Analysis	**Restrictions**
S_{21}	S_{21max}>0dB
S_{11}	S_{11max}<-10dBΩ
Z_{11}	45Ω < Z_{11max}<55Ω
NF	NF_{min}<3.5dB
M_1	M_{1OP}=2
M_2	M_{2OP}=2
M_3	M_{3OP}=1
M_4	M_{4OP}=1

6.2.2.2 Results

The results in this method are obtained after 500 generations with an initial population of 120 individuals. In Table.6.34 is presented the obtained sizing of the best individual among the population, where it is possible to observe that there are a few changes related to the voltage sources V_{B1}, V_{B2} and V_{Tune}.

Table 6.34: Best individual in Wideband Optimization with Polarization.

V_{DD}	V_{B1}	V_{B2}	V_{Tune}	L	I_{DC}	W_1	W_2	W_3	W_4
1.2V	915mV	450mV	220mV	120nm	2mA	49µm	97µm	16µm	18µm

Considering the more freedom applied to the optimization process, it is visible that the active loads are much closer and more in the triode region, providing a higher impedance value, shown in Table.6.35, which will directly boost the circuit's gain. It is also noticeable that the V_{Tune} responsible for the polarization of the active loads also decrease which is supposed to boost the bandwidth and linearity in the topology.

Another important aspect related to the components sizes is the increase in the transistors M_1 and M_2 widths difference, and also the decrease in M_3 and M_4. This changes despite boosting the achieved gain, will also decrease the linearity and reduce the bandwidth of the circuit.

Table 6.35: Transistors and Gains relationship in Wideband Optimization with Polarization.

R_{DS3}	R_{DS4}	N_W	N_{RDS}
394	279	1.98	1.13

Having in mind the analysis results the taken conclusions from the obtained sizing are also verified in in Table.6.36. It is possible to observe that the topology performance

actually increased when compared with the previous implemented optimization methods. The main issue is related to the decrease of the bandwidth and also the linearity which as also decreased.

Table 6.36: Simulations results from Wideband Optimization with Polarization.

Analysis	**Results**
S_{21max}	13.7dB
S_{11min}	-31.2dB
IIP3	-5.81dBm
P_{DC}	4.8mW
NF_{min}	2.8dB
BW	6.4GHz
Δ_V	0.1dB
A_{VCG}	14.8dB
A_{VCS}	14.7dB
FOM	9614

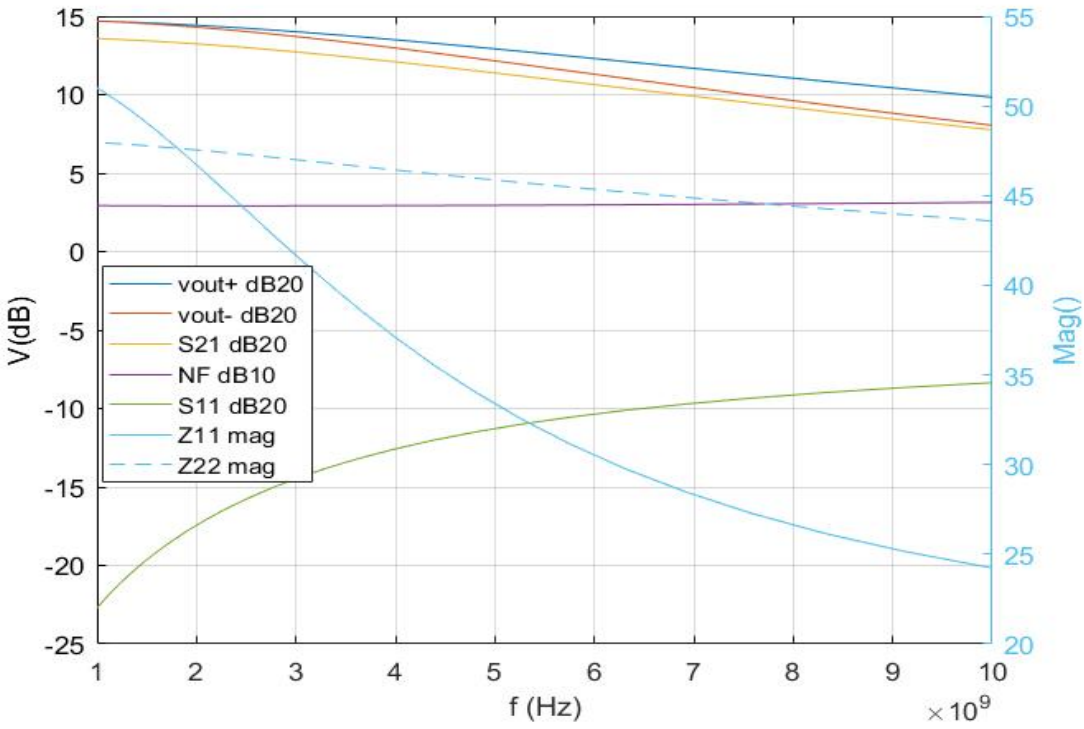

Figure 6.6: Simulation results for Wideband Optimization with Self-Polarization.

6.2.2.3 Conclusions

Considering the presented method and establishing a brief comparison between the other implemented optimizations it is noticeable that, by introducing a self-polarization into account, also increases the sizing rating. Although the search space increases widely and

the number of iterations to obtain the optimal solution increases, the number of local maximums also increases and gives more freedom to the GA to be able to find a adequate sizing. The most important change in the polarization is due to the new sizing for the V_{Tune} voltage source, which is responsible for the polarization of the active loads. By decreasing its size, the gain and noise figure also decrease which takes the obtained results to average ones. Another issue is related to the increase of the components size, that decreased considerably the bandwidth of the architecture when compared with [1][10]. This shows that the GA has to deal with a lot of trade-offs that are not easy to fulfill, in such a short number of iterations.

Having in consideration the low number of generations used by the GA, it is very easy to conclude that it is almost impossible to achieve the best possible solution. Hypothetically, the developed platform would be able to achieve a solution capable of outdoing the State-of-the-Art presented optimizations, but when considering the amount of variables and solutions in the search space it is set as impossible when not having more computational power.

Taking in account the changes provided through all optimization methods which were previously presented, it is possible to establish a comparison between the State-of-the-Art optimizations and the best obtained in all the presented methods, which is represented in Table.6.37.

Table 6.37: Comparison of wideband balun LNAs optimizations.

Ref	BW [GHz]	Gain [dB]	NF [dB]	Power [mW]	IIP3 [dBm]	balancing?	Gain imbal. [dB]	V. Sup.	Tech.
This Work	6.4	14.8	2.8	4.8	-5.9	Yes	<0.5	1.2V	130nm
[1]	5.2	13-15.6	<3.5	21	>0	Yes	<0.7	1.2V	65nm
[10]	5-10	12.4-20.2	2.6-3.2	4.8	-10.9 - 0.7	Yes	<1.2	1.2V	130nm

From Table.6.37 is possible to observe that the obtained sizing stands in line with the other related works. If the necessary number of generations to achieve the optimal solution were able to achieve in a reasonable spent time, it is almost implicit that the GA would be able to find a higher rated solution that the presented one.

In a brief summary for the obtained sizing result it is possible to verify that the proposed platform may ensure a good and viable solution for the optimization process, if computational power and spent time isn't a major issue.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Considering the recent years, it is noticeable the increase of evolutionary algorithms applied to optimization problems. In this thesis, a GA is proposed to optimize a CMOS wideband LNA with noise cancelling[1][10].

Having in mind the previous studies around the circuit to optimize it is visible that a relationship between the transistors and loads sizing can be altered for gain and noise improvement purpose. Therefore, different optimization methods were implement in order to obtain an optimal sizing for the components.

In chapter 5 are provided the implement optimization methods, as well as the obtained results from each one. Establishing a comparison between the methods it is possible to conclude:

- N-Points Optimization- In this method only specific frequencies are considered for the optimization purpose. The implemented simulations only consider the obtained results from the selected frequencies, not providing the most accurate results for a proper sizing of the circuit's components. Another issue relates to the balancing of the A_{VCG} and A_{VCS} which is not achieved properly in this optimization method.

- Wideband Optimization- This method considers a typical sizing of the components in the circuit. The implemented simulations in this method are based on the maximum and minimum obtained results from the executed simulations, which ensure a more properly adjusted results when compared with the previous method. By including the circuit's bandwidth in the fitness calculation of the individuals, containing the components sizing, it is visible that the transistors and loads generated size don't tend to go to higher dimensions, thus affecting the bandwidth. Despite

the better approach to the problem, the same issue regarding the balancing output operation is also present in this optimization method.

- N-factor Optimization- In this method the approach is based on the study provided from[1] where a N relationship between the size of the components is established to boost the gain and reduce the noise effect in the circuit. Having that in mind the generated dimensions of the transistors and loads are always generated based on a N sizing relationship. Considering this fixed relationship it is clear to see that the search space of the GA is drastically reduced resulting in a worse rating. Despite the worse fitness rating, the differential outputs are much more accurate in terms of the balancing operation.

- Wideband balancing Optimization- This optimization method is based on the previous implemented wideband method, with a different FOM for fitness calculation purpose, capable of comparing both differential gains. With this method it is considered the difference between A_{VCG} and A_{VCS}, leading to a more accurate balancing between both differential outputs.

- Wideband Optimization with Active Loads- This optimization is similar to the previous method, despite the adjusts required with the necessary changes regarding the active loads which replaced the previously used resistors. Considering the circuit changes applied in this method, the search space for the GA to search for an optimal sizing increases due to higher dynamic output resistance when considering the same DC voltage drop. This circuit changes lead to a higher rating when compared with the previous optimization.

- Wideband Optimization with Self-Polarization- This method is based on the previous method, considering the changed resistors for active loads. The changes to the previous method are related to the used polarization in the optimization process. In this method the voltage and current sources are also added into the optimization process, giving the GA a more freedom, thus increasing the search space. Considering this changes, the obtained results in this simulation are the highest rated and when compared with the state-of-the-art presented works it is shown that the optimization platform was able to provide a well balanced sizing for the circuit, also providing a good overall performance.

The main issue related to the implemented optimization methods relates to the executed number of generations. Considering that the fitness calculation uses the provided results from the simulations given by the Cadence Spectre, the spent amount of time from each simulation makes the developed software a not so viable solution if the spent time is a criteria with a relevant weight. Therefore, the shown results may not be the optimal sizing due to the low number of iterations.

Another issue relates to the initial population generated for each population, where a trade-off between a reasonable number of individuals and time needs to be considered. Since the increase of the initial population may provide better adjusted individuals, and also guaranteeing that the GA does not end up in a local solution, the number of individuals in the initial population should be as high as possible, but on the other hand the spent time in executing more simulations in not relevant individuals isn't a good compromise. Considering this issue the used solution was to add an individual with a known sizing, where $M_1 = M_2$ and $R_1 = R_2$ or $M_3 = M_4$.

Taking in consideration each described optimization methods and their issues, it is possible to conclude that each presented method increases the sizing rate considerably, being the wideband optimization with self-polarization the one with the highest rated individual. With all the developed methods, an eminent conclusion consists in the fact that $N_W \neq 1$ and $N_R \neq 1$ showing that the different sizing relationship, between the components, increases the circuits' performance.

Considering the simulation results it is also possible to conclude that with different relationship between the transistors and loads, the circuit's gain is improved, as well as the noise figure. Due to consumption and matching requirements, the difference between M_1 and M_2 size is not that big, which limits the circuit's gain but increases the bandwidth. The difference between the loads also show that the GA tends to increase both sizes, but with a relationship which considers the balancing output conditions.

In conclusion, the implemented optimization platform, is a viable tool for the presented circuits, which stands in conformity with the other state-of-art related works and could easily be adapted to other LNA architectures.

7.2 Future Work

For future work purpose the following topics are suggested:

- Use the Cadence Cloud service in the optimization process, which would be able to improve the simulations' spent time. Also the idea of multiple simulations running at once may be implemented.
- The circuits' components and layout are described in files, as well as the simulations results, which could be replaced for a more efficient method.
- Add the linearity study to the fitness evaluation process.
- Develop a more efficient way of providing the transistors and loads polarization, to guarantee a higher rated fitness value.
- Integrate the platform with an online database to be able to save all the generations information.

- Add additional operators to the GA capable of reducing amount of non relevant individuals.
- Search for other LNA topologies, and apply smaller CMOS technologies, that can lead to better performance.

Bibliography

[1] S. C. Blaakmeer, E. A. M. Klumperink, S. Member, D. M. W. Leenaerts, and B. Nauta. "Wideband Balun-LNA With Simultaneous Output." In: *IEEE Journal of Solid-State Circuits* 43.6 (2008), pp. 1341–1350. ISSN: 00189200. DOI: 10.1109/JSSC.2008.922736.

[2] H.-h. Chen, M.-h. Chen, and C.-y. Tsai. "Optimization of Low Noise Amplifier Designs by Genetic Algorithms." In: *2013 International Symposium on Electromagnetic Theory* (2013), pp. 493–496. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp={\&}arnumber=6565786.

[3] D. A. Coley. *An introduction to genetic algorithms for scientists and engineers*. World Scientific, 1999, p. 227. ISBN: 9810236026. URL: https://b-ok.cc/book/592351/2078d6.

[4] M. De Souza, A. Mariano, and T. Taris. "Reconfigurable Inductorless Wideband CMOS LNA for Wireless Communications." In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.3 (2017), pp. 675–685. DOI: 10.1109/TCSI.2016.2618361. URL: http://ieeexplore.ieee.org/document/7795252/.

[] *DEAP documentation — DEAP 1.2.2 documentation*. URL: https://deap.readthedocs.io/en/master/ (visited on 02/07/2019).

[5] S. D. E. Decis. "Introdução às Técnicas de Computação Evolutiva." In: (2016).

[6] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing (Google eBook)*. 2015, p. 299. ISBN: 3540401849. DOI: 10.1007/BF01168946. arXiv: 9809069v1 [arXiv:gr-qc]. URL: http://books.google.com/books?id=RRKo9xVFW{_}QC{\&}pgis=1.

[7] M. B. Fallahpour, K. Delfan Hemmati, and A. Pourmohammad. "Optimization of a LNA Using Genetic Algorithm." In: *Electrical and Electronic Engineering* 2012.2 (2012), pp. 38–42. DOI: 10.5923/j.eee.20120202.07. URL: http://journal.sapub.org/eee.

[8] M. D. Fernandes and L. B. O. Editors. *Wideband CMOS Receivers*. 2015, p. 98. ISBN: 3319189204. DOI: 10.1007/978-3-319-18920-8. URL: https://books.google.com/books?id=DWknCgAAQBAJ{\&}pgis=1.

[9] K. D. Hemmati, M. Hojjati, M. Behzad, and A. Golmakani. "Design of a Low Noise Amplifire (LNA) with Image Rejection Filter." In: 2.6 (2012), pp. 351–354. ISSN: 2162-9455. DOI: 10.5923/j.eee.20120206.02.

[10] I. Iuri and A. Bastos. *A MOSFET-Only Wideband LNA Exploiting Thermal Noise Canceling and Gain Optimization*. Tech. rep. 2010. URL: https://run.unl.pt/bitstream/10362/4867/1/Bastos{_}2010.pdf.

[] S. Kang, B. Choi, and B. Kim. "Linearity Analysis of CMOS for RF Application." In: 4 (2002), pp. 363–366.

[11] Y. Li. "A simulation-based evolutionary approach to LNA circuit design optimization." In: *Applied Mathematics and Computation* 209.1 (2009), pp. 57–67. ISSN: 00963003. DOI: 10.1016/j.amc.2008.06.015. URL: http://dx.doi.org/10.1016/j.amc.2008.06.015.

[] M Manghisoni, L Ratti, V Re, V Speziali, S. Member, and G Traversi. "Noise Characterization of 130 nm and 90 nm CMOS Technologies for Analog Front-end Electronics." In: (2006), pp. 214–218. DOI: 10.1109/NSSMIC.2006.356142.

[] S. Manjula and D. Selvathi. "Design and optimization of ultra low power low noise amplifier using particle swarm optimization." In: 8.36 (2015). ISSN: 0974-5645. DOI: 10.17485/ijst/2015/v8i36/54954.

[12] D. Pozar. *Microwave Engineering 2nd Ed David Pozar*. Tech. rep. URL: https://hasanahputri.staff.telkomuniversity.ac.id/files/2016/09/Microwave-Engineering-2nd-Ed-David-Pozar-OCRed.pdf.

[13] *RF Circuit Design,Theory and Application.pdf*. DOI: 10.1186/1479-5868-8-127.

[14] *Simulation of a common-source circuit using SOCAD and ADE-L — SOCAD 0.1.0 documentation*. URL: https://socad.readthedocs.io/en/latest/tutorials/common{_}source.html (visited on 02/07/2019).

[] B. Toole, C. Plett, and M. Cloutier. "RF Circuit Implications of Moderate Inversion Enhanced Linear Region in MOSFETs." In: 51.2 (2004), pp. 319–328.

[15] S. Ülker. "DESIGN OF LOW NOISE MICROWAVE AMPLIFIERS USING PARTICLE SWARM OPTIMIZATION." In: *International Journal of Artificial Intelligence & Applications (IJAIA)* 3.4 (2012). DOI: 10.5121/ijaia.2012.3407. URL: https://arxiv.org/ftp/arxiv/papers/1208/1208.6028.pdf.

[16] P. Vancorenland, C. De Ranter, M. Steyaert, and G. Gielen. "Optimal RF design using smart evolutionary algorithms." In: *Proceedings of the 37th conference on Design automation - DAC '00* June (2000), pp. 7–10. ISSN: 0738100X. DOI: 10.1145/337292.337299. URL: http://portal.acm.org/citation.cfm?doid=337292.337299.

[17] A. Zalzala and P. Fleming. "Genetic algorithms in engineering systems." In: *IEEE Antennas and Propagation Magazine* 39.4 (1997), pp. 7–21. DOI: 10.1049/PBCE055E.

ANNEX I

Analysis Results from Cadence

I.1 N-Points Optimization

In this section are presented the analysis results from the best individual provided by the N-Points optimization method.

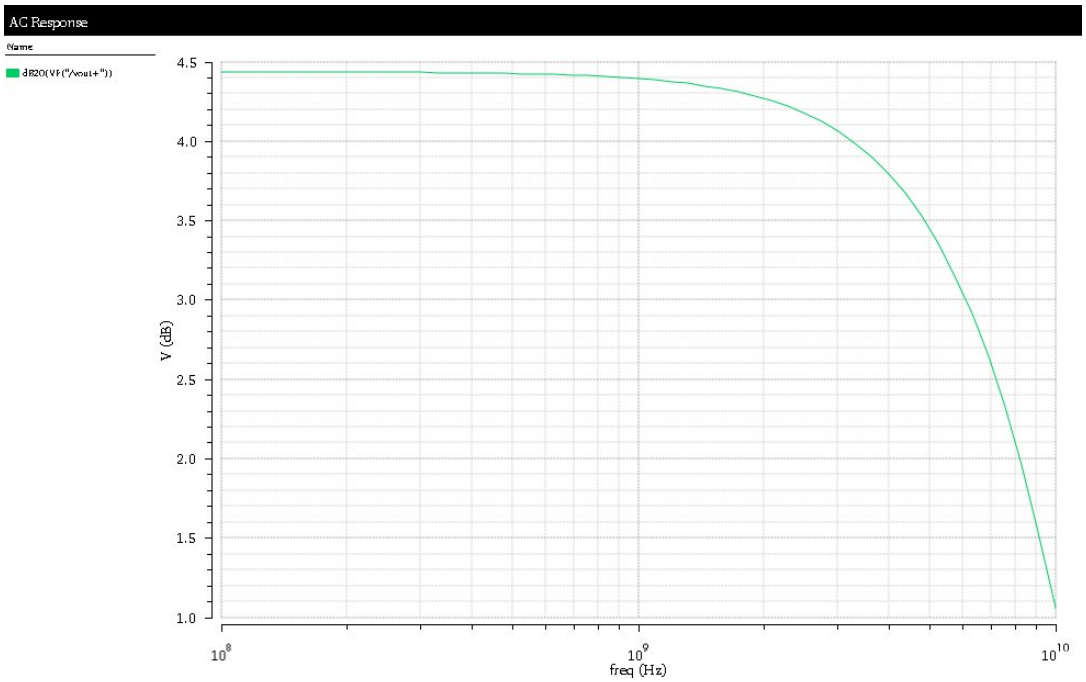

Figure I.1: A_{VCG} analysis in N-Points Optimization.

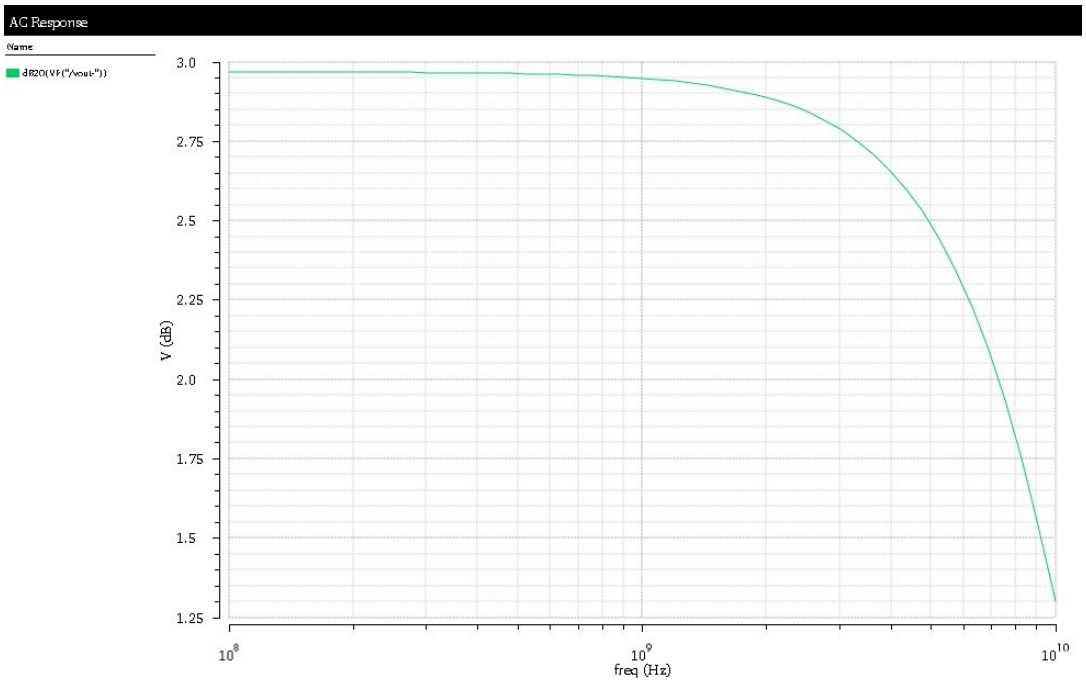

Figure I.2: A_{VCS} analysis in N-Points Optimization.

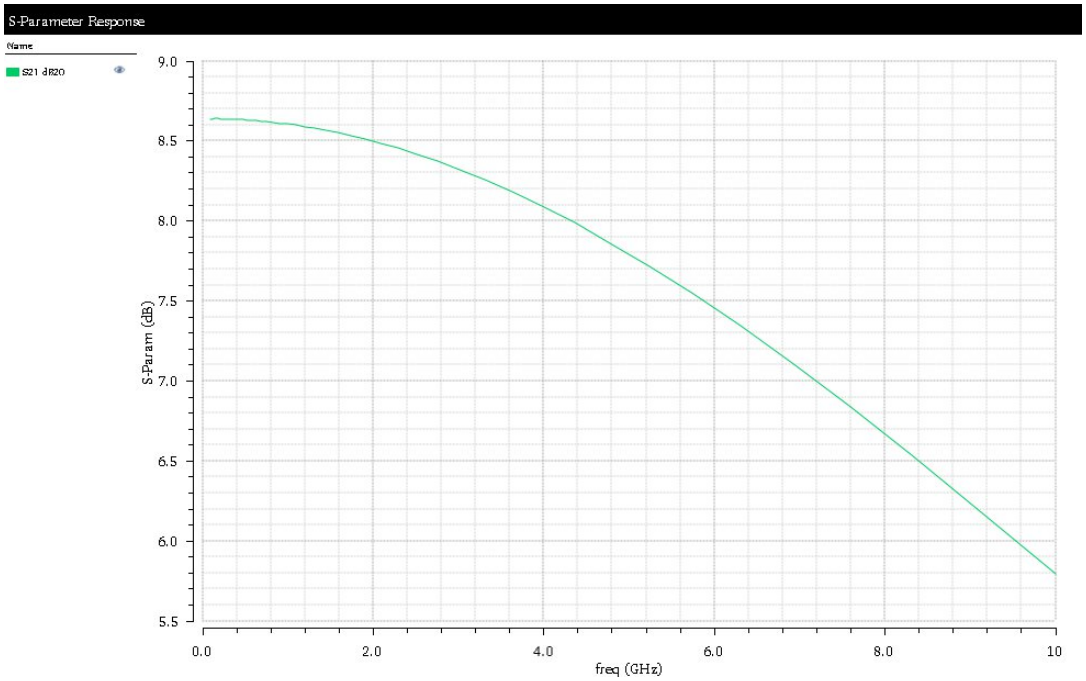

Figure I.3: S_{21} analysis in N-Points Optimization.

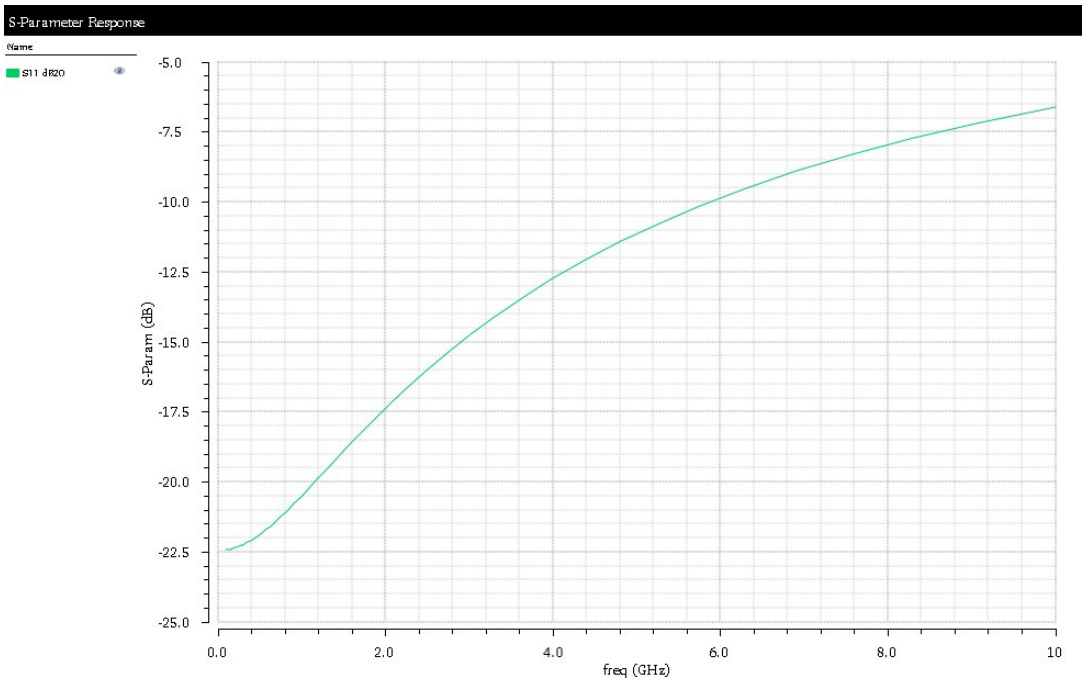

Figure I.4: S_{11} analysis in N-Points Optimization.

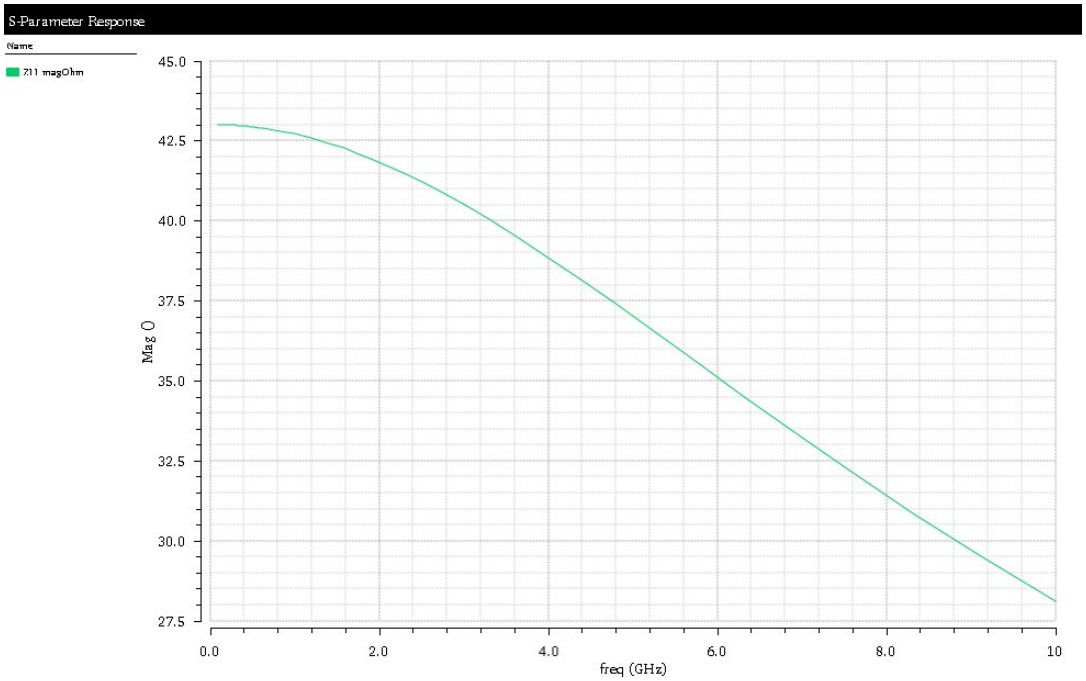

Figure I.5: Z_{11} analysis in N-Points Optimization.

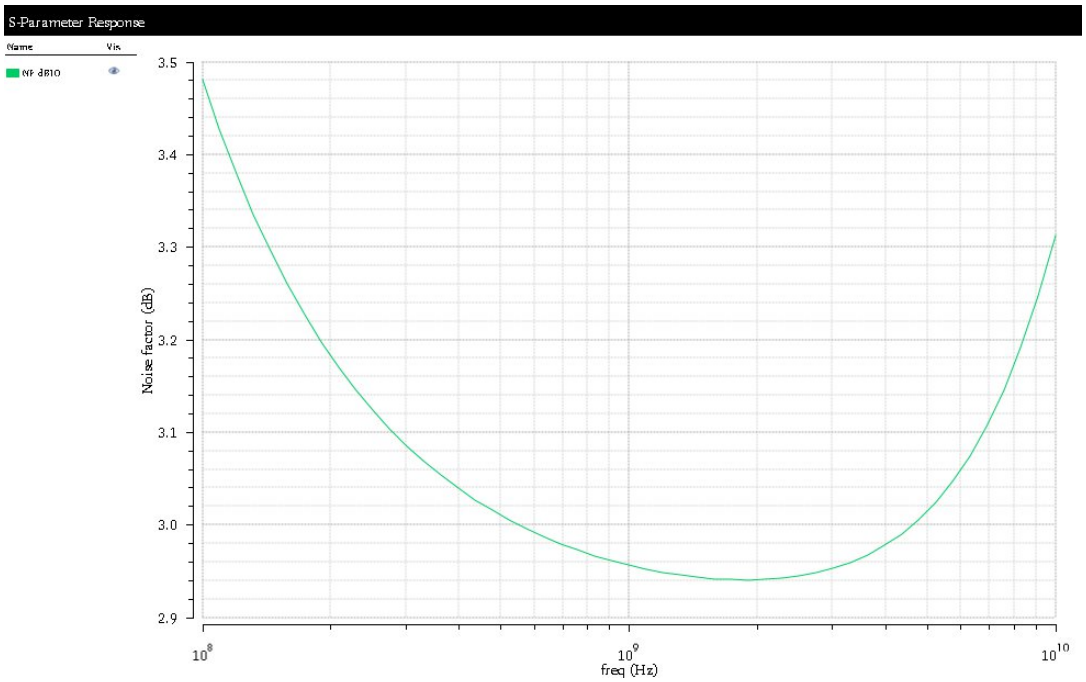

Figure I.6: NF analysis in N-Points Optimization.

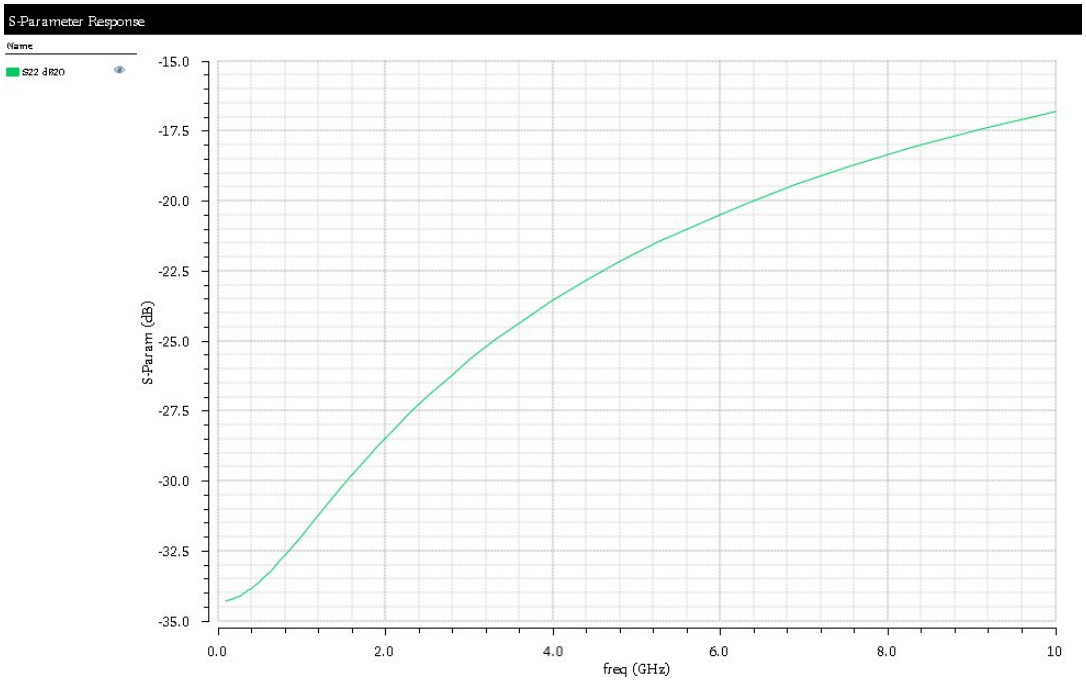

Figure I.7: S_{22} analysis in N-Points Optimization.

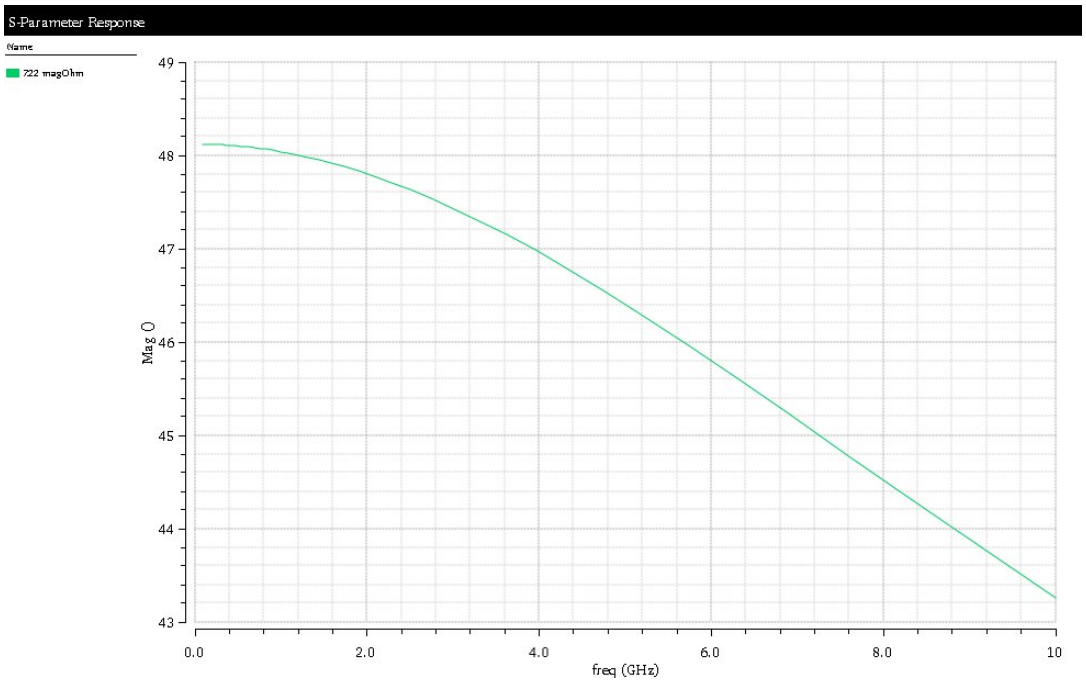

Figure I.8: Z_{22} analysis in N-Points Optimization.

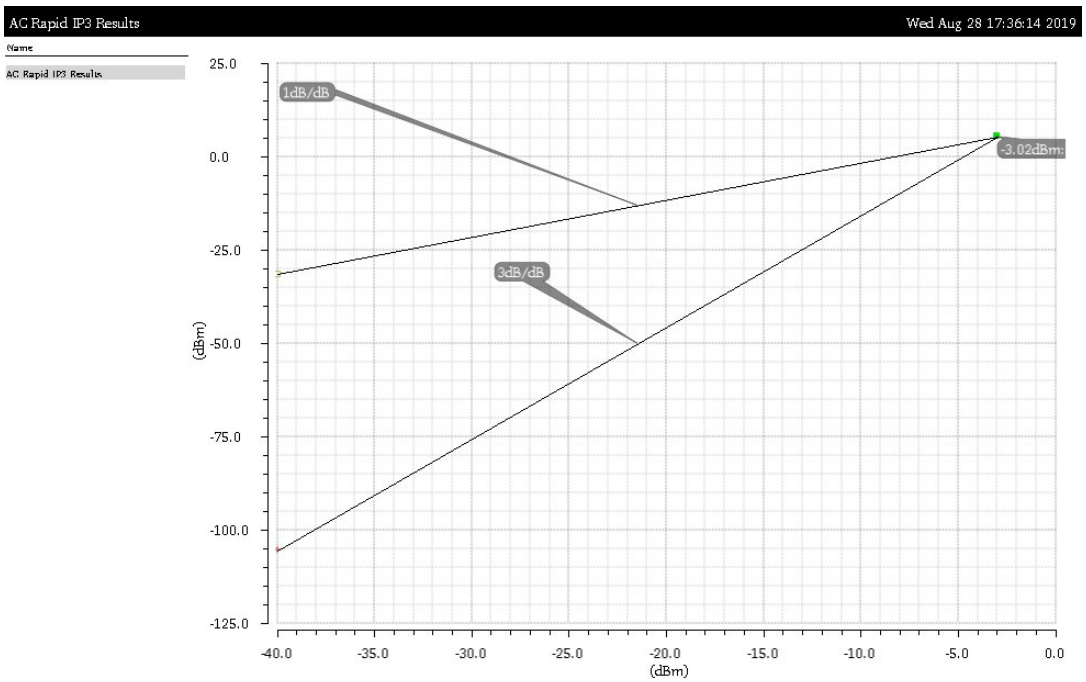

Figure I.9: $IP3$ analysis in N-Points Optimization.

I.2 Wideband Optimization

In this section are presented the analysis results from the best individual provided by the wideband optimization method.

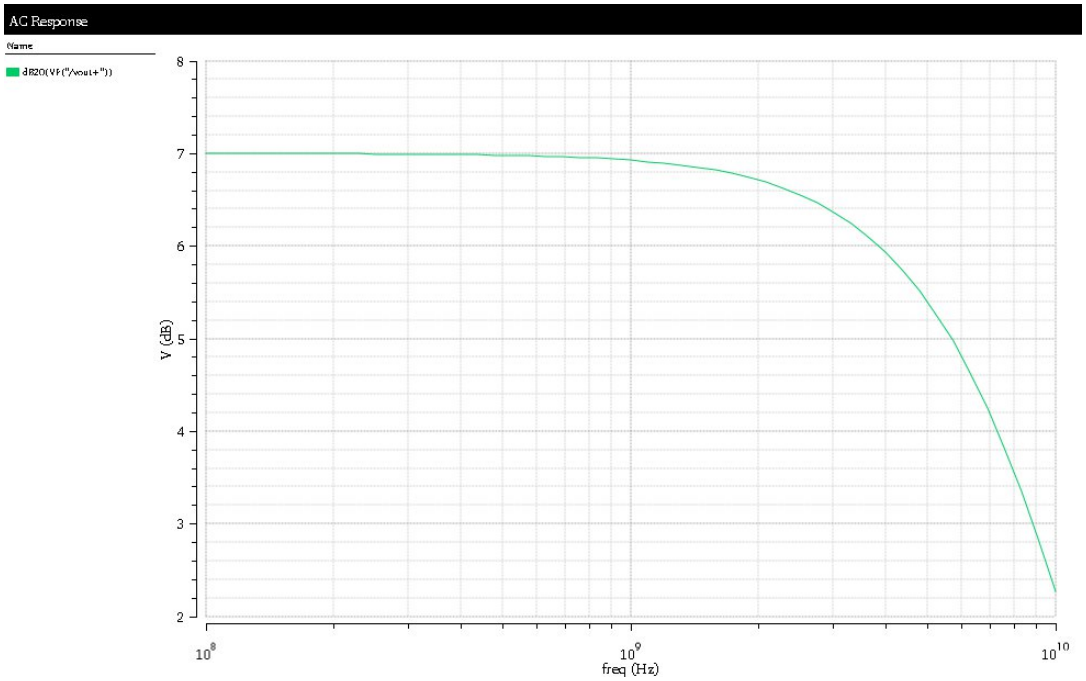

Figure I.10: A_{VCG} analysis in Wideband Optimization.

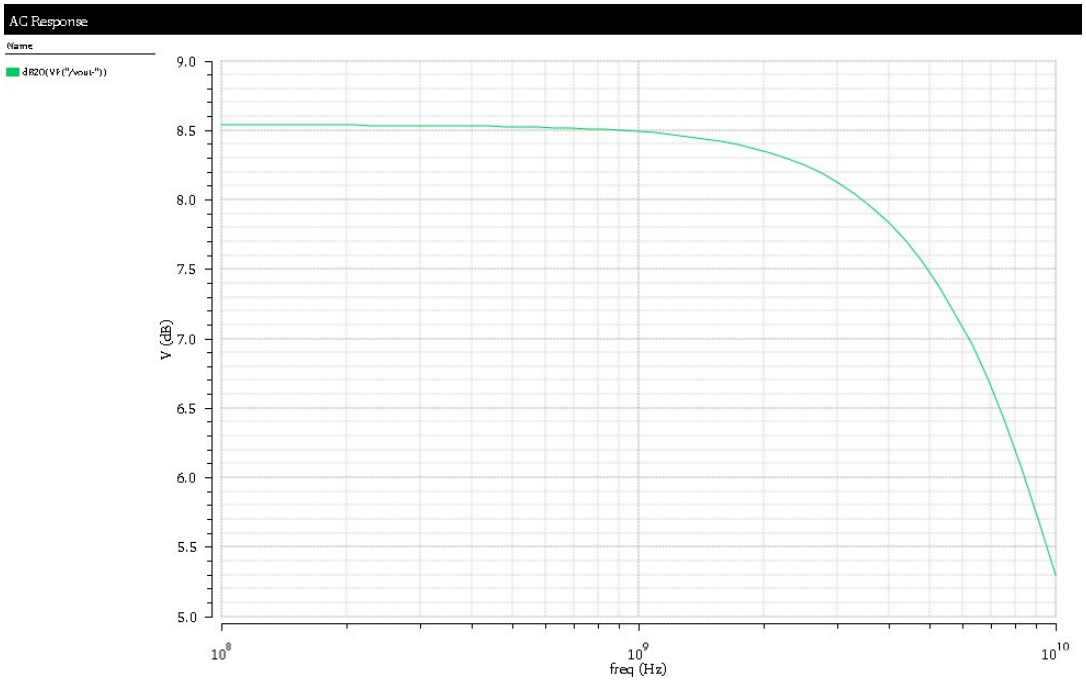

Figure I.11: A_{VCS} analysis in Wideband Optimization.

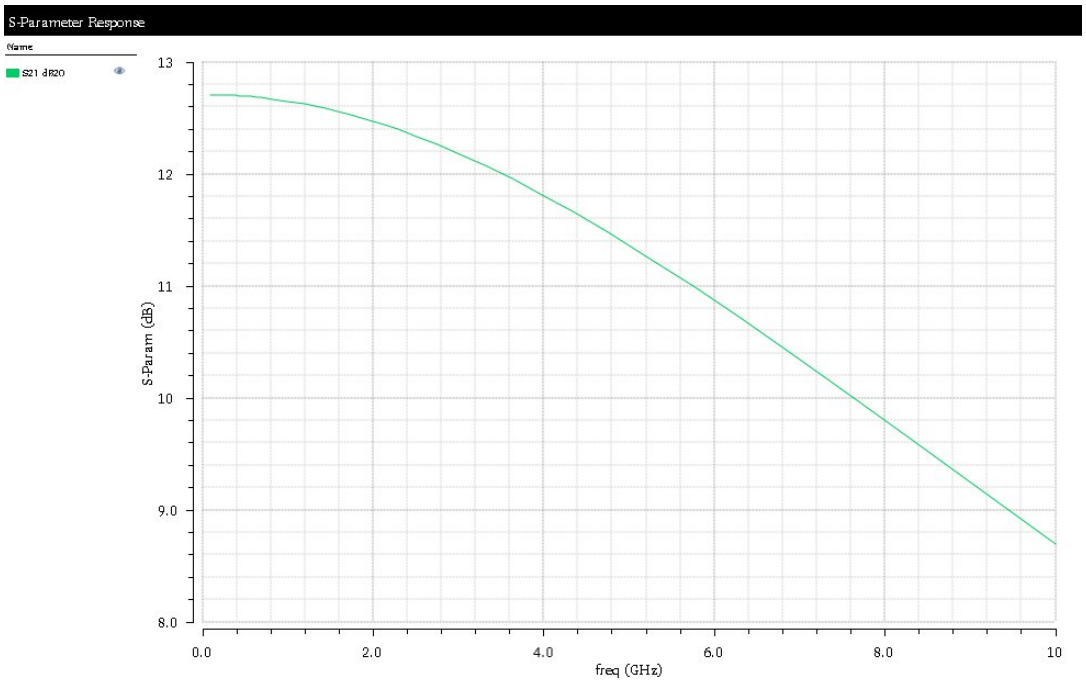

Figure I.12: S_{21} analysis in Wideband Optimization.

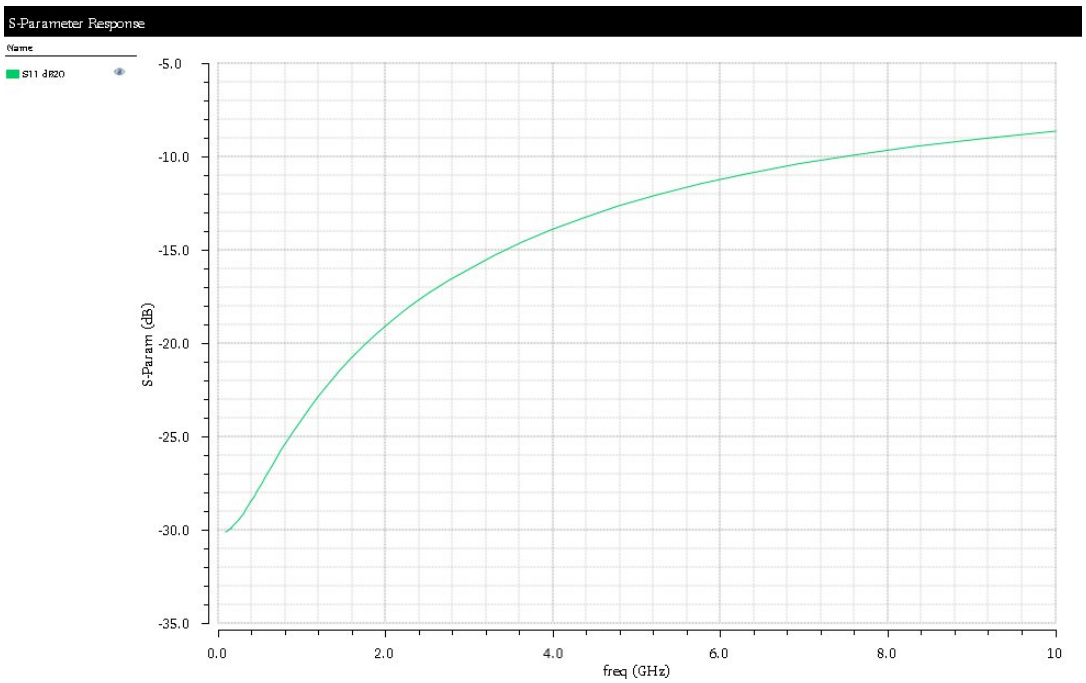

Figure I.13: S_{11} analysis in Wideband Optimization.

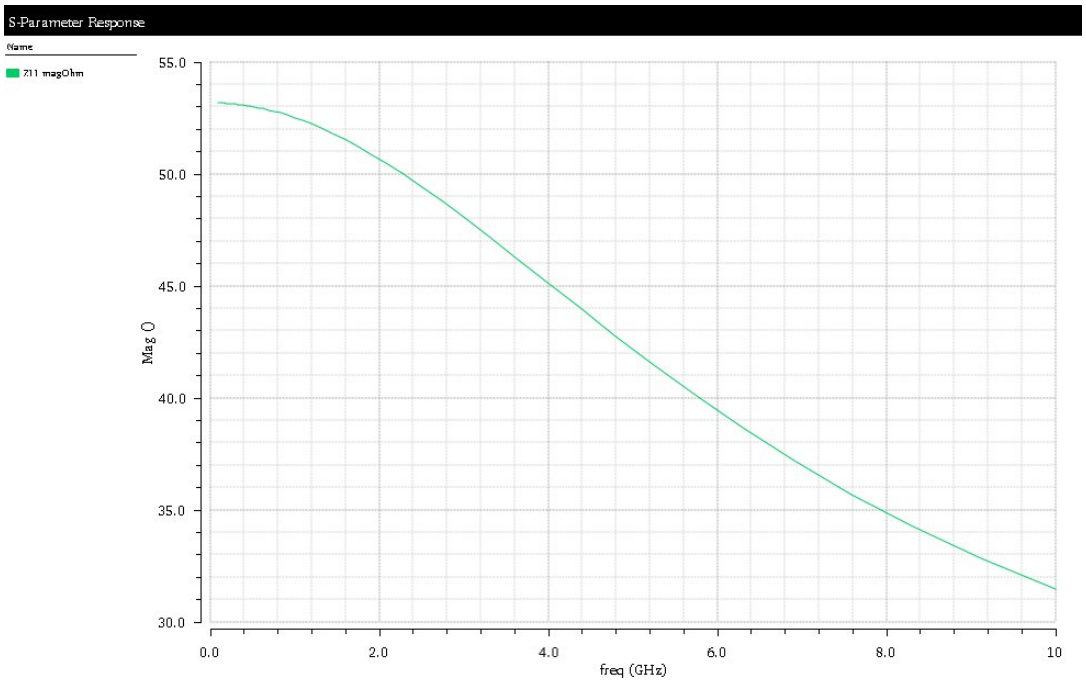

Figure I.14: Z_{11} analysis in Wideband Optimization.

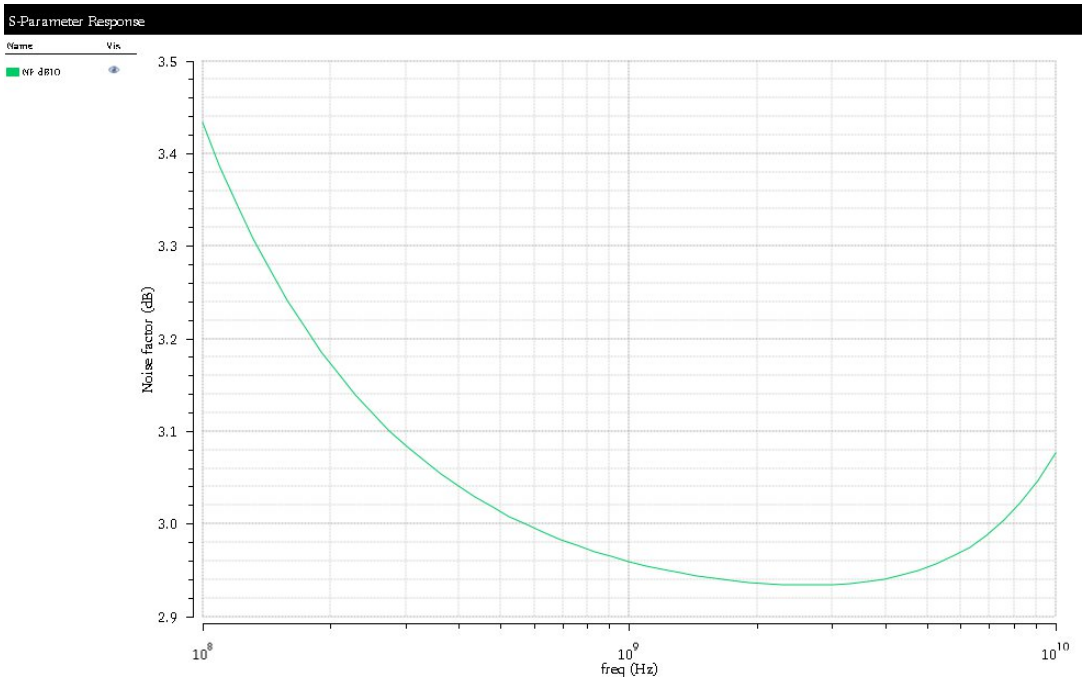

Figure I.15: NF analysis in Wideband Optimization.

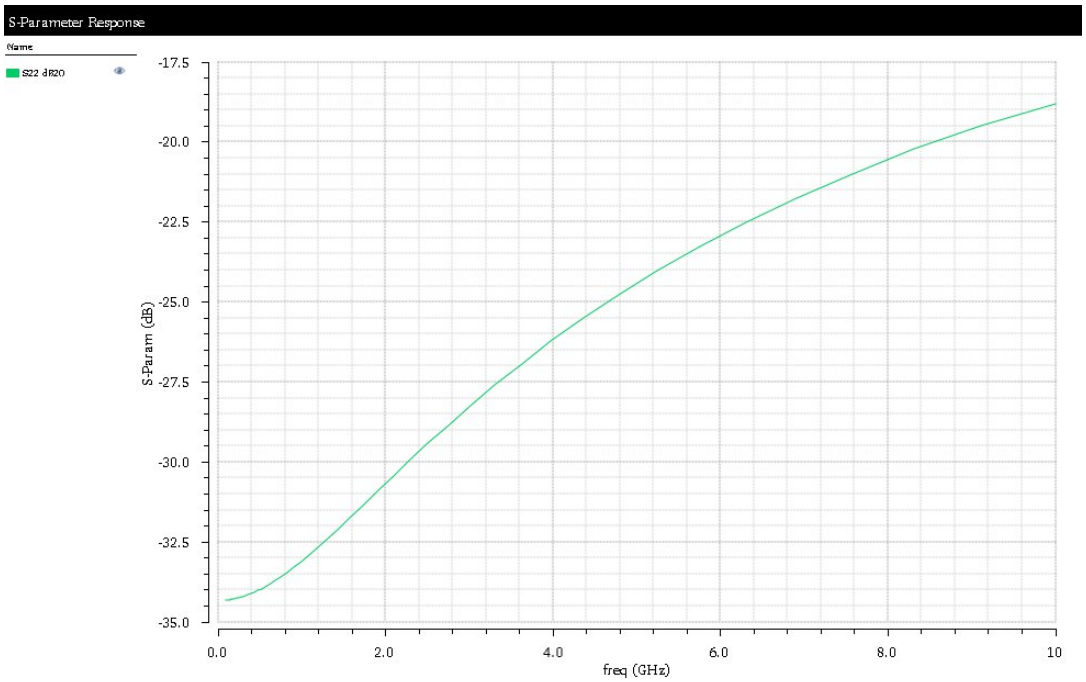

Figure I.16: S_{22} analysis in Wideband Optimization.

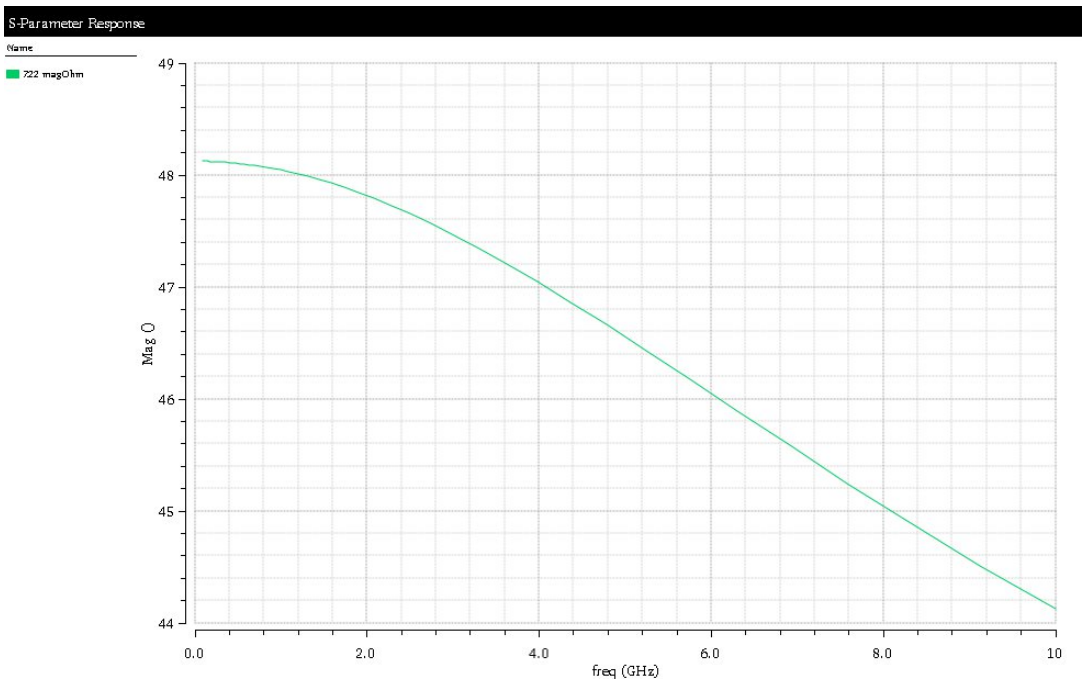

Figure I.17: Z_{22} analysis in Wideband Optimization.

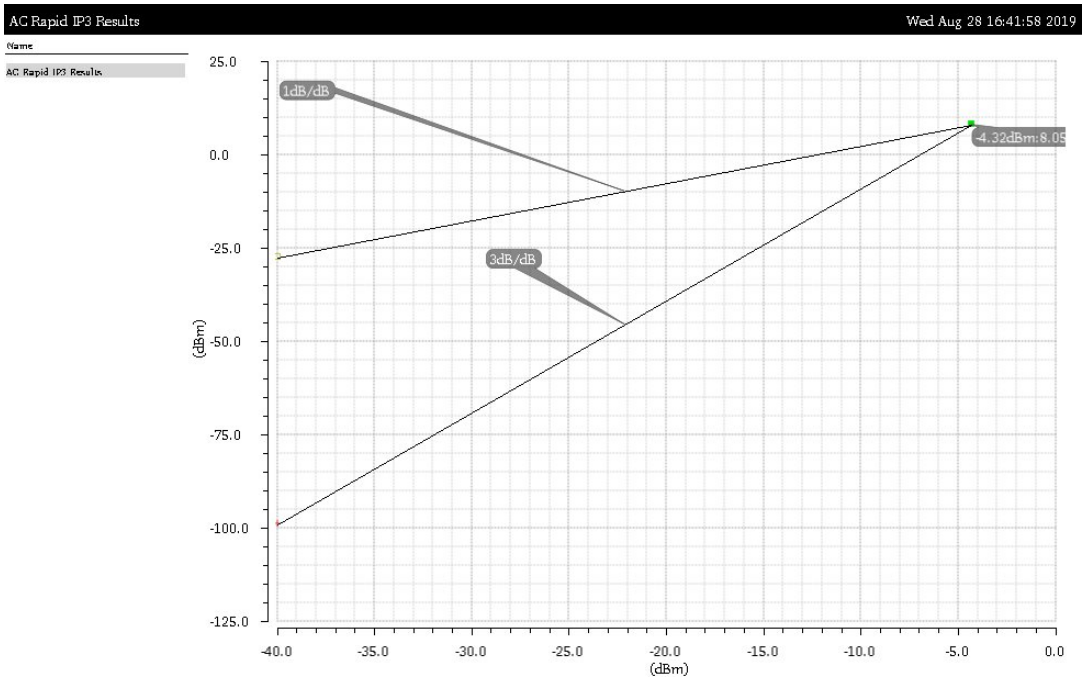

Figure I.18: IP3 analysis in Wideband Optimization.

I.3 Wideband N-factor Optimization

In this section are presented the analysis results from the best individual provided by the N-factor optimization method.

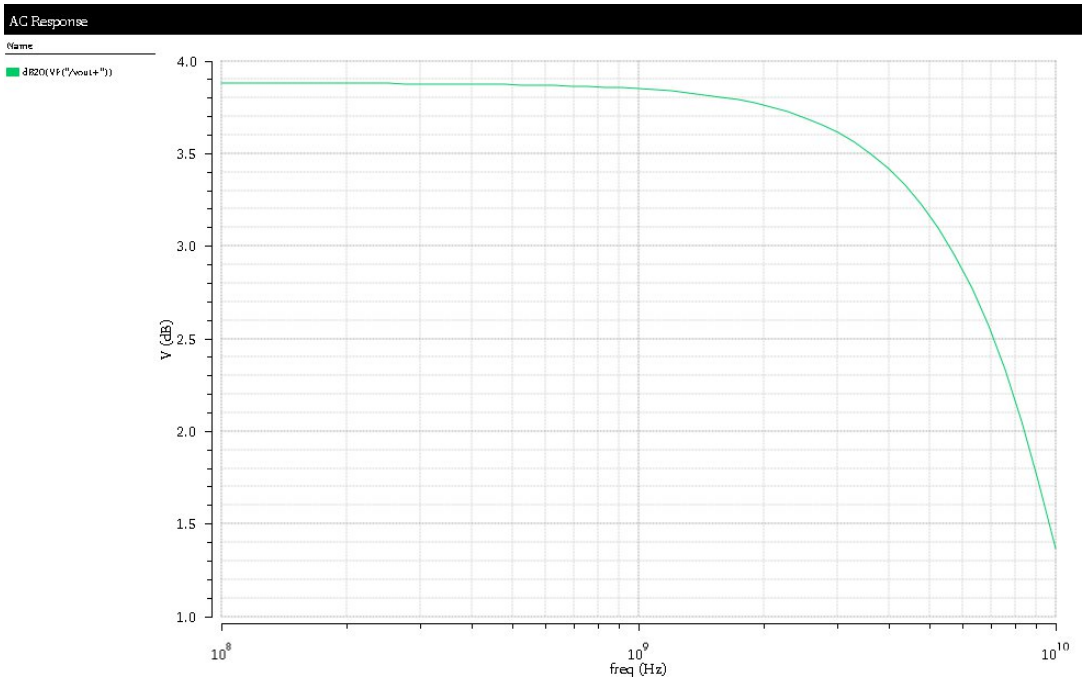

Figure I.19: A_{VCG} analysis in Wideband N-Factor Optimization.

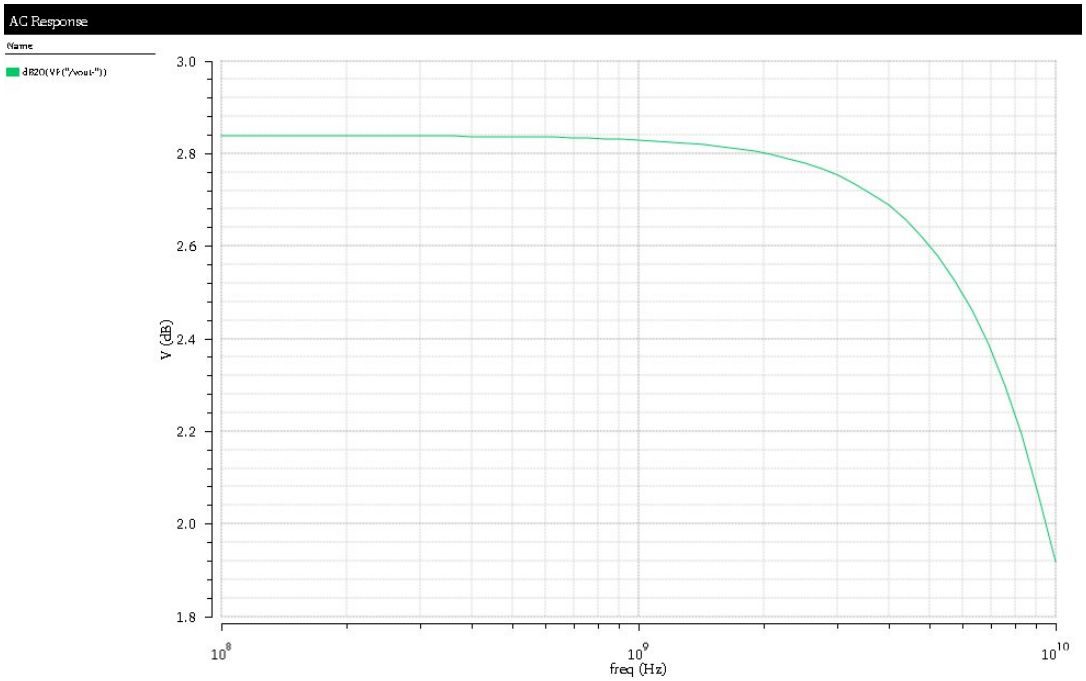

Figure I.20: A_{VCS} analysis in Wideband N-Factor Optimization.

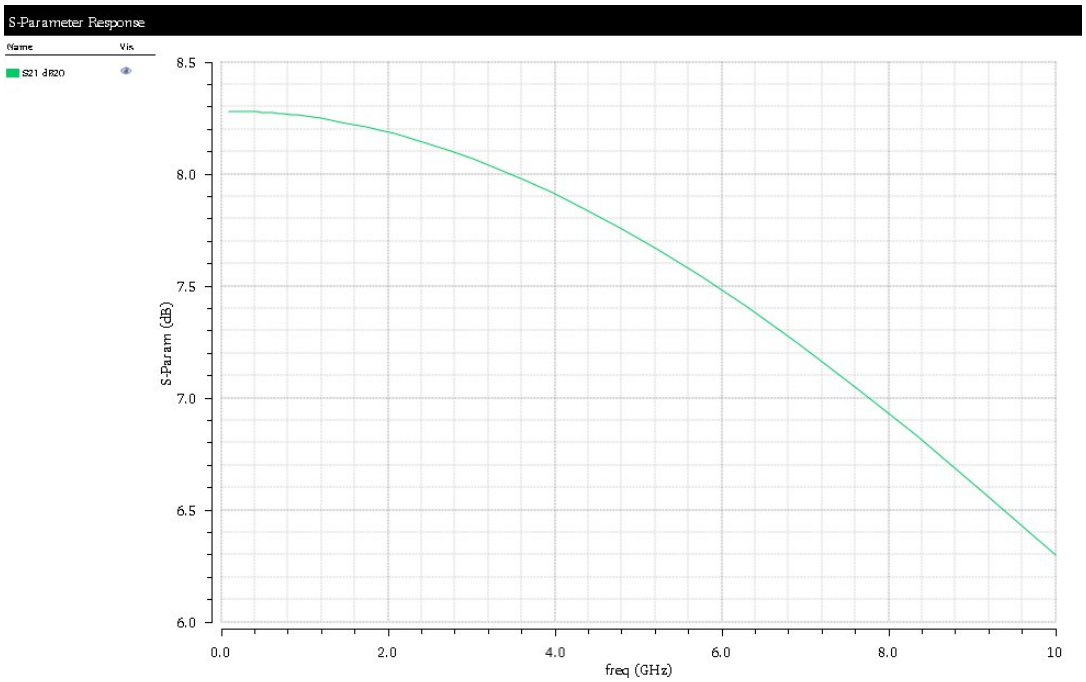

Figure I.21: S_{21} analysis in Wideband N-Factor Optimization.

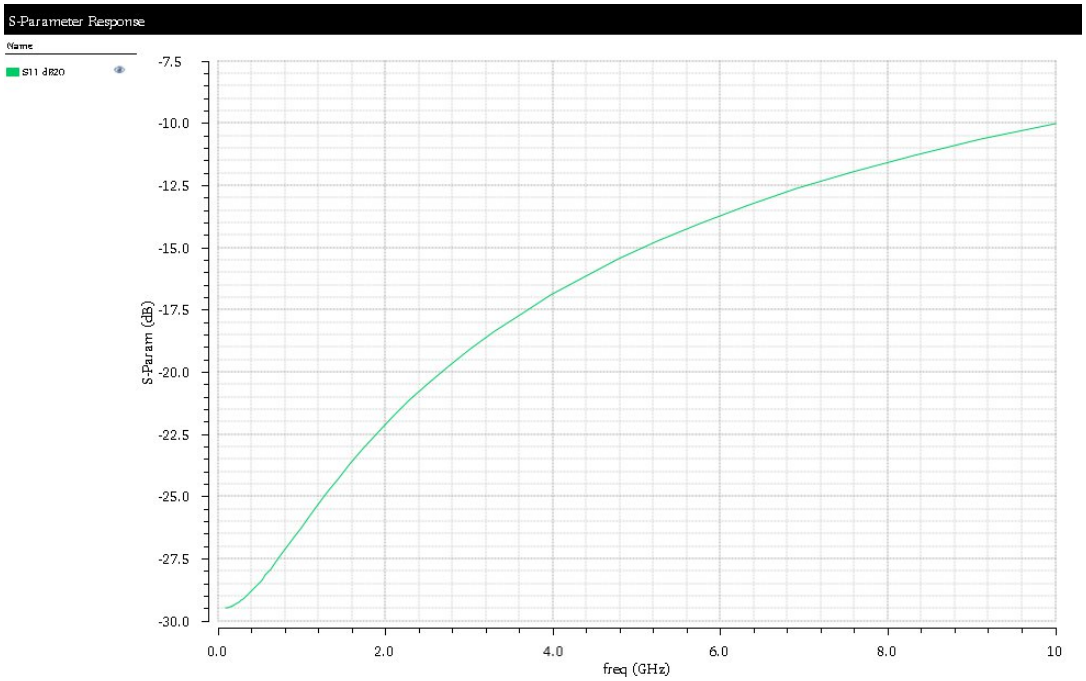

Figure I.22: S_{11} analysis in Wideband N-Factor Optimization.

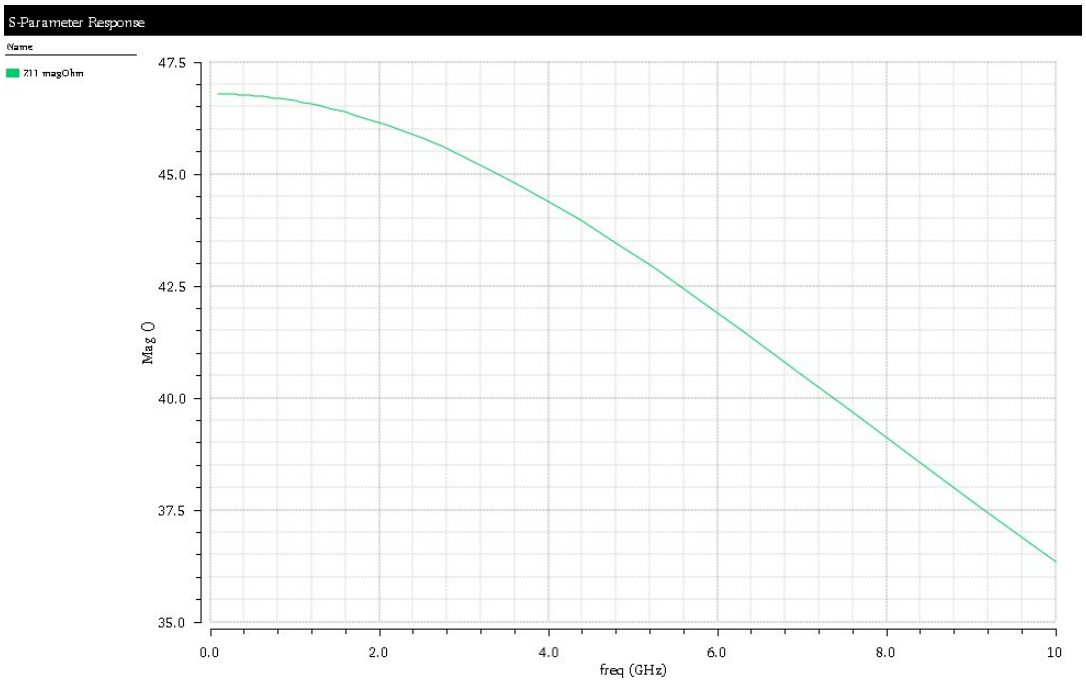

Figure I.23: Z_{11} analysis in Wideband N-Factor Optimization.

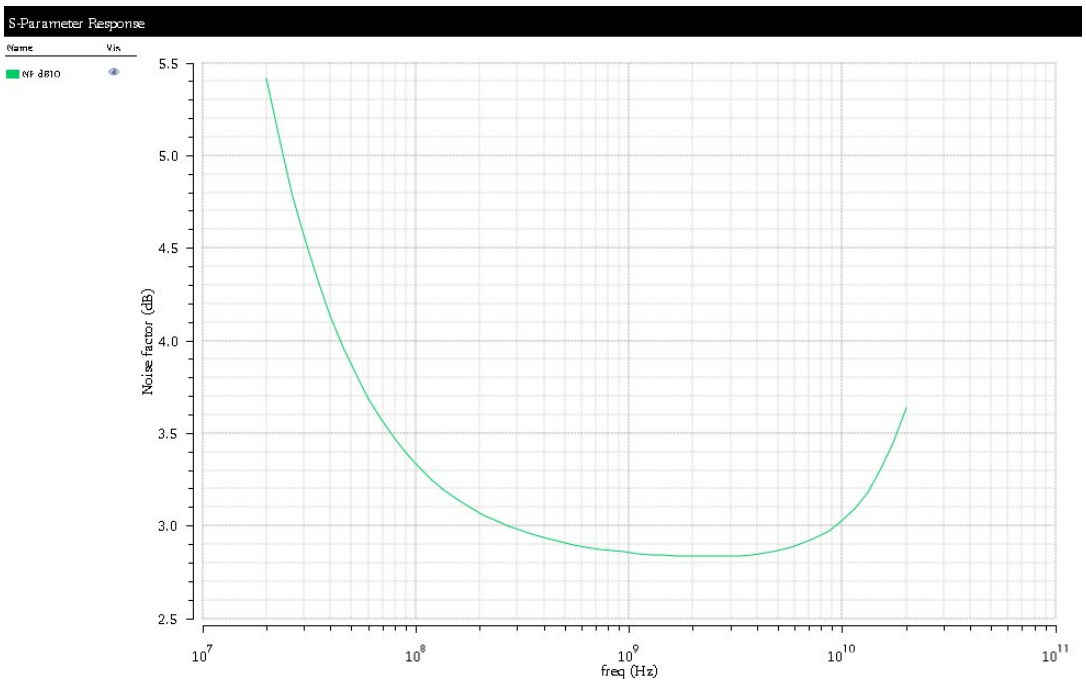

Figure I.24: NF analysis in Wideband N-Factor Optimization.

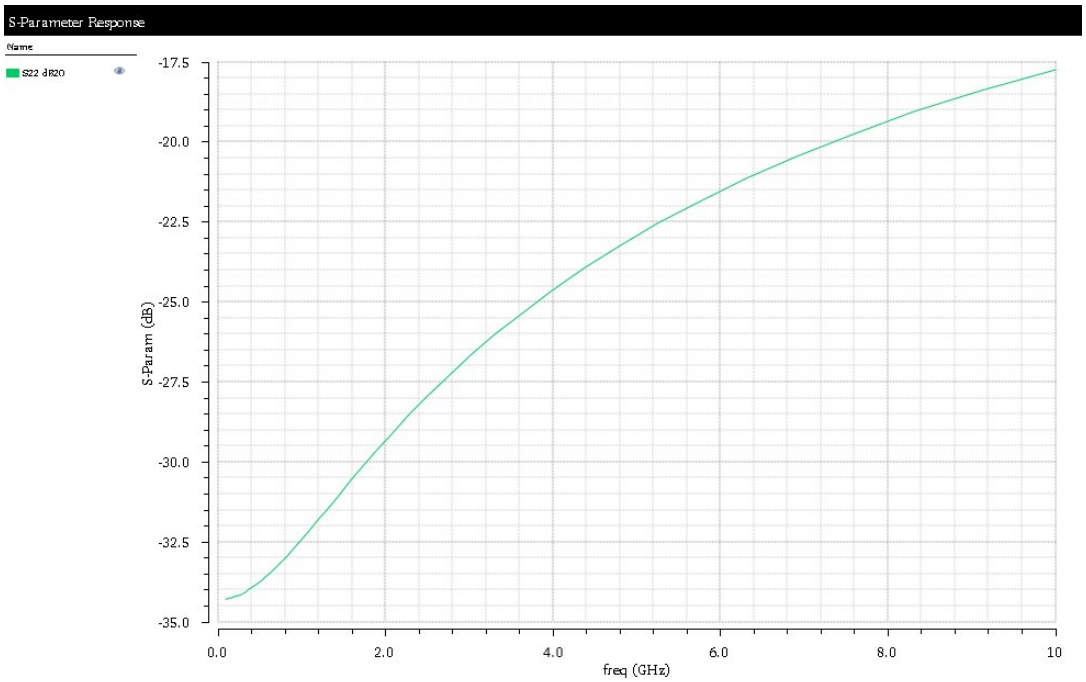

Figure I.25: S_{22} analysis in Wideband N-Factor Optimization.

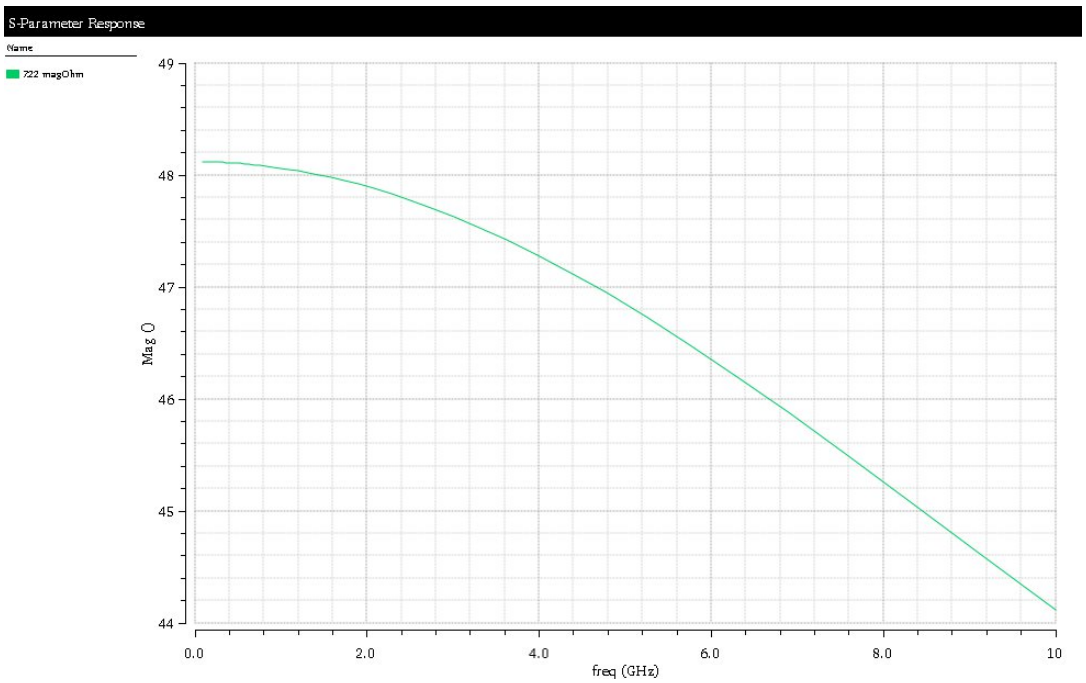

Figure I.26: Z_{22} analysis in Wideband N-Factor Optimization.

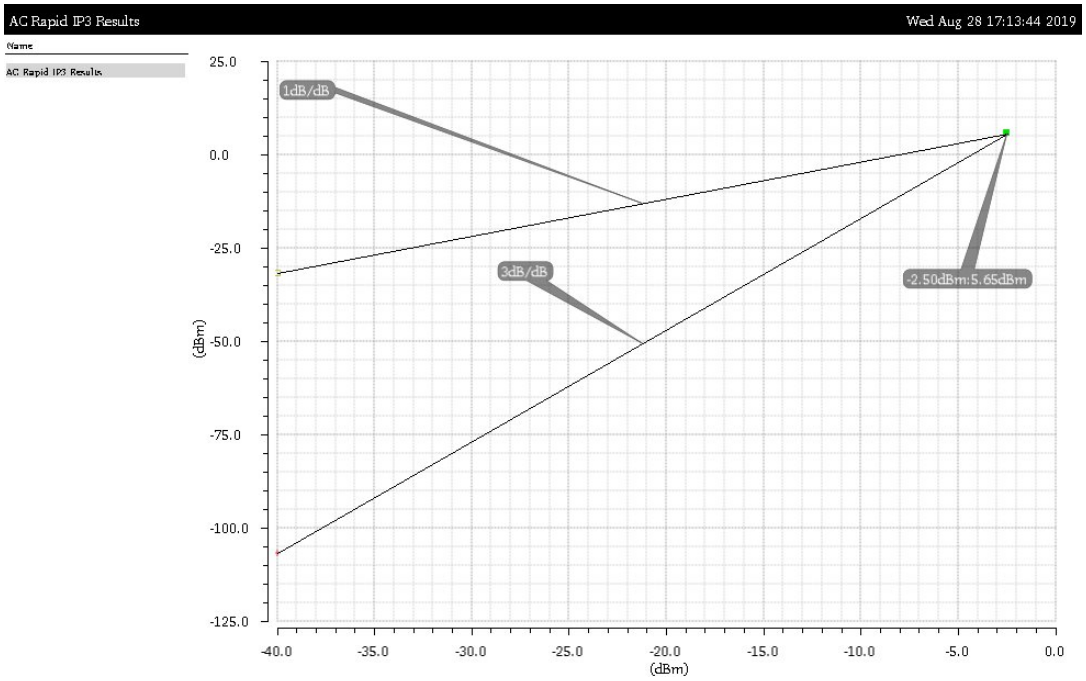

Figure I.27: IP3 analysis in Wideband N-Factor Optimization.

I.4 Wideband Balancing Optimization

In this section are presented the analysis results from the best individual provided by the wideband balancing optimization method.

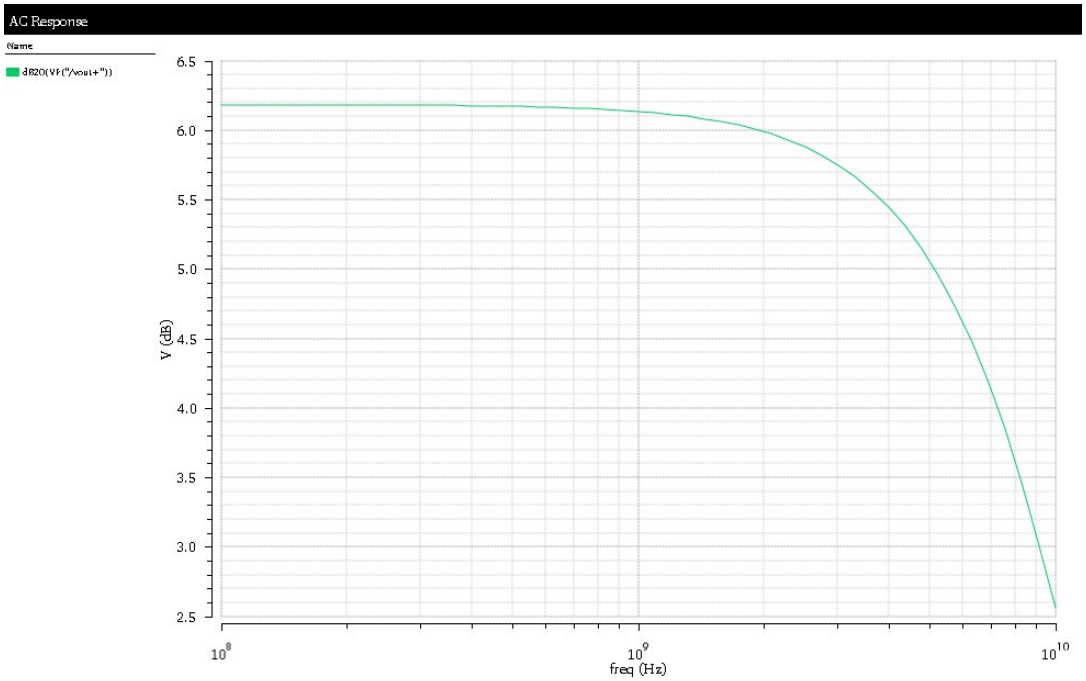

Figure I.28: A_{VCG} analysis in Wideband Balancing Optimization.

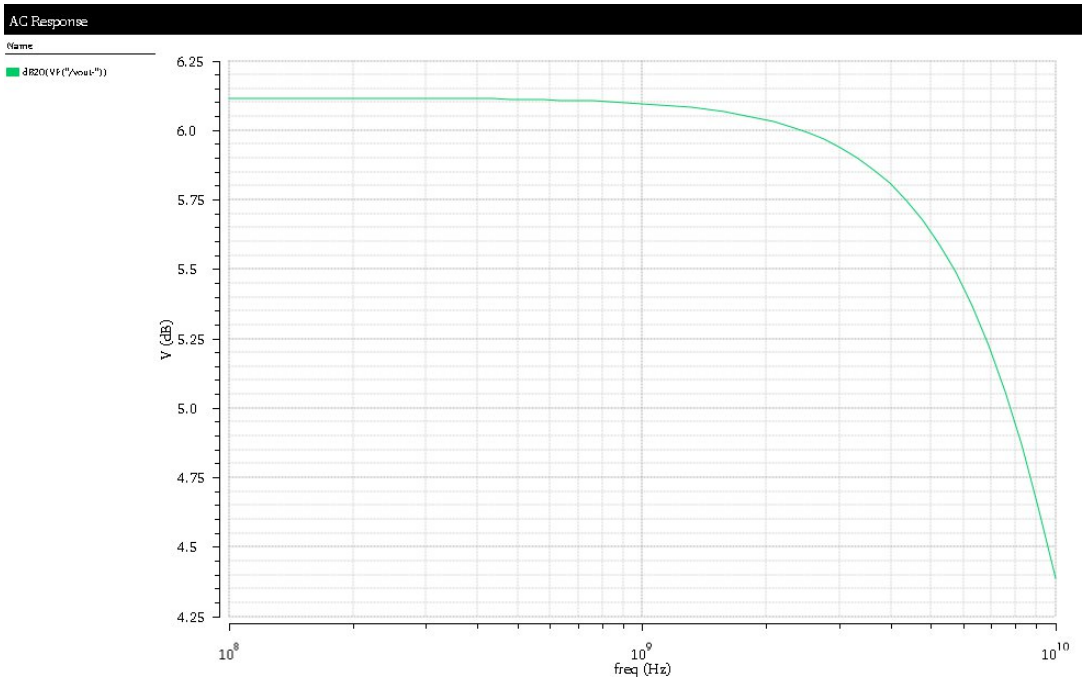

Figure I.29: A_{VCS} analysis in Wideband Balancing Optimization.

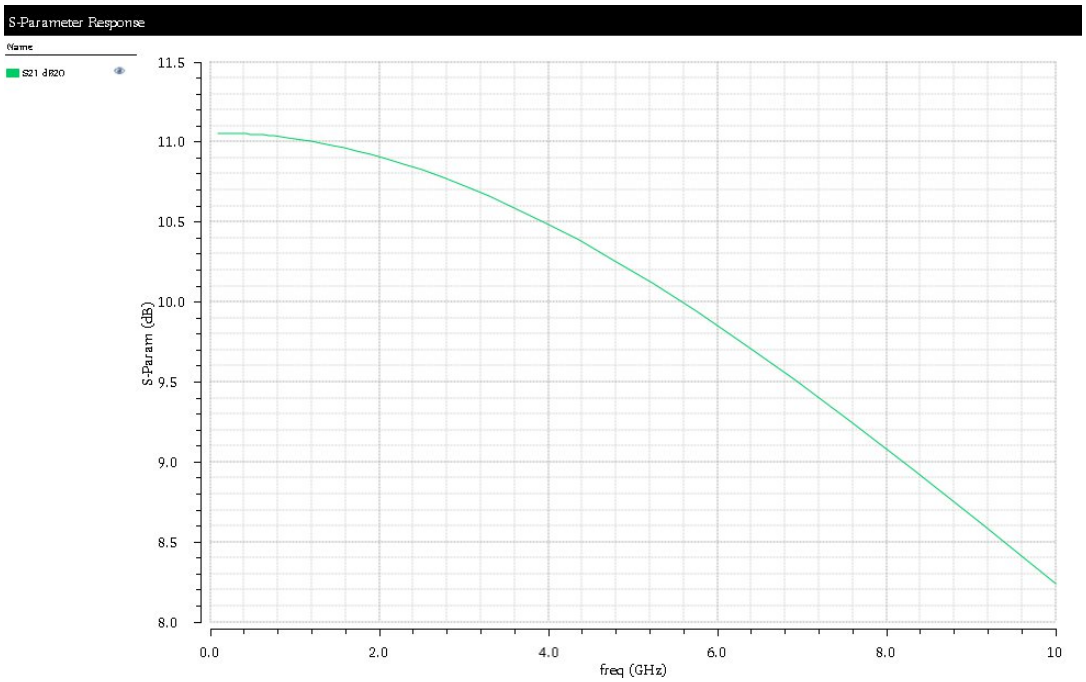

Figure I.30: S_{21} analysis in Wideband Balancing Optimization.

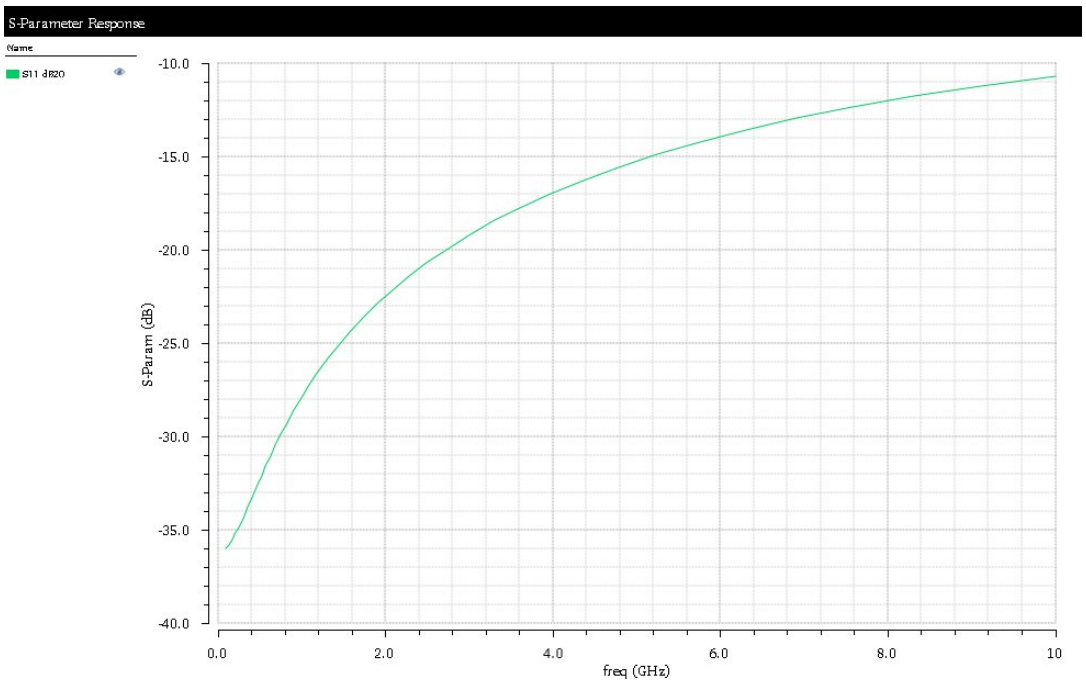

Figure I.31: S_{11} analysis in Wideband Balancing Optimization.

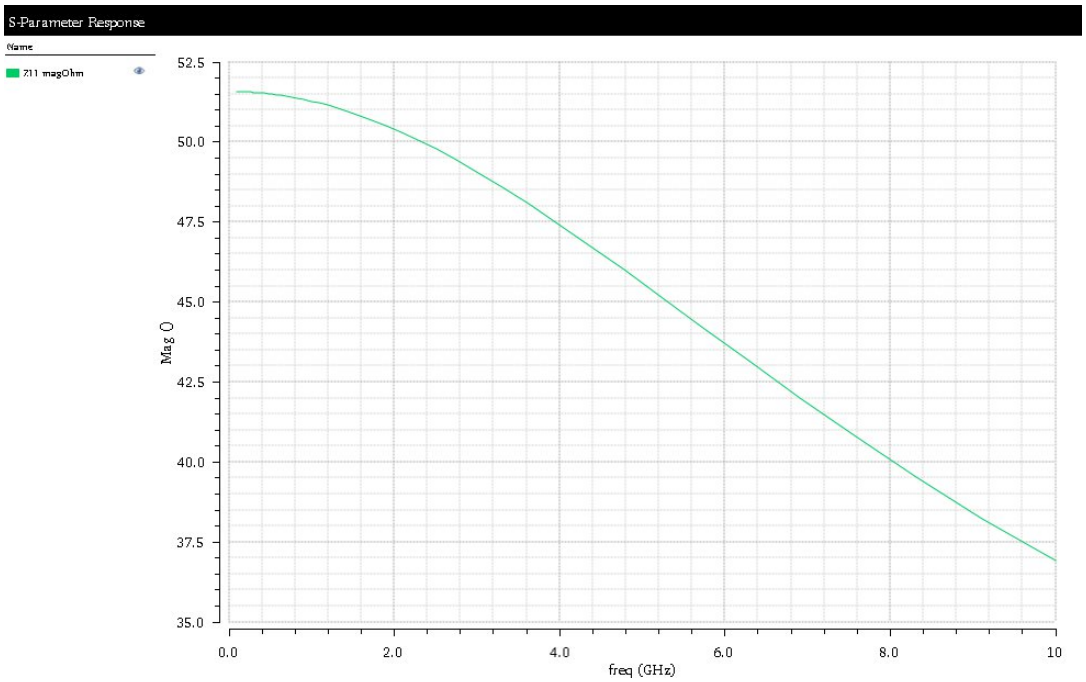

Figure I.32: Z_{11} analysis in Wideband Balancing Optimization.

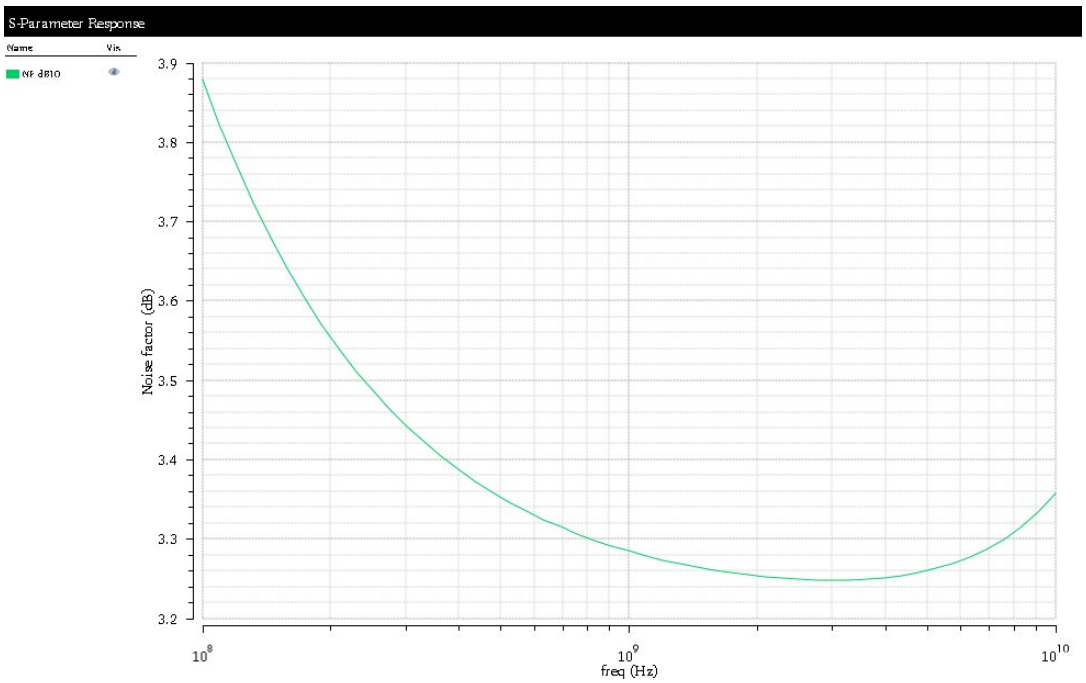

Figure I.33: NF analysis in Wideband Balancing Optimization.

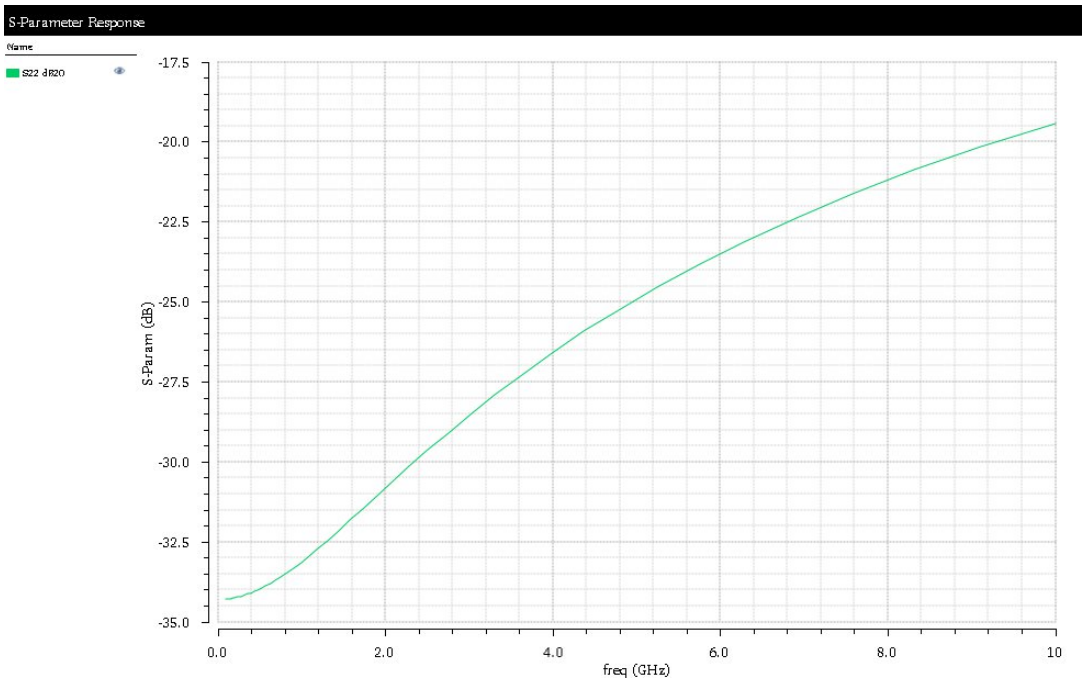

Figure I.34: S_{22} analysis in Wideband Balancing Optimization.

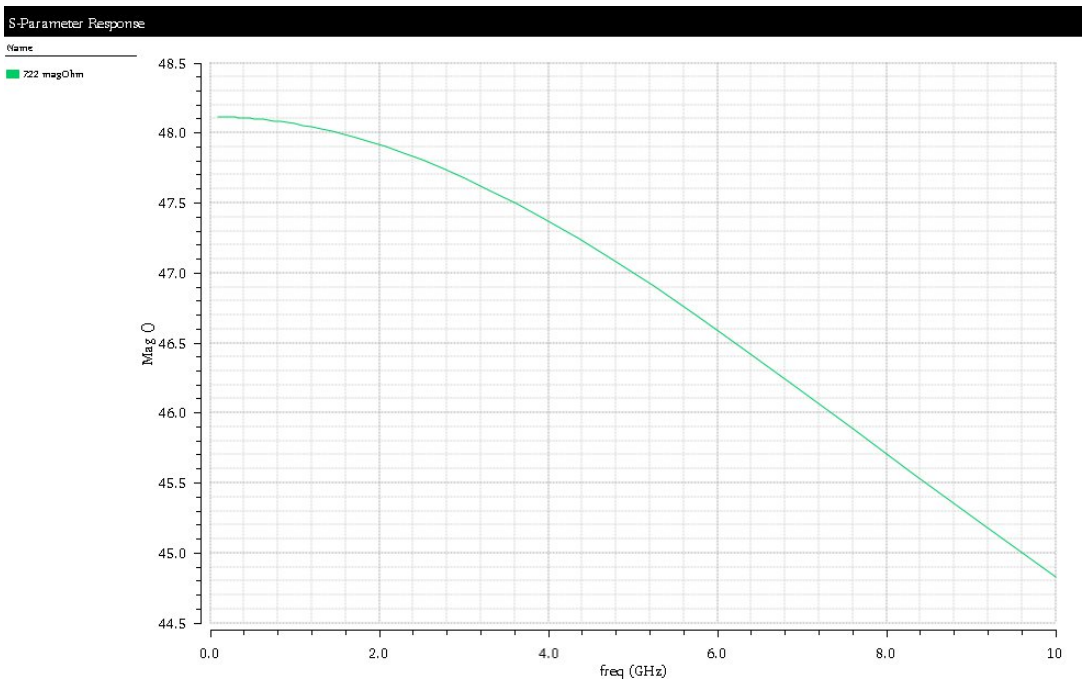

Figure I.35: Z_{22} analysis in Wideband Balancing Optimization.

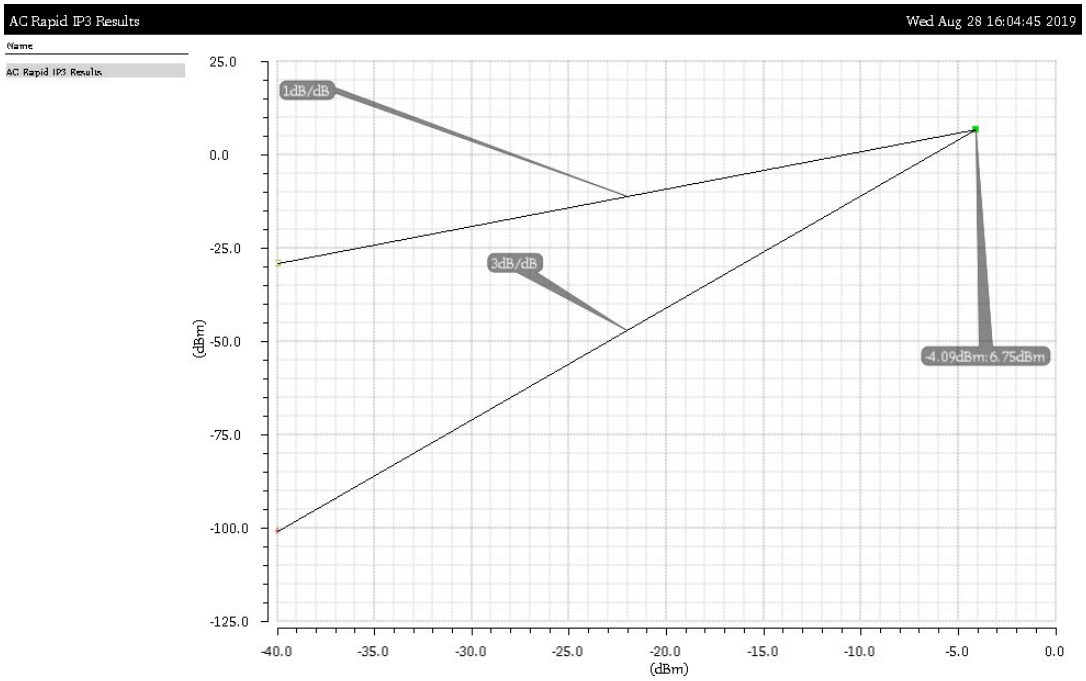

Figure I.36: IP3 analysis in Wideband Balancing Optimization.

I.5 Active Loads LNA-Wideband Optimization

In this section are presented the analysis results from the best individual provided by the wideband optimization method considering the changes applied to the circuit's loads.

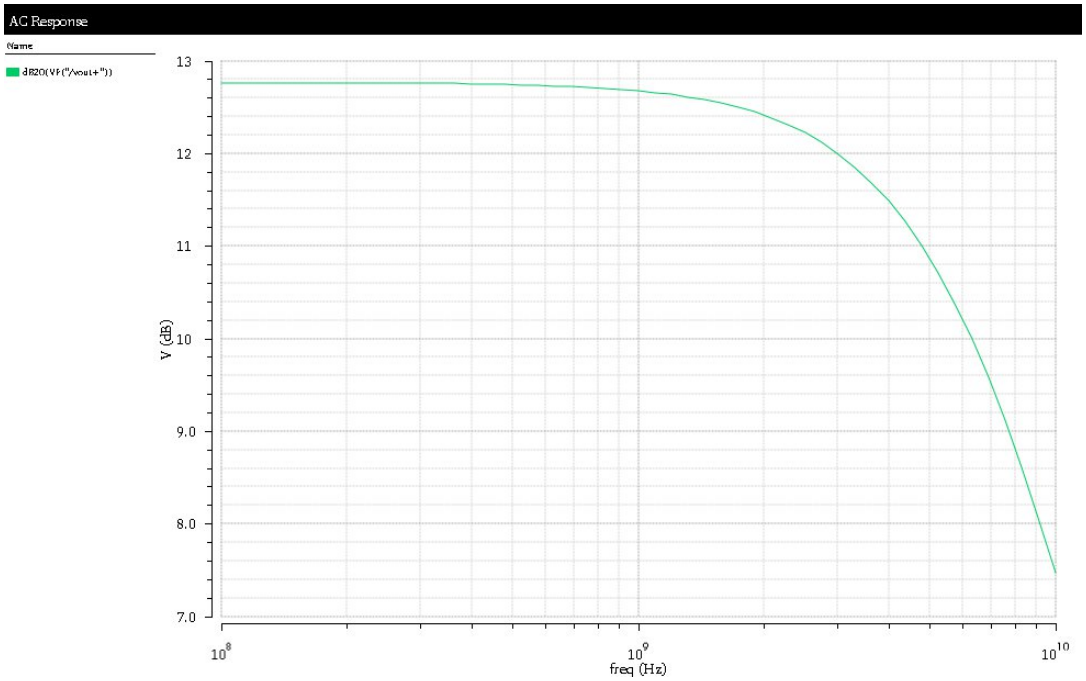

Figure I.37: A_{VCG} analysis Active Loads Wideband Optimization.

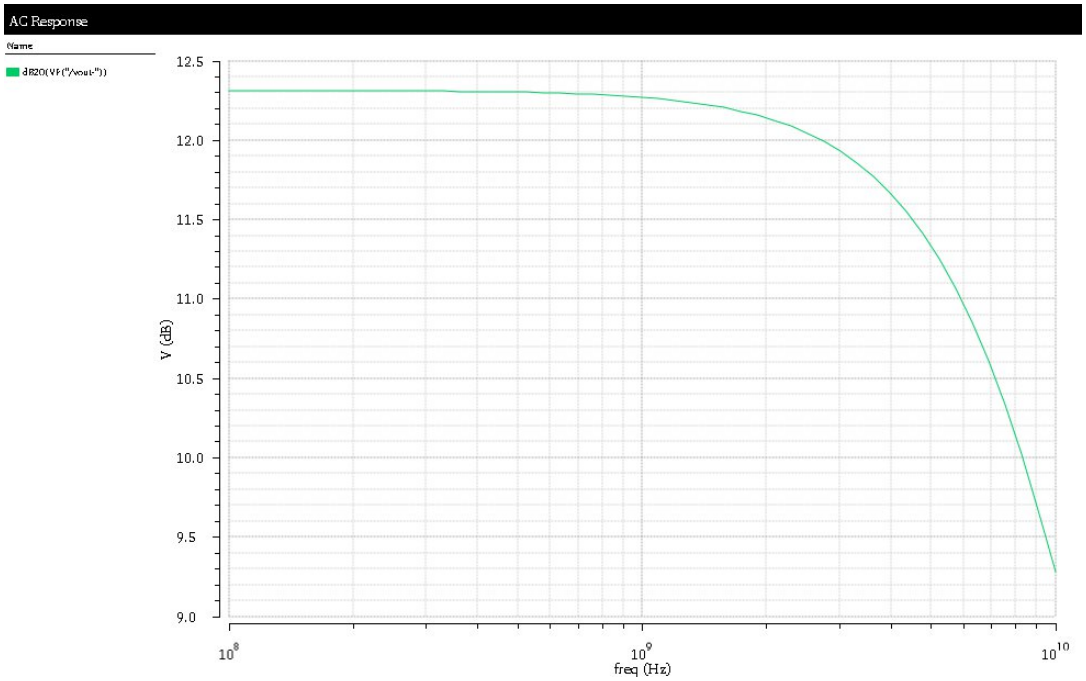

Figure I.38: A_{VCS} analysis Active Loads Wideband Optimization.

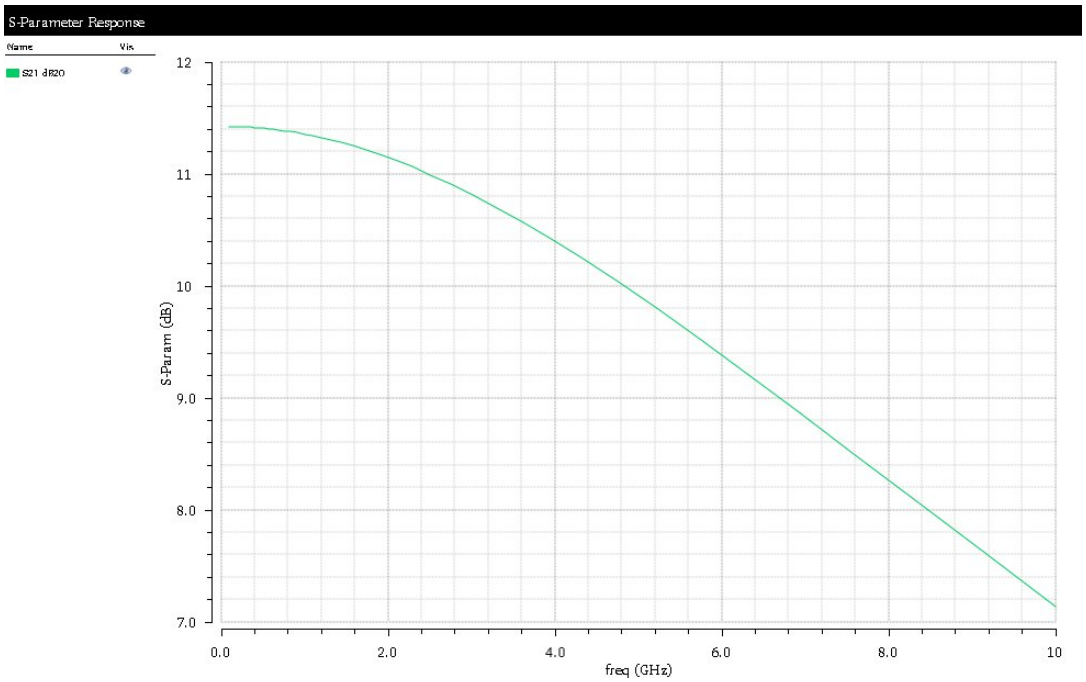

Figure I.39: S_{21} analysis Active Loads Wideband Optimization.

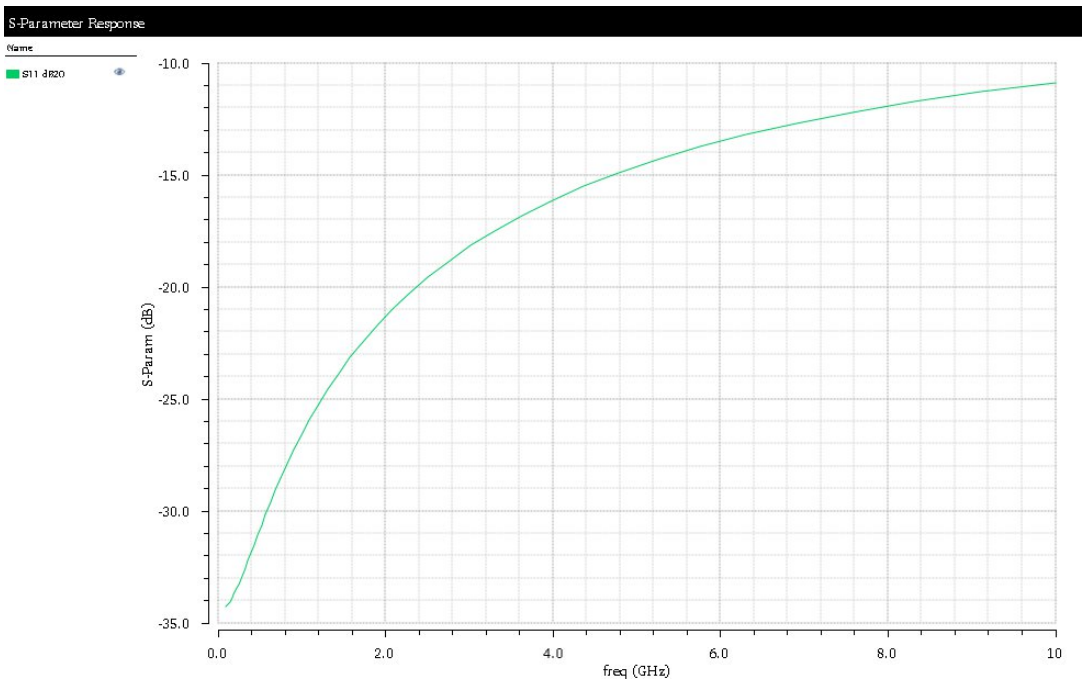

Figure I.40: S_{11} analysis Active Loads Wideband Optimization.

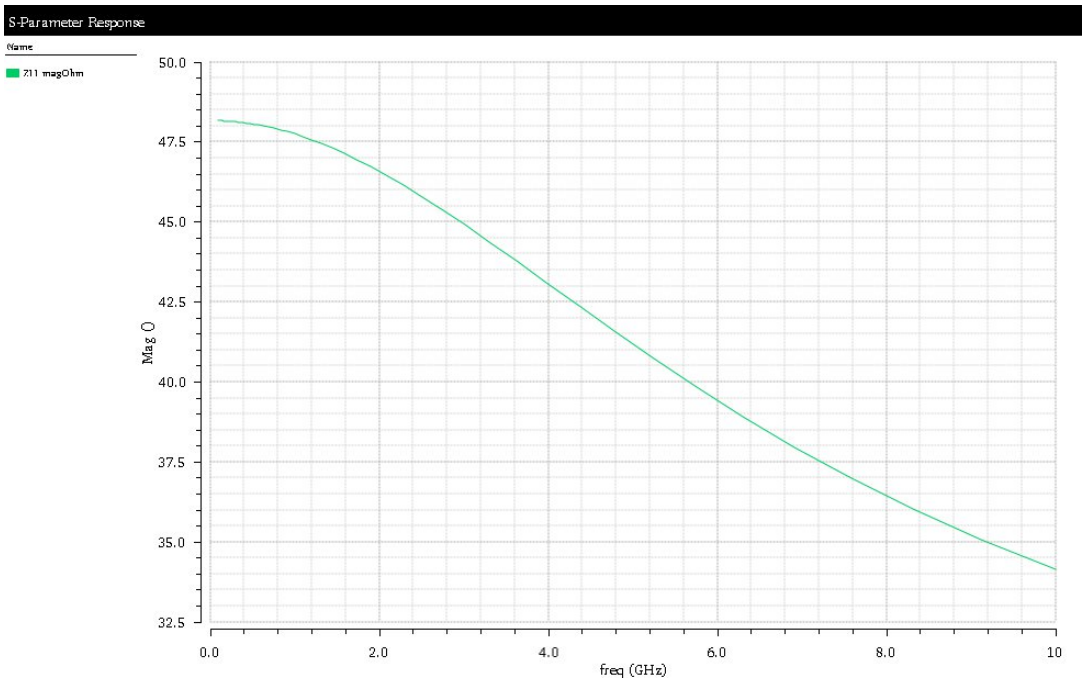

Figure I.41: Z_{11} analysis Active Loads Wideband Optimization.

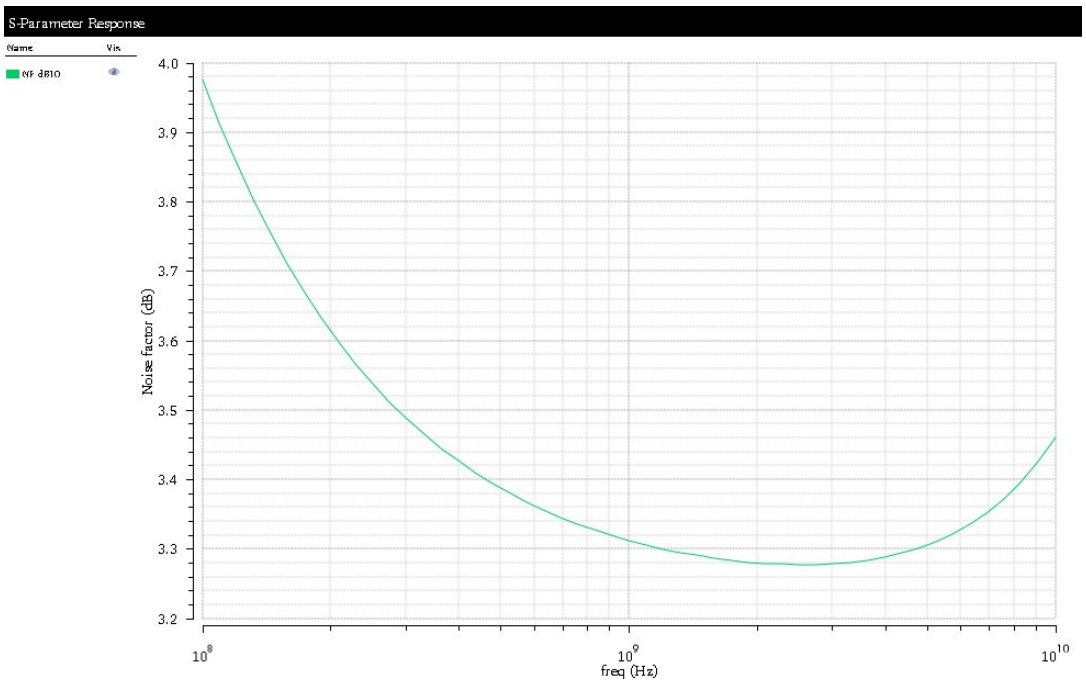

Figure I.42: NF analysis Active Loads Wideband Optimization.

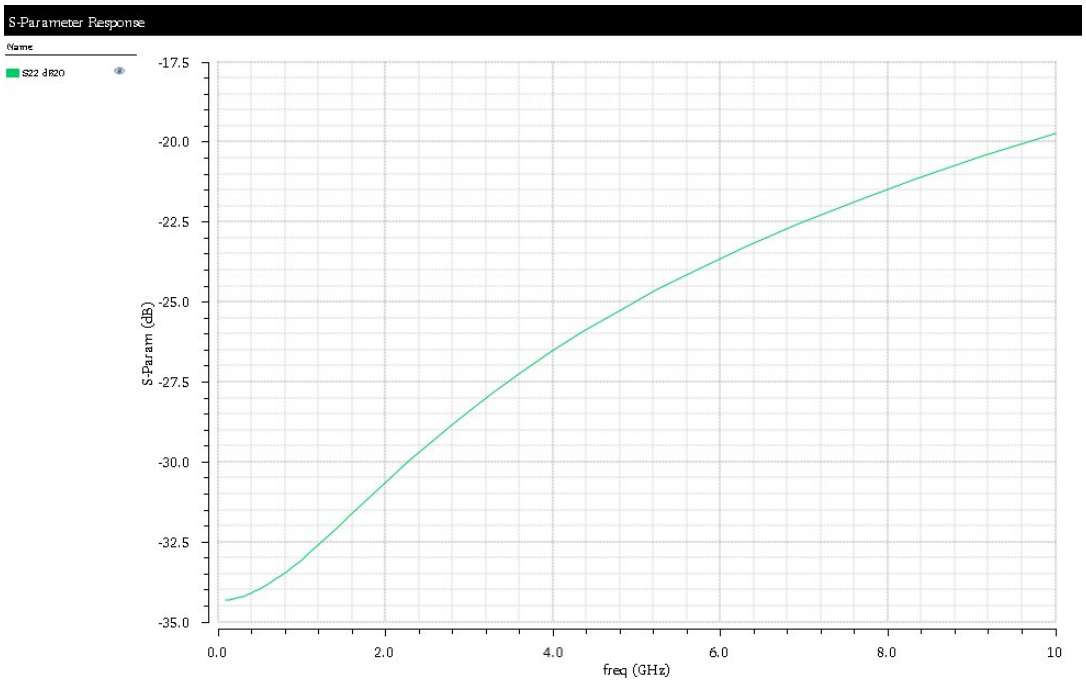

Figure I.43: S_{22} analysis Active Loads Wideband Optimization.

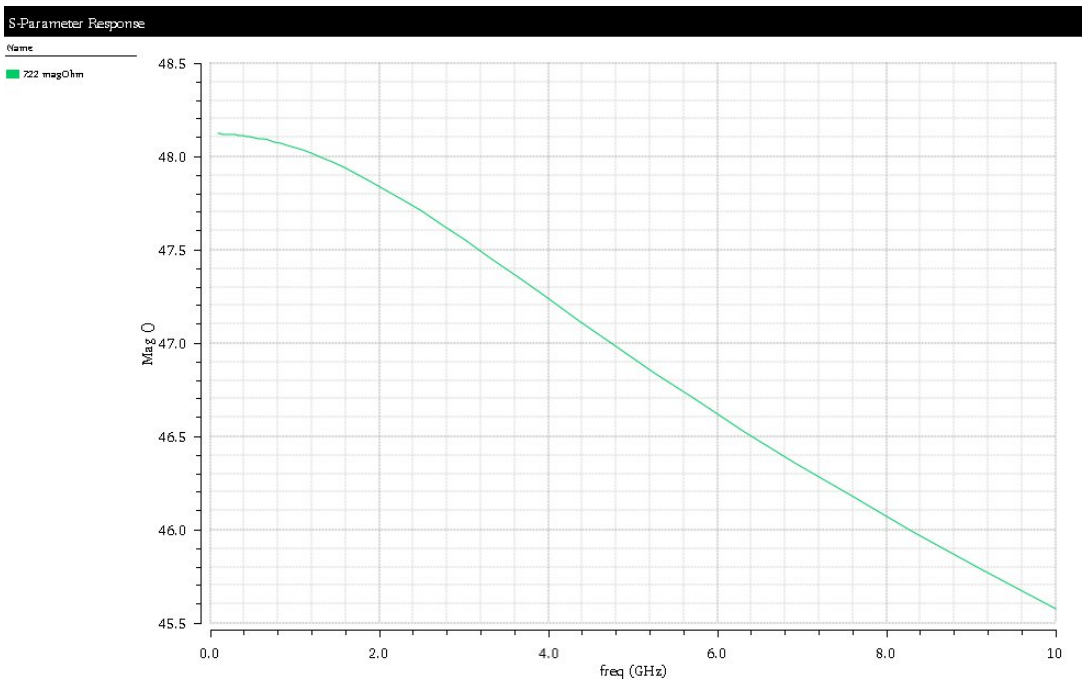

Figure I.44: Z_{22} analysis Active Loads Wideband Optimization.

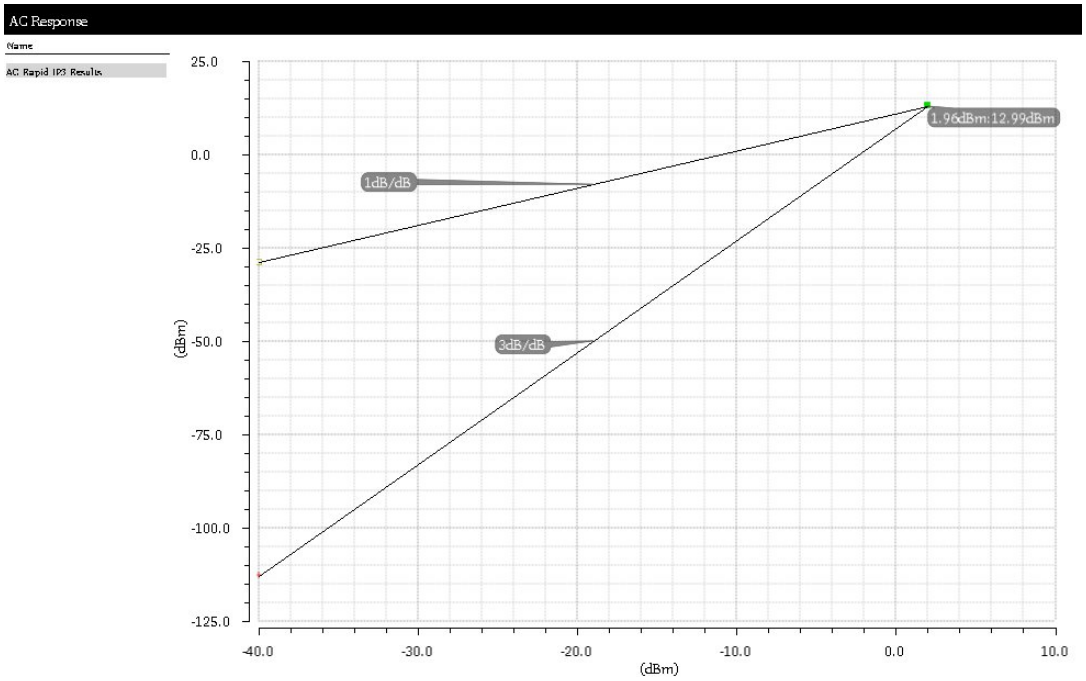

Figure I.45: IP3 analysis Active Loads Wideband Optimization.

I.6 Active Loads LNA- Wideband Optimization with Self-Polarization

In this section are presented the analysis results from the best individual provided by the wideband optimization with self-polarization method considering the changes applied to the circuit's loads.

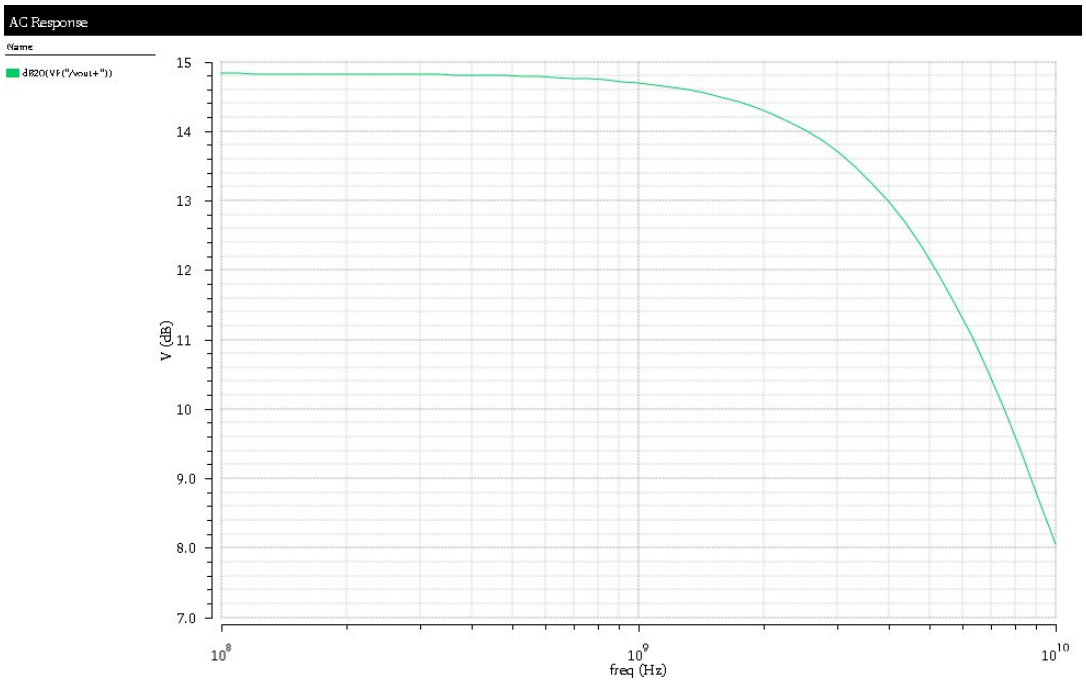

Figure I.46: A_{VCG} analysis in Active Loads with Self-Polarization.

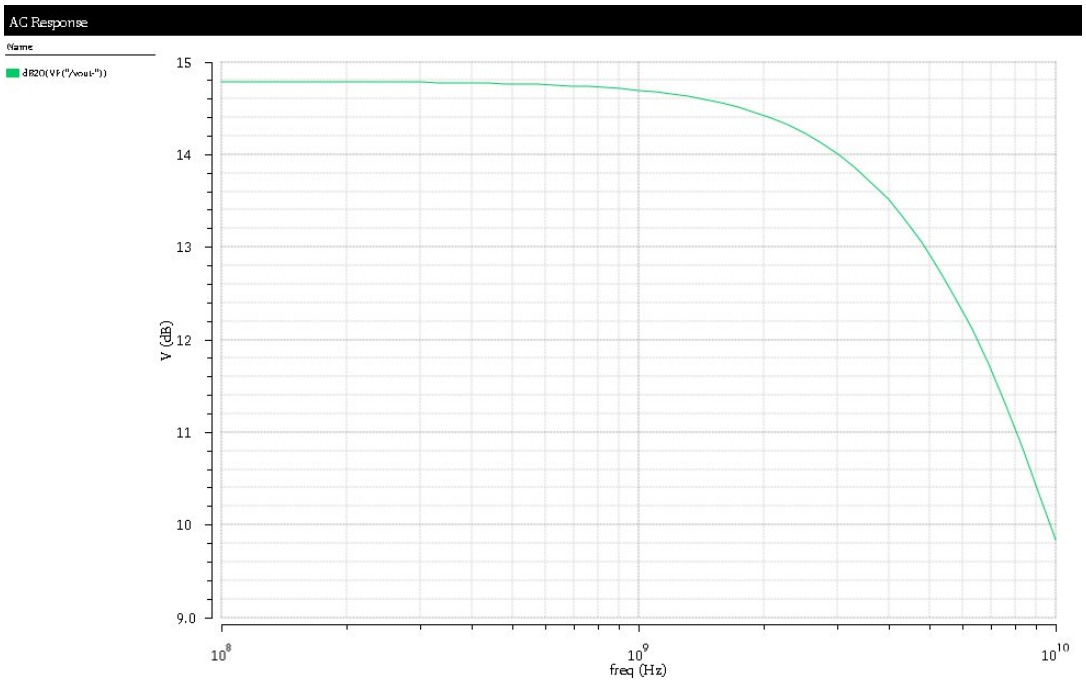

Figure I.47: A_{VCS} analysis in Active Loads with Self-Polarization.

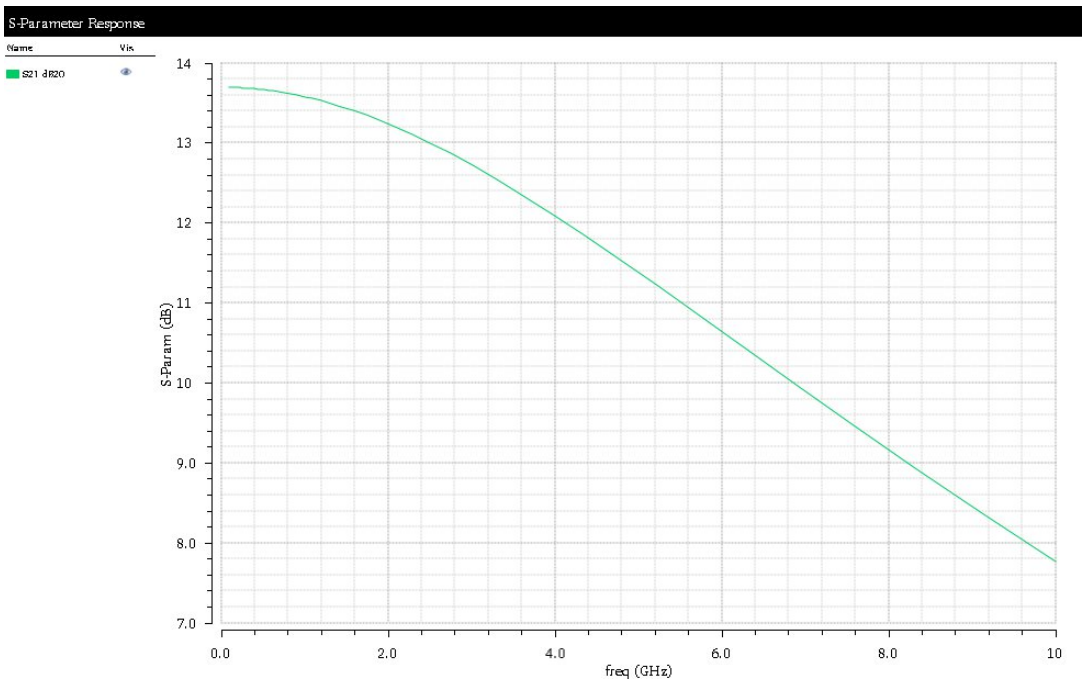

Figure I.48: S_{21} analysis in Active Loads with Self-Polarization.

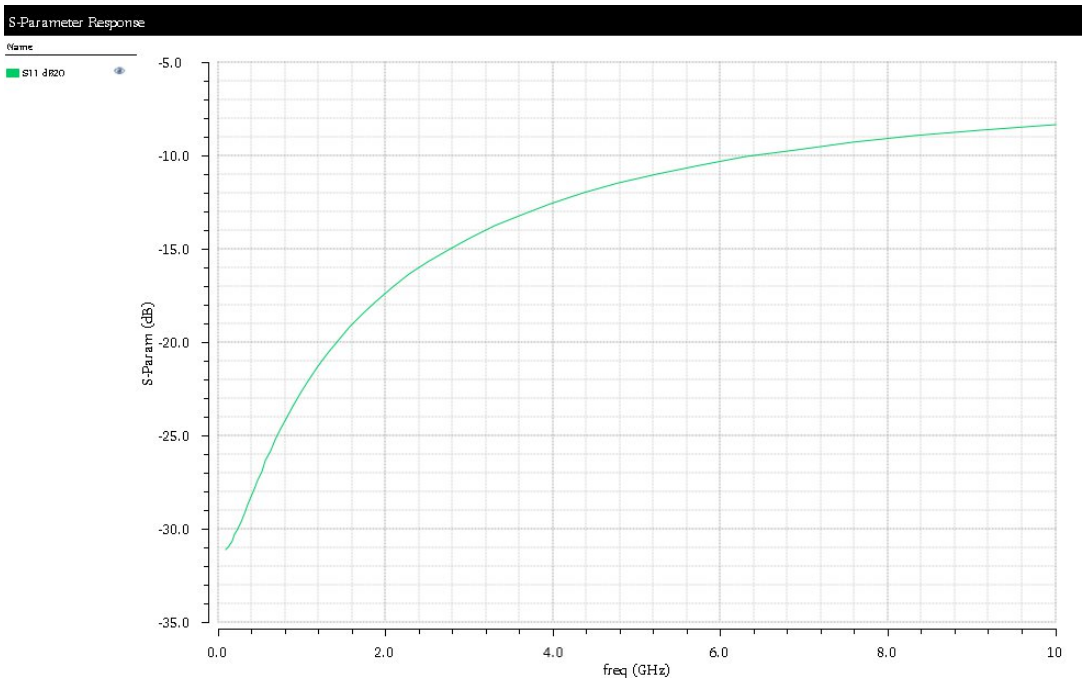

Figure I.49: S_{11} analysis in Active Loads with Self-Polarization.

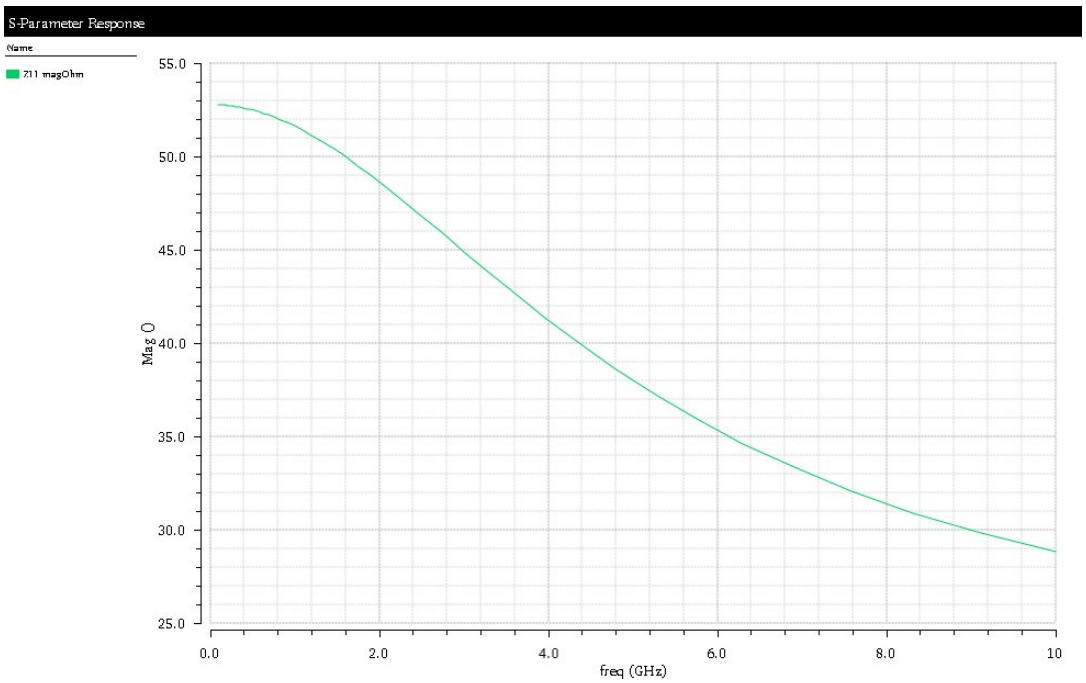

Figure I.50: Z_{11} analysis in Active Loads with Self-Polarization.

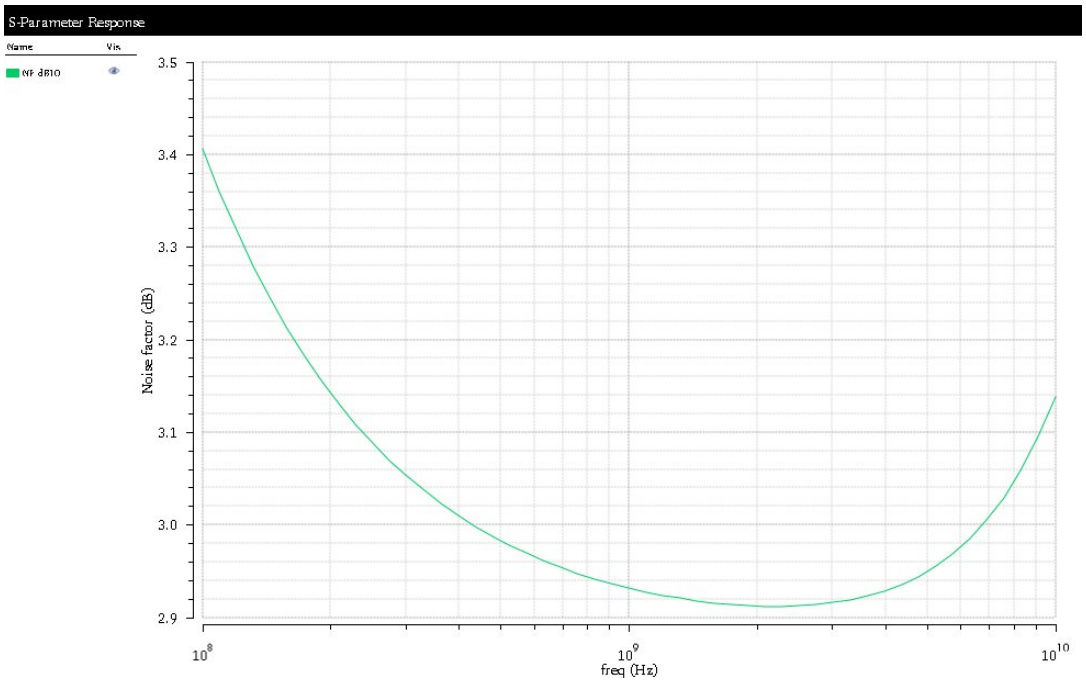

Figure I.51: NF analysis in Active Loads with Self-Polarization.

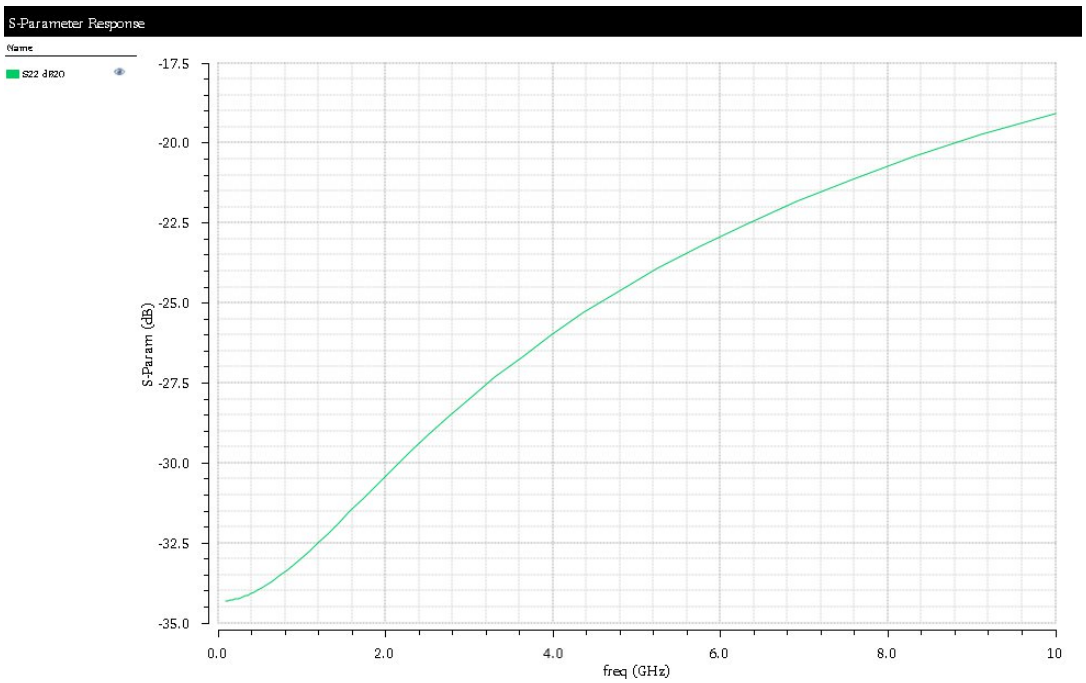

Figure I.52: S_{22} analysis in Active Loads with Self-Polarization.

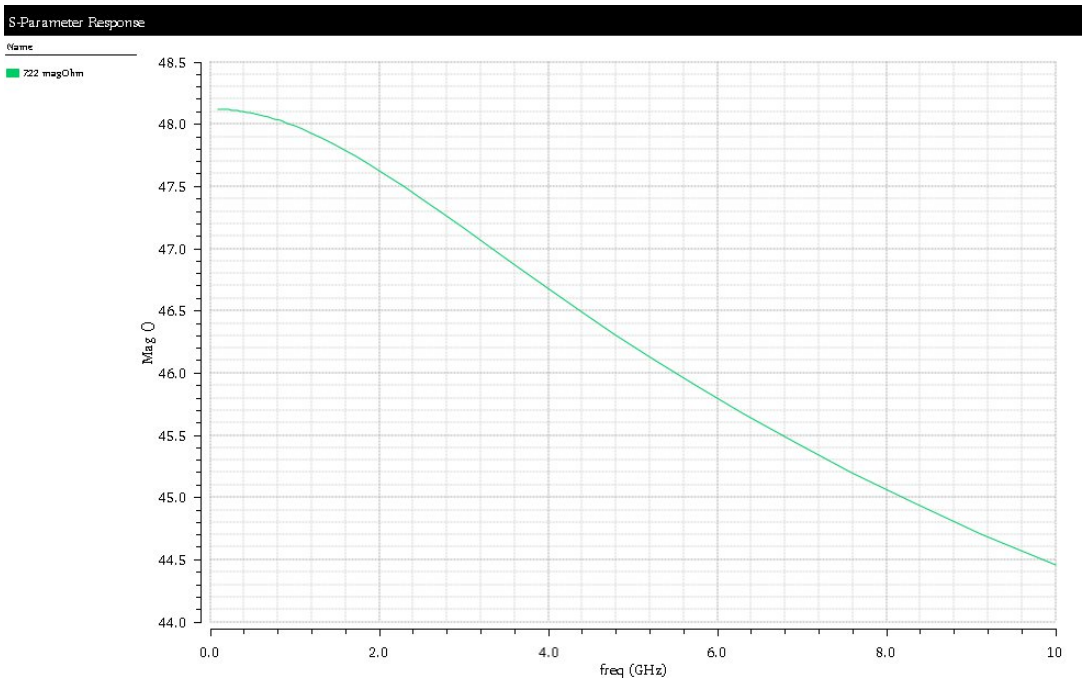

Figure I.53: Z_{22} analysis in Active Loads with Self-Polarization.

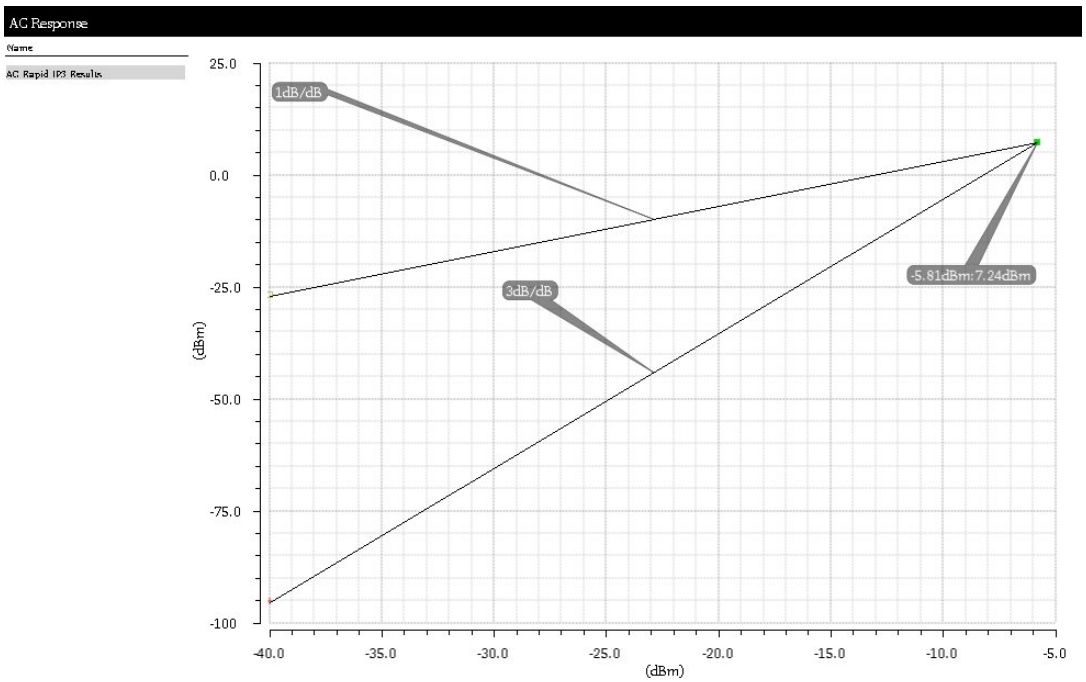

Figure I.54: IP3 analysis in Active Loads with Self-Polarization.