Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Thesis in Ingegneria Elettronica

Machine Learning techniques for the design automation of microwave circuits

Master Candidate

Davide Rossato

Student ID 2026722

Supervisor

Prof. Andrea Neviani

University of Padova

Co-supervisor

Prof. José Carlos Pedro

Universidade de Aveiro

Academic Year
2023/2024
Graduation Date
02/12/2024

Ai miei genitori, da sempre i miei punti di riferimento, la mia forza, e la mia ispirazione.

Alla mia famiglia che, pur numerosa, non mi ha mai fatto mancare il suo sostegno costante e affettuoso, con i miei nonni in primis, che ne fungono da collante e colonne portanti.

Ai miei fratelli d'altre madri, vi devo tutto.

A Giulia, la mia parte razionale.
A tutti i miei amici, e alle persone che mi sono state vicine, vi voglio bene.

Ad Enrico, presenza costante nella mia vita.
Agli amici Unipd che mi hanno sempre accompagnato dal 2017 ad oggi, rendendo più leggero ogni passo di questo percorso.

Agli amici di Aveiro, con cui ho condiviso sei fantastici mesi della mia vita, siete una seconda famiglia.

"I miei migliori amici sono la mia fortuna, so di non capirli fino in fondo, li scopro giorno per giorno ed è questo che mi culla."

Contents

Chapter 1

Introduction

1.0.1 Outline

As can be seen in 1.1, microwaves are essential to modern technology, playing critical roles in communication, military, medical, and scientific applications, as well as in household appliances. They are key in radars, satellites, automotive, data processing, computing, and are increasingly important for smart cities, aiding in intelligent transportation, energy management, and healthcare systems. One of the major challenges in this field is ensuring proper impedance matching between devices, as it plays a critical role in optimizing power transfer and minimizing signal loss.

The primary aim of this study is to exploit Machine Learning, specifically Reinforcement Learning, to automate and optimize the process of matching between input and output impedance in microwave circuits.
The project begin with an exploration of reinforcement learning techniques in Matlab, initially focusing on maze-solving to establish a foundation for developing and adapting code for the impedance matching problem. Given an input and an output impedance, the goal is to allow reinforcement learning to handle the calculation of internal impedances and finally output the set of inductors and capacitors that best approaches the final matching, that is achieving a higher reward.
It then progress to a more in-depth exploration in Python, including work with OpenAI Gym and its libraries, with training algorithms such as theProximal Policy Optimization (PPO) and Neural networks such as LSTM or RNN layers. Although a final working solution was not reached, this project represents an in-depth exploration of the challenges involved in implementing RL in this context. Through numerous attempts and iterative improvements, valuable insights were gained into the constraints, limitations, and opportunities that arise when combining advanced computational techniques with classical engineering problems. This thesis documents these

efforts, highlighting both the progress made and the challenges that remain in this complex area of research.

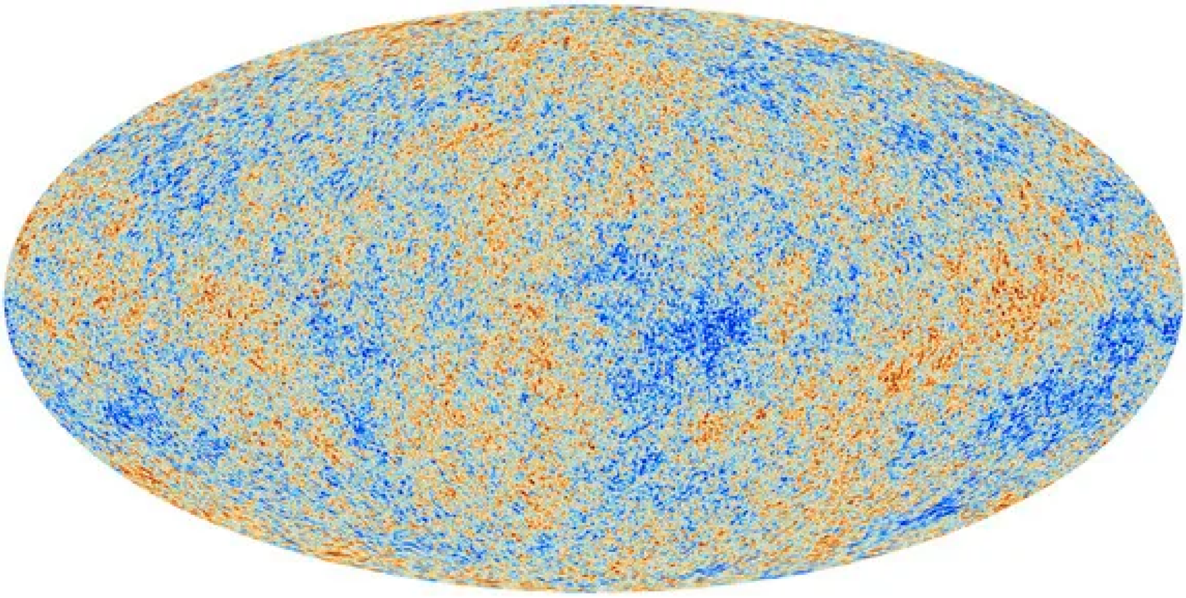


Figure 1.1: This image unveiled March 21, 2013, shows the cosmic microwave background (CMB) as observed by the European Space Agency's Planck space observatory. (Image credit: ESA and the Planck Collaboration)

1.1 Microwaves

1.1.1 Theory and definition

To better understand the challenges and techniques discussed in this thesis, it is essential to first define the fundamental concepts of microwave theory. This section provides an overview of the principles governing microwave signals, their properties, and their applications, serving as the foundation for the advanced topics explored in later chapters Microwaves are a type of electromagnetic radiation with wavelengths shorter than those of traditional radio waves but longer than infrared waves. Electromagnetic radiation can be transmitted either as waves or particles, with varying wavelengths and frequencies. This wide range of wavelengths is referred to as the electromagnetic spectrum (EM spectrum).

The spectrum is typically divided into seven regions (1.2), organized by decreasing wavelength and increasing energy and frequency. These regions include radio waves, microwaves, infrared (IR), visible light, ultraviolet (UV), X-rays, and gamma rays. Microwaves occupy the section of the EM spectrum between radio waves and infrared light. They have frequencies ranging from about 1 billion cycles per second, or 1 gigahertz (GHz), up to about 300 gigahertz and wavelengths of about 30 centimeters (12 inches) to 1 millimeter . However, in radio-frequency engineering, microwaves are often defined more narrowly, usually covering the frequency range from 1 to 100 GHz (with wavelengths between 30 cm and 3 mm), or in some cases, from 1 to 3000 GHz (wavelengths between 30 cm and 0.1 mm). Microwaves are extensively utilized in modern technology for various applications, such as point-to-point communication links, wireless networks, microwave radio relay systems, radar, satellite and spacecraft communications. They are also used in medical diathermy and cancer treatment, remote sensing, radio astronomy, particle accelerators, spectroscopy, industrial heating, collision avoidance systems, as well as in everyday devices like garage door openers, keyless entry systems, and microwave ovens for cooking food.

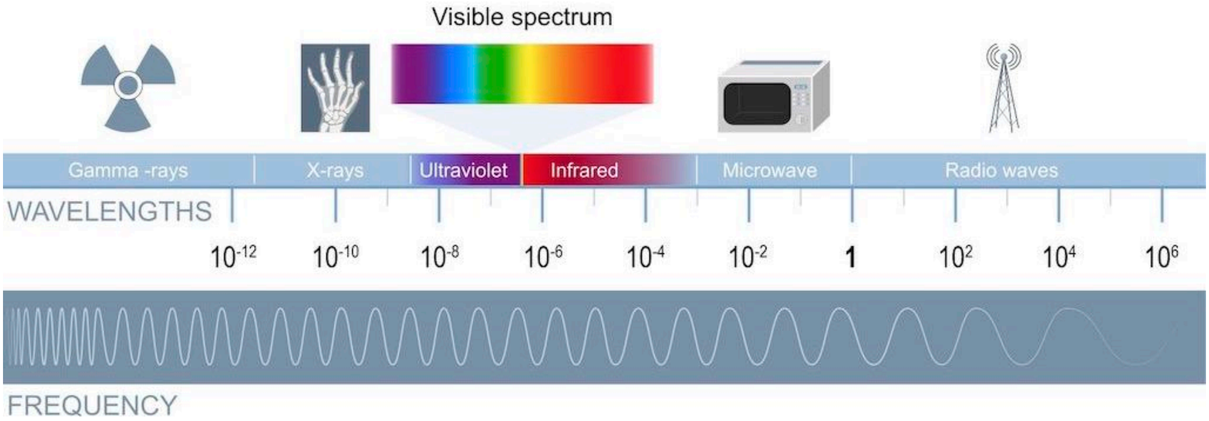


Figure 1.2: Electromagnetic spectrum, from highest to lowest frequency waves ((Image credit: Shutterstock)

1.1.2 Frequency bands

Frequency bands within the microwave spectrum are labeled with letters (1.3). However, there are multiple, conflicting systems for designating these bands, and even within the same system, the frequency ranges associated with certain letters can differ slightly depending on the application. The letter system originated during World War II as part of a classified U.S. radar band classification, giving rise to the earliest system, known as the IEEE radar bands. Here one such set of microwave frequency band designations, established by the Radio Society of Great Britain (RSGB):

- *L band*: 1 to 2 GHz, used in military telemetry, GPS, mobile phones and amateur radio;
- *S band*: 1 to 2GHz, used in weather radar. microwave ovens, mobile phones, wireless LAN Bluetooth, GPS, amateur radio;

- *C band*: 2 to 4GHz, used in long-distance radio telecomunications, wireless LAN, amateur radio
- *X band*: 4 to 8GHz, used in satellites communications, radar, amateur radio, space communications, terrestrial roadband;
- $K_\mu band$: 8 to 12GHz, used in satellites communications, molecular rotatonal spectroscopy;
- *K band*: 12 to 18GHz, used in radar, satellites communications, automotive radar, molecular rotatonal spectroscopy;
- K_a band: 18 to 26.5GHz, used in satellites communications, molecular rotatonal spectroscopy;
- *Q band*: 26.5 to 40GHz, used in molecular rotatonal spectroscopy, satellite communications, terrestrial microwave communications;
- *U band*: 33 to 50GHz: used in cavity Fourier transform microwave (FTMW) spectroscopy and used for molecular measurements;
- *V band*: 40 to 60GHz, used in millimeter wave radar research, molecular rotational spectroscopy and other kinds of scientific research;
- *W band*: 50 to 75GHz, used in satellite communications, millimeter-wave radar research, military radar targeting and tracking applications, and some non-military applications, automotive radar;
- *F band*: 75 to 110GHz, used in SHF transmissions: Radio astronomy, microwave devices/communications, wireless LAN, most modern radars, communications satellites;
- *D band*: 110 to 170GHz, used in EHF transmissions: Radio astronomy, high-frequency microwave radio relay, microwave remote sensing, amateur radio, directed-energy weapon, millimeter wave scanner.

The division of K-band waves into a lower band K_μ and an upper band K_a dates back to World War II, because at the time radar technology was first developed on K band, it was not known that there was a nearby absorption band caused by water vapor and oxygen in the atmosphere.

1.1.3 Propagation

Microwaves only propagate through direct line-of-sight paths. Unlike lower frequency radio waves, they do not travel as ground waves that follow the Earth's surface or reflect off the ionosphere, known as skywaves. This

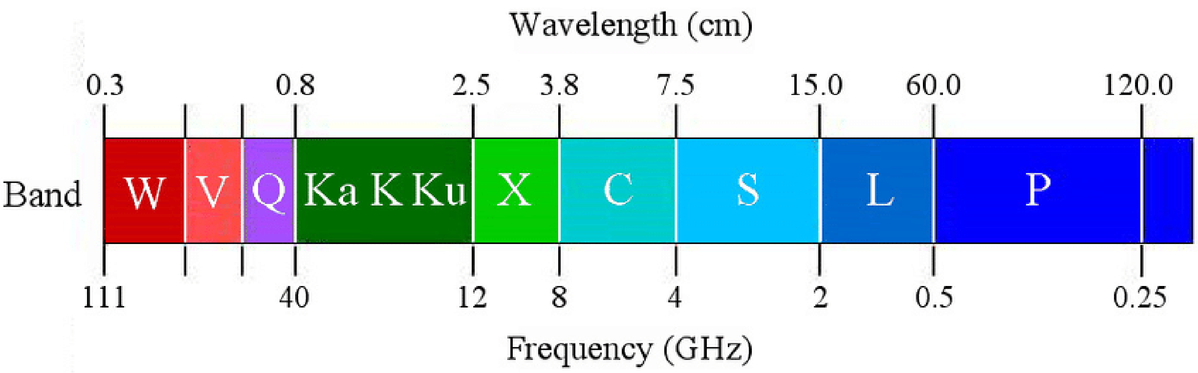


Figure 1.3: Band designation of microwave spectrum used for SAR.

means that their transmission range is limited to the distance where both the transmitting and receiving antennas have a clear view of each other, typically up to 30-50 km under normal conditions, depending on the height of the antennas. While lower-frequency microwaves can penetrate building walls sufficiently for usable reception, it is generally necessary to have rights of way cleared up to the first Fresnel zone for optimal transmission. A problematic scenario could be tha shading effect on mountains. Especially in areas with complex terrain, the signal path can be seriously affected. Therefore, when we are located in mountainous areas, we often experience poor signal and no connection to the Internet. The challenges posed by mountainous terrain on microwave communications can be addressed by adjusting the placement of antennas and installing relay stations. Precise calculation of antenna deployment position and angle, as well as setting up repeater stations at critical locations, can effectively bypass terrain obstacles and ensure stable signal transmission. In point-to-point wireless communications, ensuring a clear Fresnel Zone is another essential point to maintain signal strength and quality.

The Fresnel Zone (1.4) is a 3D elliptical region between the transmitting and receiving antennas, through which the majority of the signal passes. It is important that this region remains free from obstructions such as terrain, vegetation, buildings, or wind farms, as any interference within the Fresnel Zone can result in signal degradation or loss. The line of sight (LOS) between the antennas must also be free of obstructions, but the Fresnel Zone expands beyond the direct LOS, covering an area where diffraction and signal bending can occur if blocked. The size of the Fresnel Zone is determined by the frequency of the transmission and the distance between the two communication points. Keeping this zone clear is critical for the optimal performance of the wireless system.

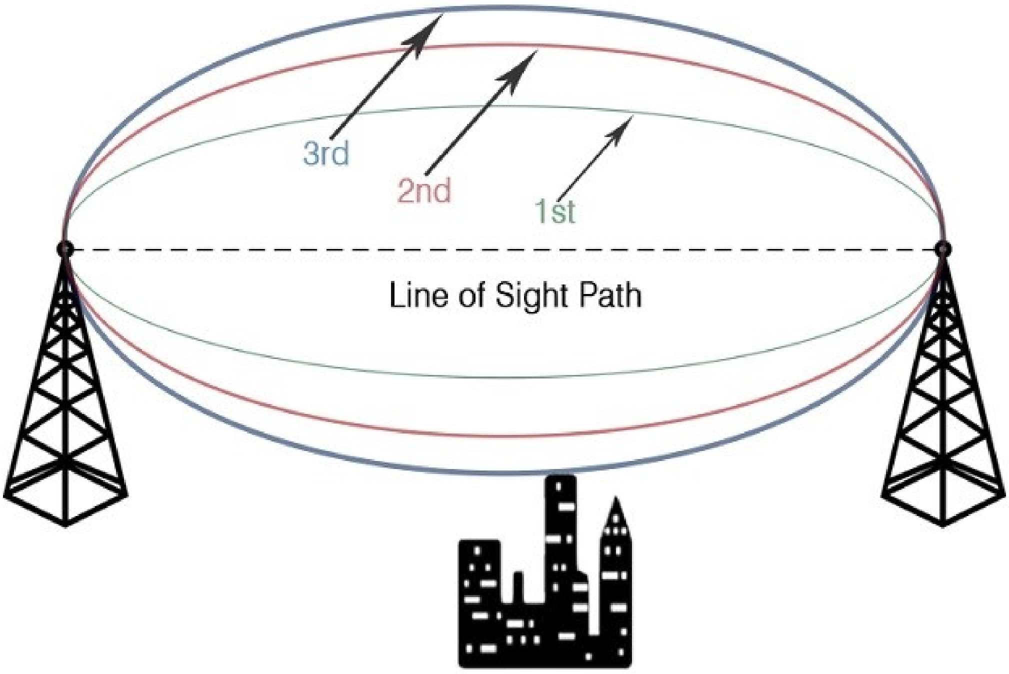


Figure 1.4: The Fresnel zone is made up of multiple zones, with zone 1 having the strongest signal and following zones (Zone 2, and Zone 3) having weaker signals.

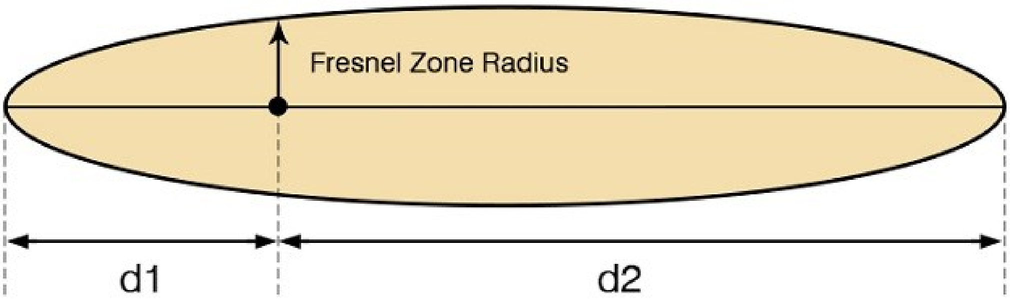


Figure 1.5: Fresnel Zone radius

Based on the figure 1.5 the Radius is calculated using the following equation:

$$R = \sqrt{\frac{nd_1d_2*\lambda}{d_1+d_2}}$$

Where
n= Fresnel zone number (should be greater than zero)
λ=frequency

Atmospheric Effects

Several factors enable microwaves to travel beyond the horizon. One key factor is the atmospheric density profile. Air is denser near the surface of the Earth and becomes thinner at higher altitudes. As shown in Figure 1.6, the atmosphere acts like a prism, continuously bending radio waves back toward the Earth. A prism bends shorter wavelengths more than longer ones, meaning the atmosphere naturally bends microwaves more effectively than UHF, and UHF bends more than VHF, and so on. In terms of frequency band openings, this results in higher frequencies, like 10 GHz, opening first, with the opening gradually extending to lower frequencies such as UHF and eventually VHF. Therefore, by the time you notice a tropospheric opening on the 2-meter band, the microwave bands have likely been open for a while. This top-down opening pattern has been confirmed by numerous propagation studies and observations from radar operators. Another atmospheric phenomenon that can reflect microwave signals is water, specifically in the form of raindrops. These small dielectric spheres are roughly the same size as half-wave dipoles at 5.7 GHz and 10 GHz, common frequencies in amateur radio bands. Sending a microwave beam into a rainstorm is similar to pointing a spotlight into a snowstorm: the signal scatters in all directions (*rain fade*). At frequencies above 100 GHz, the Earth's atmosphere absorbs electromagnetic radiation so effectively that it becomes essentially opaque. However, the atmosphere becomes transparent again at certain frequencies known as the infrared and optical window ranges. The effects of rain fade can be reduced using methods like station diversity, uplink power control, variable rate coding, larger receiving antennas, and hydrophobic coatings. These approaches improve the system's resistance to signal interference, help maintain continuous and stable communication, and ensure strong performance even during extreme weather. Another noteworthy effect is related to the rain, and it is obtained when 10-GHz signals reflect off moving objects, obtaining a *Doppler shift*, which is a perceived change in frequency depending on your transmission frequency, the speed of the moving object, and whether it's moving toward or away from you. So, the Doppler shift from billions of moving raindrops results in billions of different frequencies. Summarizing, microwave propagation is highly dependent on line-of-sight, atmospheric conditions, and the presence of obstacles. While microwaves offer high data transmission capacity and are essential in modern communication technologies, their range and reliability can be affected by environmental factors like weather and physical obstructions. To ensure efficient microwave communication, careful system design is essential, including selecting the right frequency, positioning antennas correctly, and using repeaters or satellites when necessary.

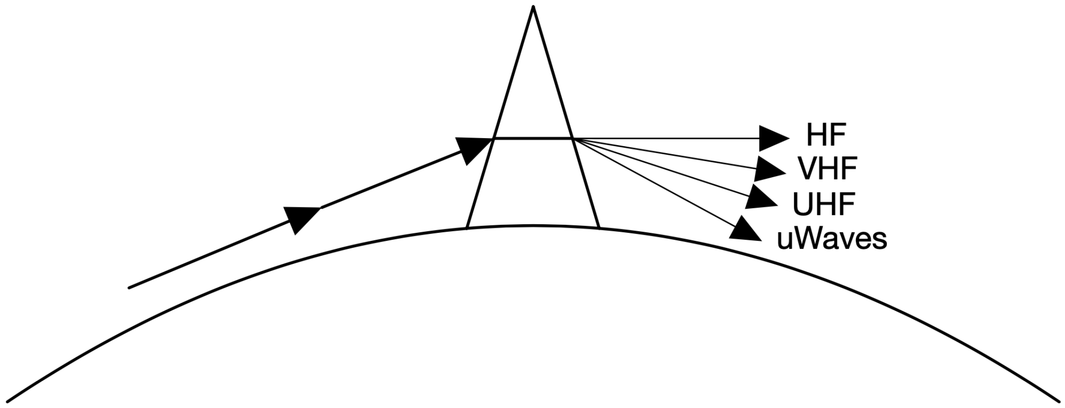


Figure 1.6: Density Profile of the Atmosphere. The air is denser near the ground and gets thinner as you go UP. The air itself is like a prism, continuously bending radio waves back towards the Earth. Since a prism bends smaller waves more than longer waves, it is most efficent at microwave frequencies

1.2 Applications and importance of microwave devices

1.2.1 Active and passive components

This section deals with the microwave devices and components that are commonly used in front end of microwave system. These include the passive and active components. Active and passive microwave devices and components are the essential building blocks of microwave circuits and systems that operate in the frequency range from 300 MHz to 300 GHz (corresponding to wavelengths of 1 m to 1 mm in free space). Generally, in electrical circuits, active components are those that supply or generate energy in the form of voltage or current, while passive components consume or store energy in these forms. Examples of active components include diodes, transistors, silicon-controlled rectifiers (SCR), and integrated circuits. In contrast, resistors, capacitors, inductors, and transmission lines are typical examples of passive components. Active components can provide power amplification and regulate the flow of current, whereas passive components cannot amplify power and are unable to control current flow. Simply put, active components act as energy providers and need an external power source to operate, while passive components function as energy receivers and do not require an external source for their operation.

In many microwave texts, active components are commonly referred to as microwave devices, while passive components are simply called microwave components.

1.2.2 Microwave devices

The term ”microwave devices” refers to components such as oscillators, generators, amplifiers, mixers (both up and down converters), and detectors that are commonly used in microwave circuits. These devices are typically classified into two types:

- *microwave solid-state diodes and transistors*, designed for low-power signals;
- *microwave vacuum tubes*, used for high-power signals.

To the first group belongs the *Backward Diode (BWD)* (1.7), one of the microwave semiconductor devices which are used as an oscillator and mixer. The backward diode is made of gallium arsenide semiconductor, and it works under frequencies of 200 GHz and at low input power and it provides the high output power.

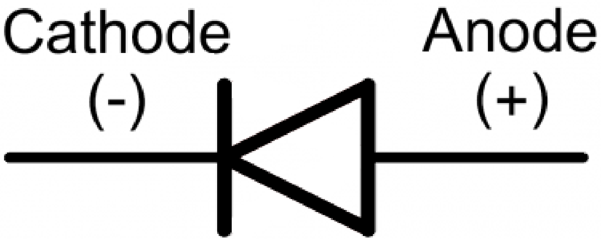


Figure 1.7: Backward Diode symbol

Another semiconductor device used for oscillators is the *Gunn Diode*, a type of diode having negative resistance. It is named after a British physicist J.B Gunn who discovered the “Gunn Effect” in 1962. (sometimes is also used as an amplifier). It consists of three N-type layers; two of them which are on the terminal’s side have a higher doping concentration whereas the middle thin layer has a lighter doping concentration. It works on range of frequency from 4GHz to 100GHz, and it’s generally made of gallium arsenide or indium phosphate mixed with the silicon (1.8).

The third type of diode is the *Impatt diode* (impact avalanche transit time diode, 1.9), it is used for oscillation and amplification, and the higher range of frequency is 200 GHz.

In the context of rectification for microwave length of frequencies, the most used is the *Schottky diode* (1.10), named after a German physicist Walter H. Schottky, is a type of diode which consists of a small junction between an N-type semiconductor and a metal. It has no P-N junction. The frequency range is from 3MHz to 10GHz, and in some cases is also used for switching and mixing purposes also.

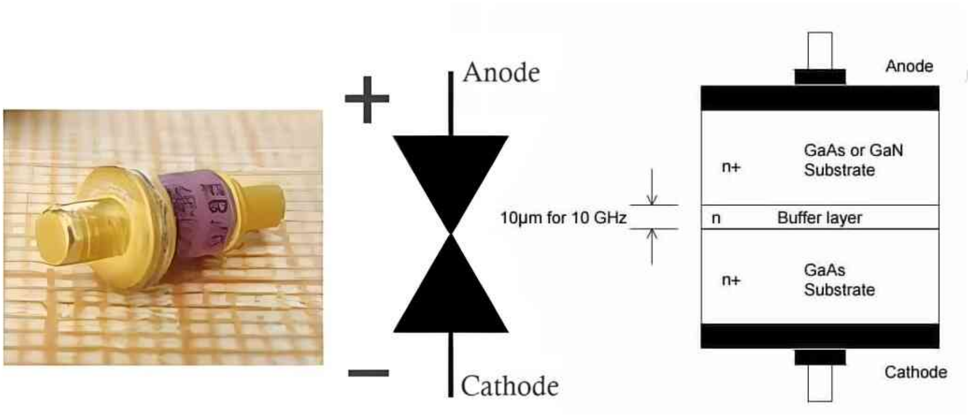


Figure 1.8: Gunn Diode

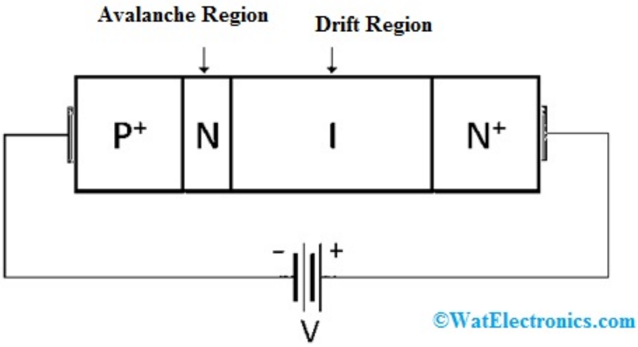


Figure 1.9: Impatt Diode

The most versatile diode is the *Tunnel diode* (1.11), used for oscillation, amplification, mixing and switching purpose. It was invented by Leo Esaki in 1958 for which he received Nobel prize in 1973, which is why it is also known as Esaki diode. A tunnel diode is a heavily doped P-N junction diode. It works on the principle of the tunneling effect. Due to heavy doping concentration, the junction barrier becomes very thin. This allows the electron to easily escape through the barrier. This phenomenon is known as tunneling effect. The Tunnel diode has a region in its VI curve where the current decreases as the voltage increases. This region is known as the negative resistance regionThe frequency range of tunnel diode is up to 100 GHz.

For the television and F.M receiver circuits the most utilized is the *Varactor diode* (1.12), also used in FM transmitted circuits. The varactor diode operates at frequencies up to 105 GHz, with its capacitance able to change in response to variations in the applied bias voltage.

Also the transitors belong to the first group, but only the *FET* ones (Field effect transistor) made of gallium arsenide, due to its efficient energy bands for very high frequency. Their purpose is the amplification of the high

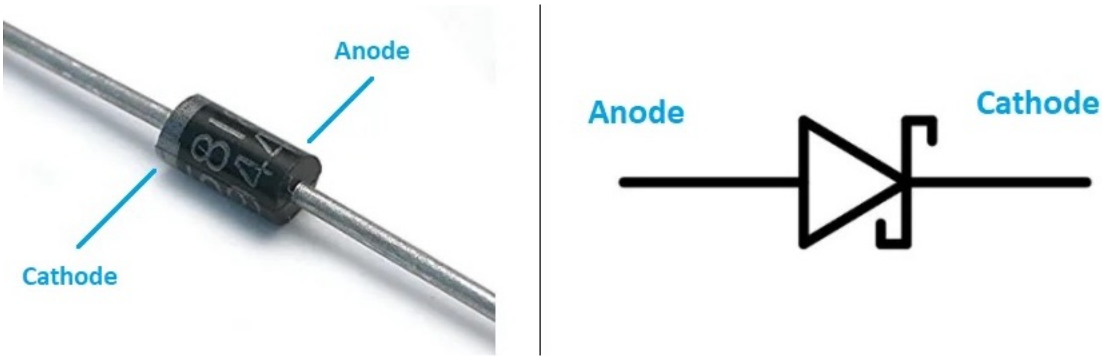


Figure 1.10: Schottky Diode

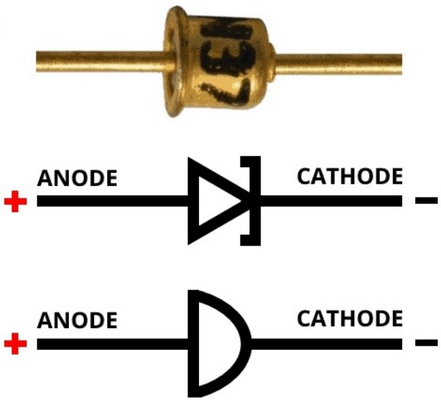


Figure 1.11: Tunnel Diode

frequencies (1.13).

Another type of microwave device belonging to the first group are the *Integrated circuits (IC)*, in particular the hybrid ones. Since inductance and capacitance at very high frequencies are expected to be quite small, the physical size of inductors and capacitors also becomes minimal at these frequencies. This makes it feasible to manufacture hybrid integrated circuits specifically for microwave frequency applications.

As previously mentioned, the second category of microwave devices consists of microwave vacuum tubes. These tubes are used to control a larger signal with a smaller one, allowing for functions such as amplification, oscillation, switching, and other operations. They are primarily employed for generating high-power microwaves. Unlike low-frequency vacuum tubes, these devices rely on the ballistic movement of electrons in a vacuum, controlled by electric or magnetic fields. Examples include the magnetron (used in microwave ovens), klystron, traveling-wave tube (TWT), and gyrotron. These devices operate in a density-modulated mode, meaning they function by modulating electron bunches passing through the tube, rather than relying on a continuous electron stream, as in current-modulated systems.

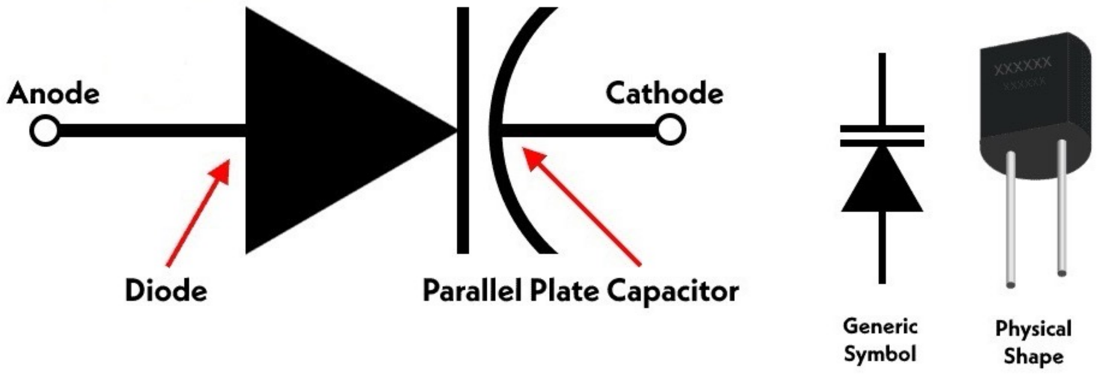


Figure 1.12: Varactor Diode

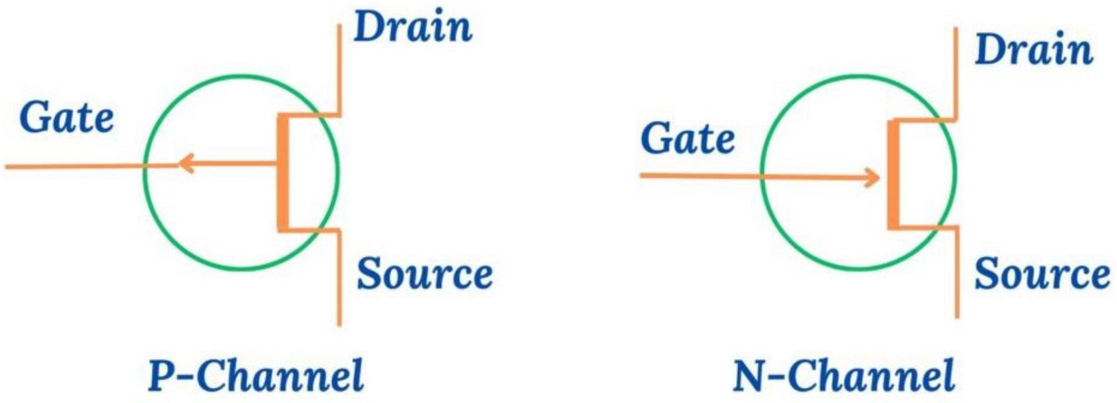


Figure 1.13: FET transistor

1.2.3 Applications

Microwave solid-state diodes, transistors, and integrated circuits play a crucial role in the manipulation of low-power signals for a wide range of applications. From communication systems and radar to satellite technology and medical devices, these components enable high-frequency operations with efficiency, precision, and minimal power consumption.
In Partiucular:

- **Schottky Diodes**
  - *Characteristics*: Fast switching speeds, low forward voltage drop, and low capacitance, making them ideal for high-frequency applications. The limitation of Schottky diode is that it has low reverse breakdown voltage and high reverse leakage current.
  - *Applications*: Mixers and detectors in radar and communication systems, frequency multipliers, and high-speed switching circuits.
- **Gunn Diodes**
  - *Characteristics*: Generate microwave oscillations using the Gunn

effect, without requiring a p-n junction.

  - *Applications*: Local oscillators in radar systems, microwave transmitters, and frequency generation circuits.

- **IMPATT Diodes (Impact Avalanche Transit-Time Diodes)**
  - *Characteristics*: High-power microwave signal generation through avalanche multiplication and transit-time effects.
  - *Applications*: High-power microwave transmitters and radar systems, collision-avoidance radar, and microwave communication links.

- **Varactor Diodes**
  - *Characteristics*: The varactor diode operates under reverse bias conditions. The depletion layer between the P and N-type material is varied by changing the reverse voltage.
  - *Applications*: The applications of Varactor diodes are as a voltage controlled oscillator in the phase-lock loop, in RF tuning filters and frequency multipliers.

- **Tunnel Diodes**
  - *Characteristics*: Exhibit negative resistance, enabling operation at high speeds for microwave applications.
  - *Applications*: High-speed oscillators and amplifiers for microwave circuits.

- **Backward Diodes**
  - *Characteristics*: A special type of tunnel diode with negative resistance in reverse operation, offering low capacitance and fast response.
  - *Applications*: Used in microwave detectors and mixers due to their low power and noise characteristics, ideal for rectification and detection at microwave frequencies.

- **Field Effect Transistors (FETs)**
  - *Characteristics*: FETs are known for their high-frequency performance, fast switching speeds, high electron mobility, and low noise operation.

  - *Applications*: Amplifiers in communication and radar systems, low-noise amplifiers (LNAs), oscillators, and frequency converters in satellite receivers, 5G networks, and microwave communication links.

- **Integrated Circuits (ICs)**
  - *Characteristics*: ICs integrate multiple components (amplifiers, mixers, oscillators) onto a single chip, offering compact, efficient solutions for microwave systems.
  - *Applications*: Used in radar systems, satellite communication, wireless communication devices, mobile phones, and microwave transceivers. Examples include MMICs (Monolithic Microwave Integrated Circuits) and RFICs (Radio Frequency Integrated Circuits).

Microwave devices play an essential role in many modern technologies due to their ability to operate at high frequencies, providing numerous advantages in fields such as communication, radar, medical systems, and industrial processes. One of the most significant contributions of microwave devices is in high-frequency communication. They are fundamental to wireless networks like 4G and 5G, enabling faster data transmission, greater capacity, and more reliable connections. In addition, microwave frequencies are crucial in satellite communications, supporting global broadcasting, GPS, and long-distance data transfer.

In radar systems, microwave devices are key components in both military and civilian applications. In defense, they enable advanced systems for surveillance, target detection, and missile guidance. Civilian applications include air traffic control, weather monitoring, and automotive radar used in advanced driver-assistance systems (ADAS), which enhance both safety and efficiency.

Medical applications also benefit significantly from microwave technology. These devices are used in diagnostic imaging techniques, such as MRI, and in therapeutic procedures like hyperthermia therapy for cancer treatment. Moreover, microwave ablation offers a minimally invasive option for destroying tumors by using precise heating, which is gaining popularity in medical procedures.

In industrial applications, microwave devices are widely used for heating and material processing. Beyond everyday appliances like microwave ovens, they are employed in industries for drying, sintering, and non-destructive testing (NDT) to inspect materials without causing damage—this is especially important in sectors like aerospace and construction.

Microwave technology is also indispensable in scientific research. In spectroscopy, microwave devices allow researchers to analyze the properties of molecules and compounds. In astronomy, they are used in radio telescopes to detect cosmic microwave background radiation, contributing to our understanding of the origins and evolution of the universe.

An additional advantage of microwave devices is their contribution to miniaturization and efficiency. Their use enables the design of smaller, more efficient systems, which are crucial in modern consumer electronics, telecommunications, and medical equipment. Moreover, operating at microwave frequencies minimizes signal interference, improving the reliability of communication systems.

Looking forward, future technologies will increasingly rely on microwave devices. With the expansion of 5G networks and the growth of the Internet of Things (IoT), microwave technology will be critical in providing fast and reliable communication between devices. Microwaves are also becoming more important in quantum computing, where they are used to manipulate quantum bits (qubits), which is expected to revolutionize computing power and efficiency.

In conclusion, microwave devices are fundamental to the development of a wide range of modern technologies. Their ability to handle high-frequency signals with precision and efficiency makes them essential in communication, defense, healthcare, scientific research, and industrial applications. As technology continues to evolve, the importance of microwave devices will only grow, driving further innovation and enhancing the quality of life across various sectors.

The wide range of microwave devices, from Gunn diodes and FETs to integrated circuits, highlights the critical role of impedance matching in ensuring efficient operation. However, the static and experimental nature of traditional matching methods limits their adaptability to the nonlinear and frequency-dependent behavior of these components. By implementing reinforcement learning, this thesis demonstrates a innovative approach to automate and optimize impedance matching, enabling real-time adaptability and enhanced performance across diverse microwave systems. The RL framework not only addresses the unique matching challenges posed by each device but also integrates fluency into modern microwave circuit architectures, offering a tunable and efficient solution for next-generation technologies.

1.3 Impedances

1.3.1 Derivation of the impedances

When alternating current is used, the ratio of voltage to current $\frac{V}{I}$ does not always remain constant. This happens because the voltage and current may reach their maximum values at different moments, especially in circuits that contain components like inductors or capacitors. Impedance, represented by the symbol Z, measures the opposition to alternating current presented by the combined effect of resistance and reactance in a circuit, and its unite measure is Ohm $[\Omega]$.
Considering the capacitor C, the main law is:

$$i_C(t) = C\frac{dv_C(t)}{dt}$$

The voltage signal is:

$$v_C(t) = V_p e^{j\omega t}$$

Consequently, by differentiating with respect to time:

$$\frac{dv_C(t)}{dt} = j\omega V_p e^{j\omega t}$$

And so, the impedance of the capacitor is:

$$Z_C = \frac{v_C(t)}{i_C(t)} = \frac{1}{j\omega C}$$

For an inductor, starting from Faraday's law:

$$v_L(t) = L\frac{di_L(t)}{dt}$$

The current signal is:

$$i_L(t) = I_p \sin(\omega t)$$

By differentiating with respect to time:

$$\frac{di_L(t)}{dt} = \omega I_p \cos(\omega t)$$

Thus, the impedance of the inductor is:

$$Z_L = j\omega L$$

Their behaviour is showed in Figure 1.14.

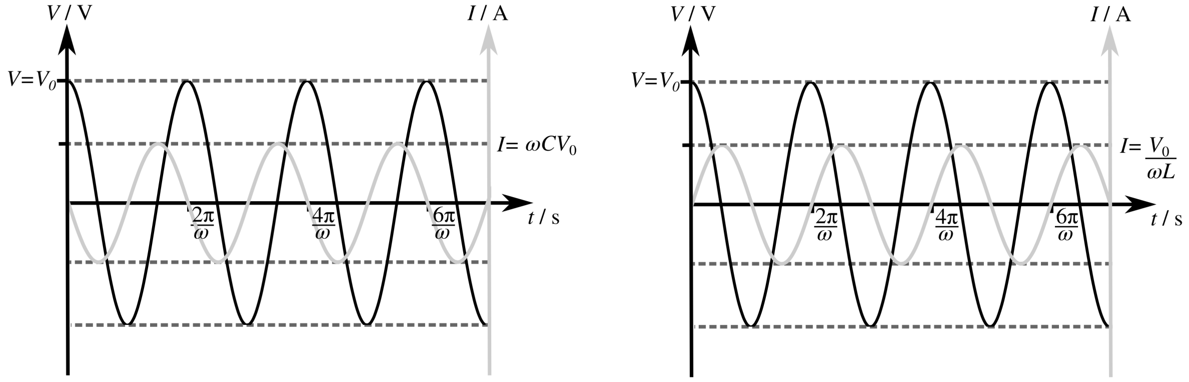


Figure 1.14: Graph of voltage across and current through a capacitor (on the left) and an inductor (on the right) over time. Voltage is the black line and current is the light grey line.

1.3.2 Input and output impedance

When connecting one circuit component to another, the behaviour of a component often changes compared to when it operates in isolation. To accurately predict how the circuit will function, it is necessary to consider the input and output impedances (1.15) of the various components involved.

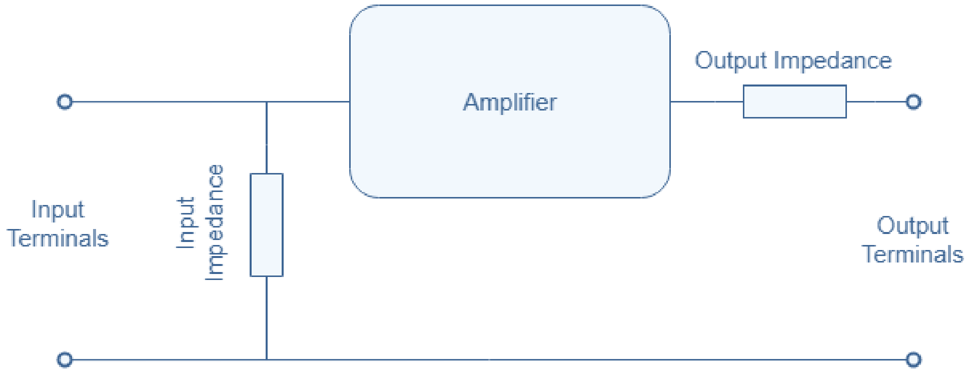


Figure 1.15: Definition of the input and output impedances

The **input impedance** of a circuit represents the opposition to current flow as seen from the perspective of a source driving the circuit. It is a combination of resistance (static opposition) and reactance (dynamic opposition due to inductance and capacitance). Essentially, input impedance tells us how much the load component resists the current flowing from the electrical source. This impedance is a key factor in determining how efficiently a signal is transferred into the circuit. From a practical standpoint, input impedance is crucial in matching circuits. For instance, if the input impedance of a load is equal to the output impedance of the source, maximum power transfer occurs. However, in many cases, it is desirable for the input impedance to be much higher than the output impedance. This high input impedance ensures that the source does not get loaded down excessively, which would otherwise reduce the voltage available to the load. Mathematically, input impedance is often represented in Thévenin's equivalent circuit, which simplifies complex networks into an ideal voltage source with a series impedance. This allows engineers to predict how the circuit will behave when connected to another component or stage. The input impedance helps determine how the current and voltage will change when different load components are introduced.

The **output impedance** refers to the opposition to current flow that a source presents to a load. Like input impedance, it comprises both resistance and reactance. Output impedance is an important characteristic of any device that supplies power or a signal, as it affects how the source behaves when delivering current to a load. When a source drives a load, the output impedance represents how much the source's voltage drops as the load draws current. No real-world source is perfect; there is always some internal impedance that causes the output voltage to decrease as current demand increases. This drop in voltage is particularly important in applications where maintaining a stable output voltage is critical, such as in power supplies and amplifiers. The output impedance is typically modeled as an ideal voltage source in series with a real impedance, often referred to as *source impedance* or *internal impedance*. This model allows engineers to approximate the behavior of the source under load and helps in designing circuits with proper impedance matching.

When connecting two components or stages of a circuit, the relationship between input and output impedance becomes critical. In most practical applications, the input impedance of the load should be much higher than the output impedance of the source. This arrangement minimizes signal loss and ensures efficient power transfer. For example, in audio amplifiers, an output impedance of around 8Ω is typical to match standard speakers, ensuring that the amplifier delivers the proper amount of power without significant losses. By considering both input and output impedance, engineers can design circuits that work efficiently when connected together. Misalignment between these impedances can lead to poor signal transfer, reduced efficiency, or even circuit instability. The use of Thévenin's theorem and Norton's theorem in circuit analysis provides tools for simplifying and understanding these relationships. In conclusion, understanding input and output impedance is key to ensuring proper circuit behavior, particularly when multiple stages are involved. Input impedance determines how the circuit accepts signals, while output impedance dictates how the source delivers power or signals to the next stage. Properly managing these impedances through techniques like impedance matching is critical to achieving efficient and reliable circuit performance.

1.3.3 Reflection problems

In microwave circuits, ensuring effective signal transmission is essential for maintaining high system performance. One of the primary challenges in achieving this is managing impedance matching between components, particularly between input and output impedances. When there is a mismatch, signal reflections occur, causing issues like power loss, signal distortion, and decreased system efficiency.

At microwave frequencies, even minor impedance mismatches can lead to significant signal reflections, where part of the signal is reflected back to the source instead of being transmitted to the load. This effect is measured using the reflection coefficient, which expresses the ratio of the reflected signal amplitude to the incident signal amplitude:

$$\Gamma = \frac{V^-}{V^+}$$

The greater the mismatch, the higher the reflection coefficient, and the more power is lost due to reflections. In particular, the relaton between the Reflection coefficient on the load and Load Impedance Z_L and characteristic impedance Z_0 is the following:

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

In an ideal case, the impedances are perfectly matched, allowing all the power from the source to reach the load without reflections. However, achieving perfect impedance matching in practical microwave systems is rare due to component variations, frequency dependency, and other design limitations. As a result, addressing reflection issues is critical to maintaining the performance of microwave circuits.

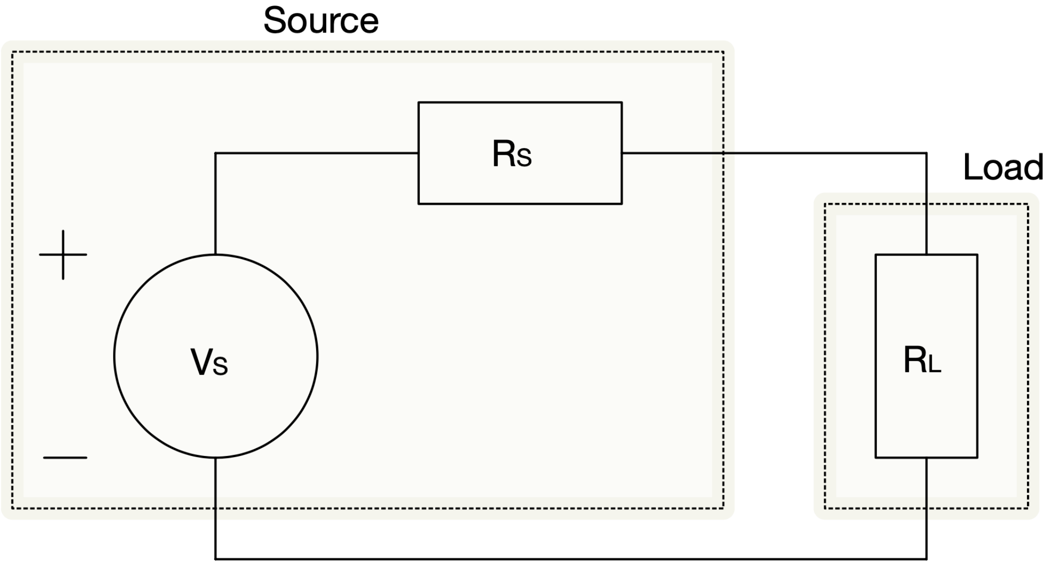


Figure 1.16: Maximum transferred power problem

1.4 Impedance matching

Impedance matching involves the intentional design of source and load impedances to minimize signal reflection or maximize power transfer. To gain a clearer understanding of the concept of power transfer, the following diagram can be analyzed, where R_S is the source resistance and R_L the load resistance: Now, the equation of Power Transfer as function of R_L, R_S and V_S:

$$P = (\frac{V_S}{R_L + R_S})^2 R_L \tag{1.1}$$

From this equation it can be easily deduced that maximum power transfer occurs when $R_L = R_S$. However, maximum transferred power does not imply maximum efficiency. Efficency refers to the percentage of power successfully transferred from the source to the load, while transferred power indicates the maximum amount of power that the load can absorb or utilize. Efficency is given by the following equation:

$$\eta = \frac{1}{1 + \frac{R_S}{R_L}} \tag{1.2}$$

Now, 1.1 and 1.2 can be plotted in *Matlab* in the same graph to better understand their relation: From Figure 1.17 can be clearly denoted that transferred power is maximized when the impedances are matched, that is when $R_L/R_S = 1$. In the case of perfect impedance matching, the efficiency reaches only 50%. An ideal efficiency of 100% is achieved when

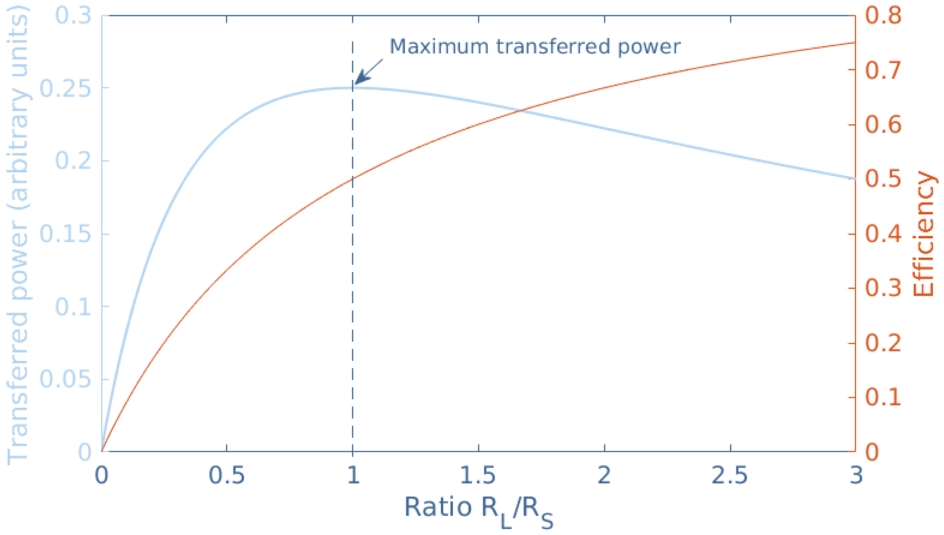


Figure 1.17: Transferred power and efficiency as a function of the ratio R_L/R_S.

the ratio R_L/R_S approaches infinity, which occurs when $R_L \to +\infty$ and $R_S \to 0$ or both. In modern systems, a high input impedance and low output impedance are generally preferred, even if this does not result in an impedance match. As previously mentioned, a key application of impedance matching lies in enhancing the power transfer efficiency from a radio transmitter through the interconnecting transmission line to the antenna. Failure to terminate the transmission line with a matching impedance results in signal reflections, causing destructive interference, characterized by peaks and valleys in voltage along the transmission line, also known as standing waves, which reduce system efficiency and degrade signal quality. Therefore, impedance matching plays a crucial role in achieving a desirable Voltage Standing Wave Ratio (VSWR). The standing wave ratio (SWR) quantifies the effectiveness of impedance matching between a load and a transmission line or waveguide. When there is a mismatch, standing waves are generated along the line. SWR represents the ratio between the highest and lowest voltage points along the line, indicating the degree of mismatch.

SWR can also be defined as the ratio of the maximum to minimum amplitude of the transmission line's current, electric field strength, or magnetic field strength. Assuming negligible transmission line losses, all these ratios are equivalent.

VSWR specifically is an indicator of how effectively radio-frequency power is transferred from a power source, through a transmission line, to a load (such as an antenna). In an ideal setup, where the source, transmission line, and load impedances are perfectly matched, all the energy is transmitted without reflection, and the AC voltage remains consistent along the line.

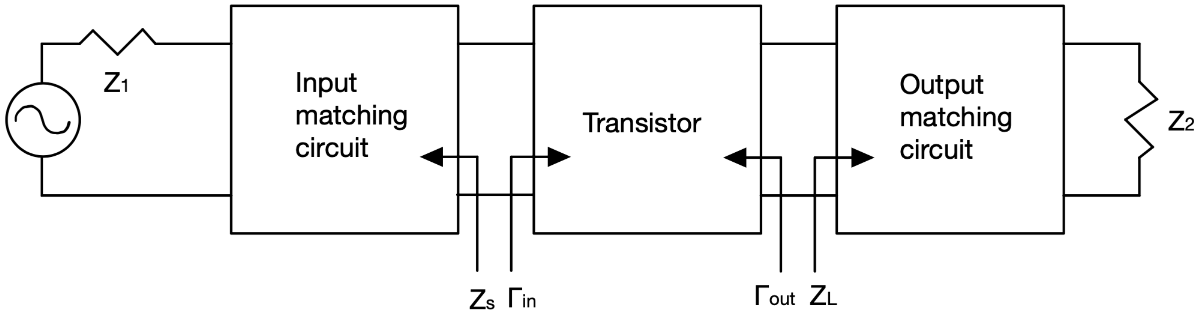


Figure 1.18: Block diagram of a microwave amplifier

However, in real-world systems, impedance mismatches cause part of the power to reflect back toward the source, leading to voltage fluctuations along the line due to destructive interference. VSWR quantifies these variations by measuring the ratio between the highest and lowest voltage points along the line. In an ideal system, this ratio is 1:1, indicating no voltage variation. When reflections occur, the VSWR increases, reflecting the degree of mismatch, for example a VSWR of 1.2:1 indicates some degree of reflection. In particular, to recall the reflection coefficient, here the relation between this last one and VSWR:

$$VSWR = \frac{|V_max|}{|V_min|} = \frac{1+|\gamma|}{1-|\gamma|}$$

Since the magnitude of γ always falls in the range [0,1], the SWR is always greater than or equal to unity.

1.4.1 Typologies

Amplifiers, in order to deliver maximum power to a load or to perform in a certain desired way, must be properly terminated at both the input and the output ports. Figure 1.18 shows a general situation, where the input matching circuit is design to transform the generator impedance Z_1 to the source impedance Z_S, and the output matching circuit transforms the Z_2 termination to the load impedance Z_L. To get this aim, many different types of matching networks can be designed (often with the help of the Smith chart), and they must be lossless, in order not to dissipate any of the signal power. There are three ways to improve an impedance mismatch, all of which are called "impedance matching":

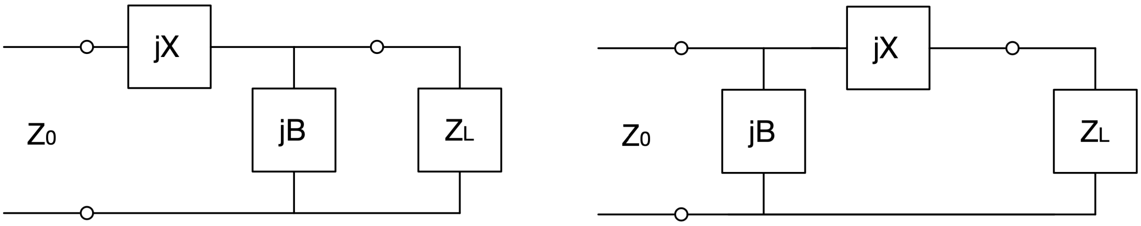
- **Complex conjugate matching**: where the aim of the matching is reaching the condition $Z_L = Z_S^*$; by maximum power theorem this is the only way to extract the maximum power from the source;
- **Complex impedance matching**: described by $Z_L = Z_{line}$, the only way to avoid reflecting echoes back to the transmission line is the reflectionless impedance matching at the end of it;
- **Apparent source resistance matching**: Devices designed with an apparent source resistance approaching zero or, equivalently, an apparent source voltage maximizing efficiency. Deployed at the inception of electrical power lines, this approach is essential for optimizing energy efficiency. Additionally, the utilization of such devices results in the reduction of distortion and minimization of electromagnetic interference. Moreover, they find application in contemporary audio amplifiers and signal-processing devices.

Complex impedance matching

While each of these methods has distinct applications and advantages, complex impedance matching is particularly important in high-frequency systems, such as RF circuits and microwave transmission lines, where minimizing signal reflections is critical to preserve signal integrity. Several practical techniques have been developed to implement complex impedance matching in real-world systems, the most common ones include the quarter wavelength transformer (single section and multisection), stub matching (single, double and triple), and lumped element matching, among others.

- *Quarter wavelength transformer*: A single $\frac{\lambda}{4}$ transmission line is used to match two different impedances. The characteristic impedance of the transformer is defined as $Z_1 = \sqrt{Z_0 Z_L}$. Characterized by narrow bandwidth, it works well only at the design frequency;
- *Multisection quarterwave transformer*: Multiple $\frac{\lambda}{4}$ sections of transmission line are used, each with a different characteristic impedance, to gradually match the load to the source impedance over a broader frequency range; although it has the drawbacks of complexity and space requirements, this method is extremely useful because it enables the synthesis of any desired reflection coefficient response as a function of frequency θ, by properly choosing the $\gamma_n s$ and using enough sections N

- *Single stub*: A short-circuited or open-circuited transmission line (stub) is placed in parallel or series with the main transmission line to cancel the reactive part of the load impedance, achieving impedance matching. Characterized by narrow bandwidth and sensitivity to frequency variations, it requires precise positioning;
- *Double-stab* or *triple-stub tuning*: Two or three stubs are placed at different points along the transmission line to offer more flexibility in matching complex impedances, allowing to match a wider range of impedances with respct to single stub tuning;
- *Matching with lumped elements*: Uses inductors and capacitors (lumped elements) to tune the impedance, compensating for reactive components in the load and matching the source and load impedances. There are two possible configurations for this network: Where, by using Smith chart and carrying out the corresponding calculations separating real and imaginary parts, can be found that for the first configuration:



((a)) Network for z_L inside the $1+jx$ circle of the Smith chart

((b)) Network for z_L outside the $1+jx$ circle of the Smith chart

Figure 1.19: L-section matching networks

$$\begin{cases} B = \frac{X_L \pm \sqrt{\frac{R_L}{Z_0}}\sqrt{R_L^2+X_L^2-Z_0R_L}}{R_L^2+X_L^2} \\ X = \frac{1}{B} + \frac{X_L Z_0}{R_L} - \frac{Z_0}{BR_L} \end{cases}$$

Instead, for the second one:

$$\begin{cases} B = \pm\frac{\sqrt{\frac{Z_0-R_L}{R_L}}}{Z_0} \\ X = \pm\sqrt{R_L(Z_0-R_L)} - X_L \end{cases}$$

1.4.2 Importance of Impedance matching

Impedance matching is essential for high-speed and high-frequency devices, especially in PCB design, to ensure the source and load impedances are properly aligned.

In ultra-high frequency applications, achieving accurate impedance matching is particularly challenging for design engineers, and it's also a difficult task when designing RF and microwave circuits. Even a small mismatch in impedance can cause pulse distortion and signal reflections.
As the frequency increases, the tolerance for error decreases. At higher frequencies, achieving maximum power transfer becomes even more critical. Circuit operate optimally and efficiently when the impedance is perfectly matched. If impedance is not properly matched, signal reflections can cause numerous problems, such as data delays, phase distortion, and a reduced signal-to-noise ratio.

1.4.3 Components

Impedance matching is a critical aspect of circuit design, particularly when efficient energy transfer between components is essential. Various devices and techniques have been developed to adjust and align impedance between sources and loads in both DC and AC circuits. This section explores the primary impedance matching components and the methods used to determine the necessary values to achieve effective matching, including the use of tools such as the Smith chart. n AC circuits, where impedance matching is often more complex due to the frequency-dependent behavior of reactive components, a range of specialized impedance matching devices are employed. Key components for impedance matching include transformers, adjustable networks, and transmission lines:

- *Transformers*: These are ideal for stepping impedance up or down by adjusting the turns ratio between primary and secondary windings, making them particularly useful in power transfer applications. However, transformers are generally limited in high-frequency applications due to core material limitations and frequency response.

- *Adjustable Networks*: These networks consist of lumped components like resistors, capacitors, and inductors arranged in configurations such as L-networks, T-networks, or Pi-networks. By adjusting the values of inductors and capacitors, engineers can fine-tune the circuit impedance, balancing both real and reactive components. This flexibility makes adjustable networks especially effective in AC circuits with variable loads or operating conditions.

- *Transmission Lines*: Transmission lines, including quarter-wave transformers, achieve impedance transformation by leveraging the characteristic impedance of the line itself. By carefully choosing the line length and impedance, these components provide a broadband matching solution that works well at high frequencies, as in RF and microwave applications.

To effectively use these impedance matching components, there are several methods to determine the correct component values.
Among the most prominent methods are **computer simulations**, with softwares like *MATLAB* and *ADS* (Advanced Design System) enabling detailed simulations of impedance matching circuits, **manual computations**, useful for straightforward applications or as a preliminary step before simulations and the **Smith chart** (1.20). In complex applications, these methods are often used in combination. For instance, an initial design might be sketched out on a Smith chart to estimate values, followed by computer simulations to refine the configuration and account for additional parasitic elements or real-world non-idealities.

1.4.4 Electrical applications of Impedance matching

Impedance matching has various applications, for example, in telephone systems and loudspeaker amplifiers, where it ensures optimal performance and signal integrity.
In *telephone systems*, impedance matching helps reducing echo on long-distance lines and allows for proper operation of components like the hybrid coil, which converts between two-wire and four-wire configurations. Typically, a 600Ω impedance is used in local loops, with exchange networks designed to match subscriber lines and minimize side tones.
In *loudspeaker amplifiers* instead, impedance matching is crucial for aligning the amplifier's output to the speaker's impedance. Vacuum tube amplifiers often use impedance-matching transformers to separate AC and DC signals and adjust impedance for efficient power transfer. On the other hand, semiconductor amplifiers with low output impedance generally don't require transformers or capacitors for impedance matching. Overall, impedance matching ensures efficient signal transfer, minimizes distortion, and avoids negative effects like signal reflections or unwanted side tones.

1.5 Smith Chart

The Smith chart (also called Smith diagram, Mizuhashi chart, Mizuhashi–Smith chart, Volpert–Smith chart or Mizuhashi–Volpert–Smith chart), is a visual tool or nomogram created for electrical and electronics engineers who work in radio frequency (RF) engineering. It helps them solving issues

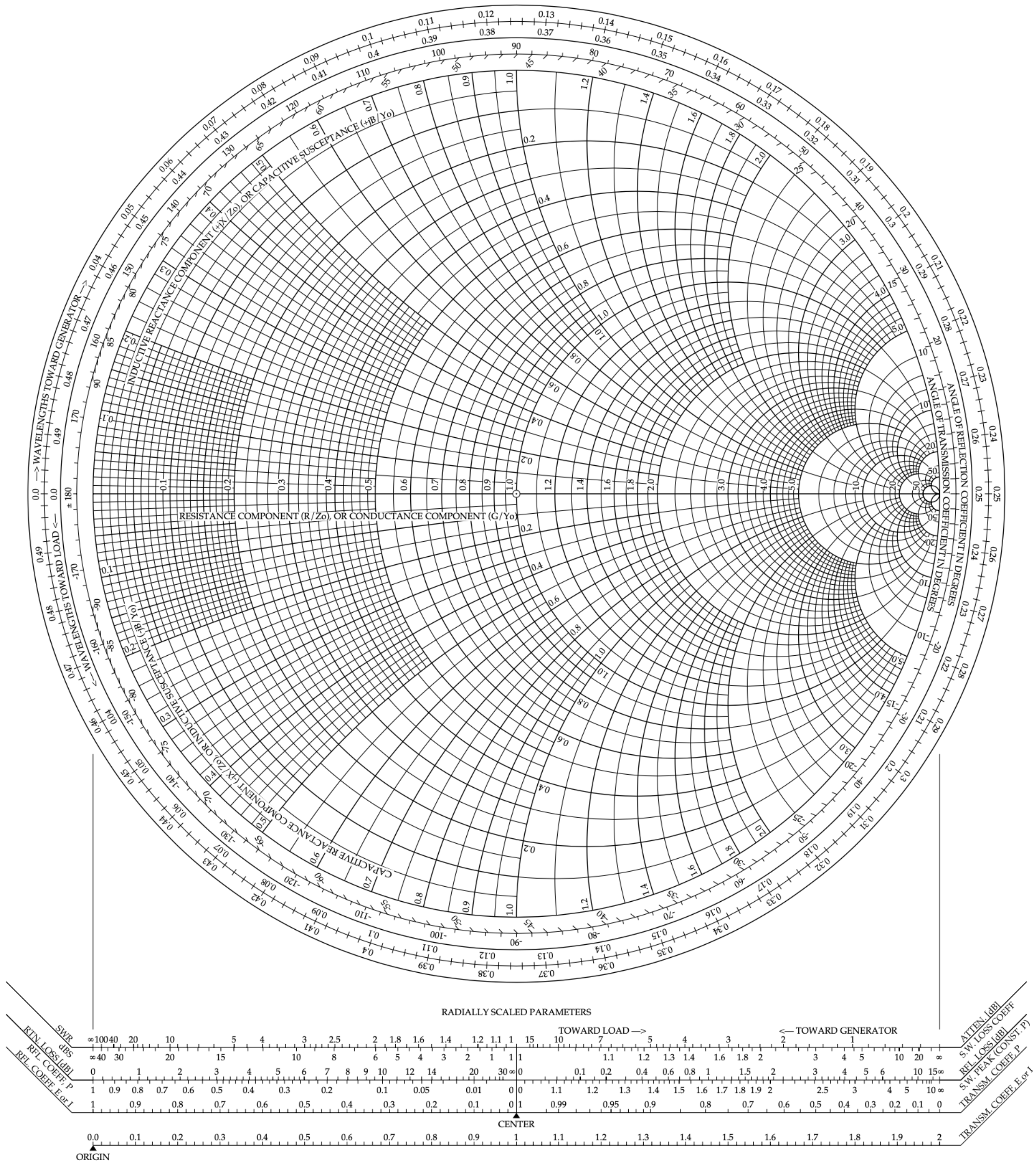

WAVELENGTHS TOWARD GENERATOR —>
<— WAVELENGTHS TOWARD LOAD <—
INDUCTIVE REACTANCE COMPONENT (+jX/Zo), OR CAPACITIVE SUSCEPTANCE (+jB/Yo)
CAPACITIVE REACTANCE COMPONENT (-jX/Zo), OR INDUCTIVE SUSCEPTANCE (-jB/Yo)
RESISTANCE COMPONENT (R/Zo), OR CONDUCTANCE COMPONENT (G/Yo)
ANGLE OF REFLECTION COEFFICIENT IN DEGREES
ANGLE OF TRANSMISSION COEFFICIENT IN DEGREES
RADIALLY SCALED PARAMETERS
TOWARD LOAD —>
<— TOWARD GENERATOR
SWR
dBS
RTN. LOSS [dB]
RFL. COEFF, P
RFL. COEFF, E or I
ATTEN. [dB]
S.W. LOSS COEFF
RFL. LOSS [dB]
S.W. PEAK (CONST. P)
TRANSM. COEFF, P
TRANSM. COEFF, E or I
CENTER
ORIGIN

Figure 1.20: Smith Chart

related to transmission lines and matching circuits. From the beginning of World War II until the development of digital computers for engineering problems, the Smith Chart was the dominant tool for microwave engineers. The first, simplified version of a Smith Chart shows both resistance in ohms (numbers on the horizontal axis that range from 0.2 to 10) and the angle of a quantity called the reflection coefficient, in degrees on the outer edge of the circle (1.21. Smith initially developed his chart to address issues he faced when transmitting radio waves through a special cable known as a transmission line, which carries the waves from a radio transmitter to an antenna. Ideally, the waves would travel seamlessly in one direction without any reflections. However, with certain types of

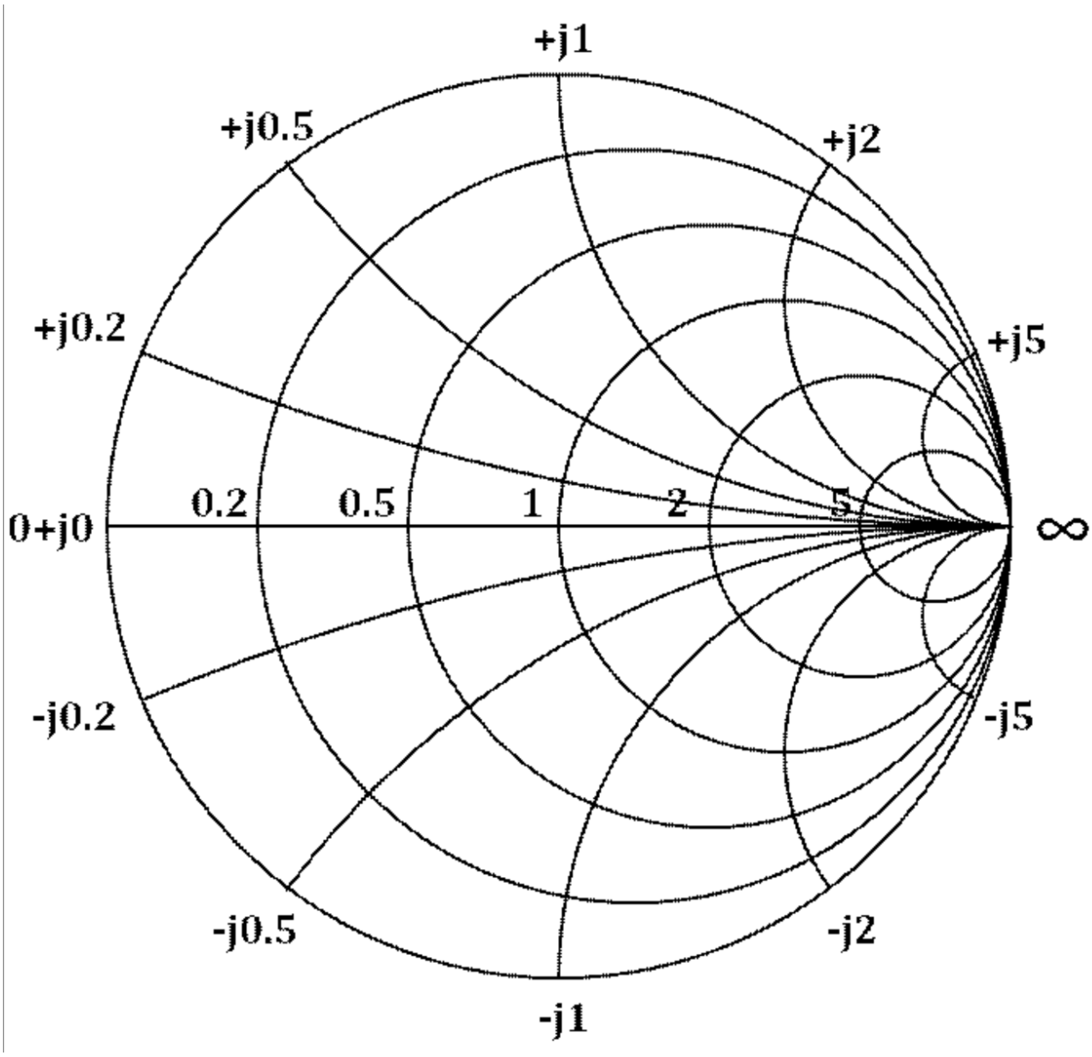


Figure 1.21: Semplified Smith Chart

antennas, some waves are reflected back into the line. These reflected waves can bounce along the line, leading to power loss and reducing the overall transmission efficiency. In 1936, Smith came up with the idea of using a circular chart and mathematically transforming the reflections so that, regardless of their size, all the values would fit within the chart's circular boundary (1.20). This innovation, along with other features, made it much easier to solve complex microwave engineering problems in circuits and transmission lines graphically, without extensive math. In the era before digital computers, any method that helped engineers avoid lengthy calculations was highly valuable, so Smith's chart quickly gained popularity among radio and microwave engineers after it was published in Electronics magazine in January 1939. While traditional methods, such as the Smith chart, provide valuable tools for impedance matching, they often require manual adjustments and can be time-consuming, especially in systems where impedance conditions change dynamically or where high precision is essential. In such cases, these conventional approaches may struggle to meet the demands of rapid, adaptive tuning.

This challenge opens the door to innovative solutions that can automate and optimize the impedance matching process. The primary aim of this thesis is to explore the potential of Machine Learning, specifically Reinforcement Learning, to address this need. With the help of RL, it becomes possible to develop an intelligent system capable of continuously adjusting impedance components in real time, effectively adapting to changing conditions and minimizing mismatch without human intervention. This approach not only enhances efficiency but also represents a significant advancement in combining classical engineering principles with modern computational techniques.

In the following sections, this thesis will detail the design, implementation, and testing of a reinforcement learning model specifically setted for the impedance matching problem. This exploration aims to provide insight into the potential of RL as a robust and adaptive solution for achieving optimal impedance matching in complex environments.

Chapter 2

Machine Learning

2.1 Machine Learning

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and statistical algorithms to learn from data and generalize to unseen data, and thus perform tasks without explicit instructions, gradually improving its accuracy. They use historical data as input to make predictions, classify information, cluster data points, reduce dimensionality and even help generate new content. Machine learning methodologies have found application across diverse domains, encompassing large language models, computer vision, speech recognition, email filtering, agriculture, and medicine. This is particularly evident in situations where the development of specialized algorithms for essential tasks proves to be prohibitively expensive.

2.2 Approaches

Traditional machine learning approaches are commonly categorized into three broad groups (2.1), aligning with distinct learning paradigms. This categorization is based on the nature of the available ’signal’ or ’feedback’ accessible to the learning system:

- **Supervised learning**: The computer is provided with input examples along with their corresponding desired outputs, supplied by a ’teacher.’ The objective is to acquire a generalized rule that effectively maps inputs to outputs;
- **Unsupervised learning**: The learning algorithm operates without the provision of explicit labels, requiring it to independently discern inherent structures within the input data. Unsupervised learning can serve as a primary objective, involving the discovery of concealed patterns in data, or it can function as a means to achieve a specific

outcome;

- **Reinforcement learning**: A computer program engages with a dynamic environment with the objective of achieving a specific goal, such as driving a vehicle or playing a game against an opponent. As the program navigates through its problem space, it receives feedback in the form of rewards, akin to which it strives to maximize. In particular this approah will be the most studied in this project, as the most suitable to this problem.

Other approaches have been devised that do not neatly conform to this three-fold categorization, and in some cases, a single machine learning system may employ more than one approach. Examples include topic modeling and meta-learning.

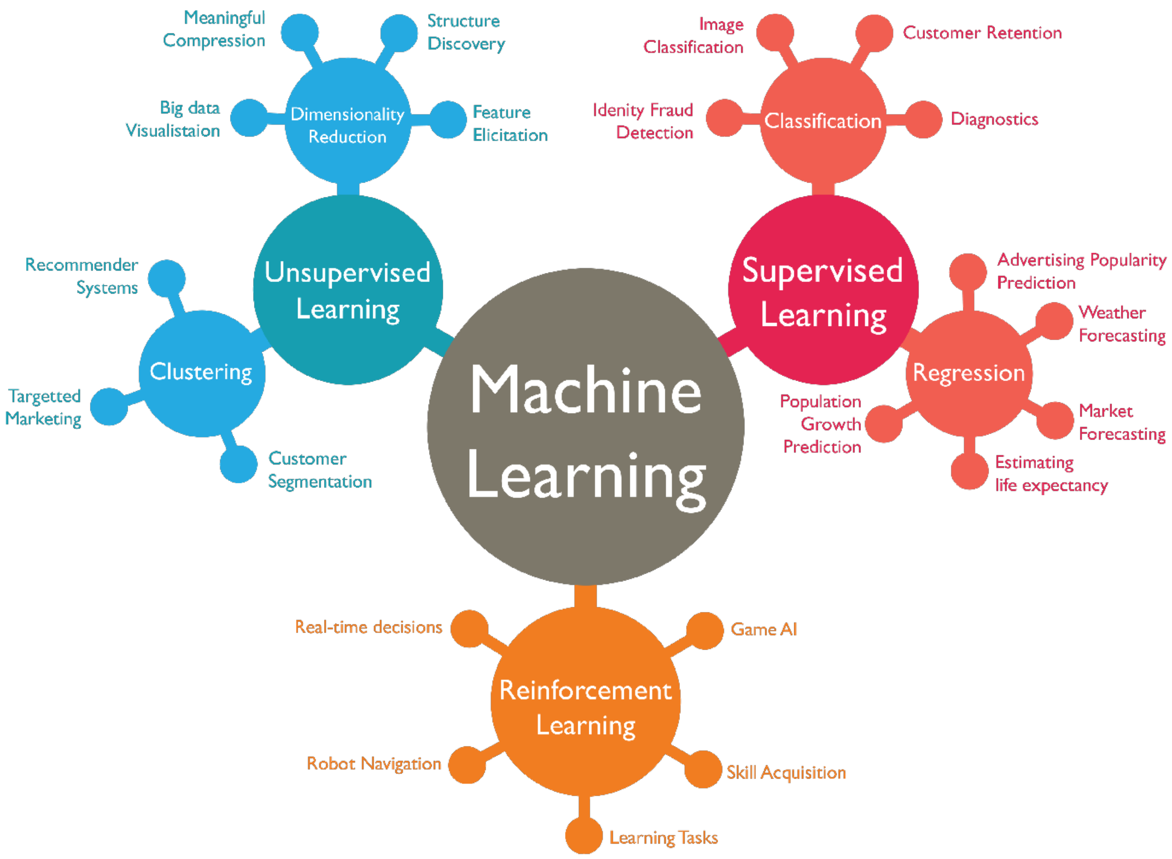


Figure 2.1: Machine learning approaches

Image credits: *https://towardsdatascience.com/machine-learning-types-2-c1291d4f04b1*

The most common Machine learning algorithms are:

- *Neural networks*: neural networks simulate how the human brain works, using a large number of interconnected processing nodes. They can recognize patterns and play a key role in applications like natural language translation, image recognition, speech recognition, and image generation.

- *Linear regression*: this algorithm is used to predict numerical values based on a linear relationship between different variables. For instance, it can be used to predict house prices based on historical data from the area.

- *Logistic regression*: this supervised learning algorithm makes predictions for categorical response variables, such as ”yes/no” answers. It can be used for applications like spam classification and quality control on a production line.

- *Clustering*: using unsupervised learning, clustering algorithms can identify patterns in data to group them together. Computers can help data scientists by identifying differences among data points that might be overlooked by humans.

- *Decision trees*: decision trees can be used both to predict numerical values (regression) and to classify data into categories. They use a branching sequence of linked decisions, often represented with a tree diagram. One advantage of decision trees is that they are easy to validate and verify, unlike the black-box nature of neural networks.

- *Random forests*: in a random forest, the machine learning algorithm predicts a value or category by combining results from a set of decision trees.

2.2.1 Disadvantages of ML

Machine learning has significant limitations, which can sometimes surpass the scale of human errors.
First, **data quality** is essential; without a credible source, machine learning outcomes can be unreliable. High-quality data is crucial, but waiting for it can delay results, emphasizing how dependent machine learning is on data accuracy.
Time and resources are also significant factors. Machine learning algorithms require substantial processing power and infrastructure to handle large, diverse datasets and must undergo extensive trial runs to achieve reliability. These trial phases are both time-intensive and costly, demanding specialized resources and expertise.

Interpretation of results presents another challenge. Although machine learning can analyze data effectively, achieving 100% accuracy is rarely possible. Algorithms need to be continually refined to improve reliability, as even minor inaccuracies can lead to incorrect insights.

Machine learning processes often introduce a **high chance of error**, especially during initial phases, where mistakes, if not addressed, can create severe problems. Errors typically stem from issues with data quality or algorithm design, meaning that precision in both is essential to prevent significant impacts on output.

On a societal level, machine learning has led to both beneficial and adverse changes. It is reshaping job roles, often **eliminating human interface** in favor of automation, which, while efficient, reduces employment opportunities and transforms the job market. Those without technical expertise may struggle to adapt to the rapidly evolving workforce demands.

The **high costs** of machine learning infrastructure also mean it is accessible primarily to large organizations and government bodies, restricting widespread use and benefits for smaller entities or individuals.

Privacy concerns further complicate machine learning's adoption. Data collection practices have sparked debates over user consent and confidentiality, with cases of unauthorized data usage by corporations highlighting privacy as a contentious issue, especially as data is a core element of machine learning.

Lastly, machine learning is a field still in development, with ongoing research needed to drive real innovation. While advancements have been made, significant **research and innovatio**n are still required to fully realize its potential and make transformative impacts across industries.

2.3 Reinforcement Learning

The Reinforcement Learning problem centers around an agent navigating an unknown environment to accomplish a goal (2.2). RL operates on the premise that all goals can be characterized by the maximization of expected cumulative reward. The agent is tasked with acquiring the ability to perceive and influence the state of the environment through its actions, with the ultimate aim of optimizing reward outcomes.

Reinforcement Learning coinsists of six main elements:

- The *agent*;
- The *State*: current situation of the agent;
- The *Environment* the agent interacts with;

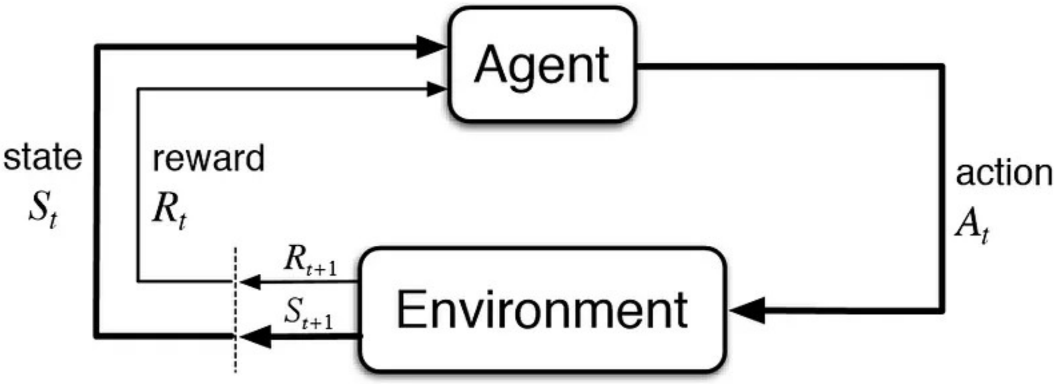


Figure 2.2: Reinforcement Learning scheme

- The *policy* that the agent follows to take actions, it is a mapping from perceived states of the environment to actions to be taken when in those states.;
- The *reward* signal that the agent observes upon taking actions. A reward function is a function that provides a numerical score based on the state of the environment;
- The *value*: the future reward that an agent would receive by taking an action in a particular state.

2.3.1 Balancing Exploration and Exploitation in RL

Since a reinforcement learning agent has no manually labeled input data to guide its actions, it must explore its environment by attempting different actions to discover those that receive rewards. These reward signals enable the agent to learn to prefer actions that have resulted in positive outcomes, aiming to maximize its cumulative reward. However, the agent also needs to keep exploring new states and actions to build experience that will further enhance its decision-making.

Thus, RL algorithms require the agent to balance exploiting its knowledge of rewarding actions with exploring other possible actions and states. The agent cannot focus only on exploration or exploitation; it must continually try new actions while also favoring single actions or sequences of actions that lead to the highest cumulative rewards.

2.3.2 Subdivision

Within the expansive realm of reinforcement learning, two important paradigms surface: *Model-Based Reinforcement Learnin*g and *Model-Free Reinforcement Learning.*

The first one involves the use of a learned or known model of the environment to make decisions and plan actions. Model-Based RL relies on the agent's internal representation of how the environment behaves. This internal model is then used for planning and decision-making.

Instead Model-Free Reinforcement Learning is a category of reinforcement learning where an agent learns to make decisions by directly interacting with an environment, without requiring knowledge of the underlying dynamics or rules governing the environment. So, the agent learns from experience rather than relying on a predefined model of the environment. Model-Free RL is well-suited for situations where the system's dynamics are complex, unknown, or hard to model accurately. Another distincion can be made between *Online* and *Offline* learning (2.3):

- *Online*: the agent gathers data by directly interacting with its environment. This data is processed and accumulated continuously as the agent continues to interact;
- *Offline*: the agent learns from pre-recorded data about the environment rather than through direct interaction, this is because it doeans not have direct access to the environment. Offline learning is commonly used in research, especially when direct interaction with the environment is impractical or challenging for model training.

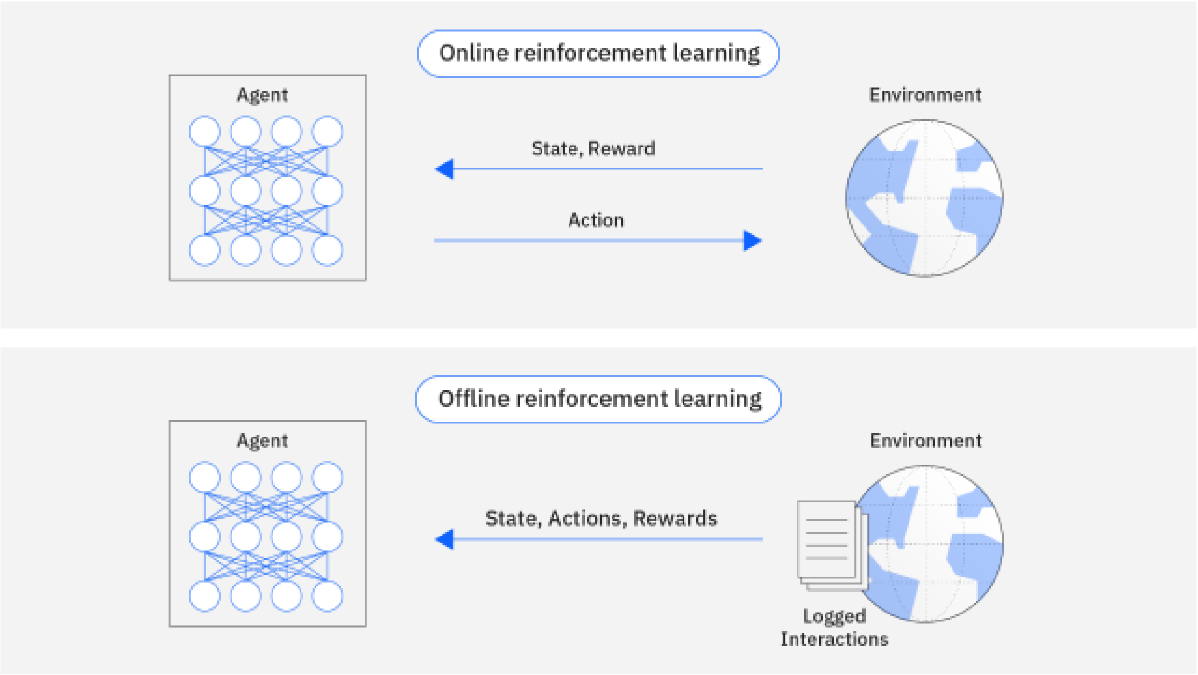


Figure 2.3: Online VS Offline learning

2.3.3 Applications

Reinforcement learning can find various practical applications, such as:

- Robotics for industrial automation: robots lack the ability to apply common sense in making moral and social decisions. This is where the combination of Deep Learning and Reinforcement Learning, known as Deep Reinforcement Learning, plays a crucial role. This approach allows robots to adopt a "Learn How to Learn" model, empowering them to refine their decisions by effectively recognizing and interacting with various objects they perceive, and tackle complex tasks, some of which are difficult for humans, as they learn what and how to learn from different abstractions within the datasets available to them;
- Machine learning and data processing;
- In algorithms to create training systems that provide custom instruction and materials according to the requirement of students.

2.3.4 Advantages and disadvantages of Reinforcement Learning

Reinforcement learning is one of the most used tecniques of Machine Learning, mostly due to it capability of correct the errors that occurr during the processes, with a drastic reduction of probability of encountering again the same error. In particular Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques, for example Robotic Control for Manipulation: training robots for fine-grained manipulation tasks, such as picking up objects with varying shapes and sizes, can be highly complex. RL allows robots to learn dexterous and adaptive control policies, enabling them to handle diverse and unpredictable scenarios. One potential drawback worth noting is that excessive use of reinforcement learning, particularly in straightforward problems, may result in an overwhelming number of states. This proliferation of states can potentially compromise the effectiveness of the learning process as it demands a substantial amount of data and computational resources. In essence, RL can be data-hungry, and its performance may be hindered when applied extensively in scenarios where the complexity does not align with the need for a large state space.
An example of this last speech could be the calculation of a simple mathematical function: in this case, the problem is highly structured, the rules are well-defined and known, and the solution can be easily obtained through conventional mathematical operations. In cases like this, employing Reinforcement Learning might be considered an overly complex and unnecessary

approach, while a conventional solution is clearer, more efficient, and easily implementable.

2.3.5 Why Reinforcement Learning for impedance matching

Reinforcement Learning offers a great approach in solving the impedance matching problem, addressing challenges that traditional methods struggle to overcome. Impedance matching, particularly in AC circuits, involves optimizing the interaction between input and output impedances to minimize signal reflection and maximize power transfer. While established techniques, such as the Smith chart and manual computations, provide effective solutions, they are often labor-intensive, require expert intervention, and may fall short in dynamic or nonlinear environments.

One of the primary advantages of RL in this context is its ability to **automate complex adjustments**. Traditional methods require iterative calculations and manual tuning of components, such as inductors and capacitors, to achieve optimal matching. RL, in contrast, automates this process, enabling real-time optimization without human intervention. This capability is particularly useful in systems where impedance conditions are dynamic—varying due to factors like frequency shifts, load changes, or environmental fluctuations. By learning adaptive policies, RL can continuously fine-tune the impedance matching process, making it ideal for real-time applications in wireless communications, radar systems, and microwave circuits.

Moreover, RL excels in **nonlinear and multi-objective scenarios**, which are common in high-frequency applications. Impedance matching often involves frequency-dependent behaviors, parasitic effects, and nonlinearities, making analytical solutions difficult. Reinforcement Learning's ability to explore and optimize in multi-dimensional spaces allows it to identify effective solutions that traditional methods maybe ar not able to detect. Through trial-and-error exploration, RL can uncover unconventional configurations of components, leading to potentially superior performance.

Another key strength of RL is its **reward-based optimization framework**, which aligns naturally with the goals of impedance matching. By defining the reward function to prioritize objectives such as minimizing reflection coefficients, maximizing power transfer, or, as in the case of this thesis, penalizing inefficient actions, RL can focus on achieving specific performance targets.

Finally, RL offers **robustness** to uncertainty. Real-world systems often face challenges like component tolerances, manufacturing imperfections, and environmental variability. RL's learning process enables it to account for these uncertainties, resulting in solutions that are more reliable and adaptable to practical conditions. This robustness is crucial in applications where precise matching is required across a range of operating scenarios.

In conclusion, the adoption of Reinforcement Learning for impedance matching represents a significant advancement in the field of microwave and RF engineering. By combining automation, adaptability, and optimization, RL addresses the limitations of traditional methods while opening new pathways for innovation. This thesis explores the implementation of RL for impedance matching, aiming to demonstrate its potential to redefine this critical aspect of circuit design.

2.3.6 Neural Networks

Another fundamental element for this thesis is neural networks, which will be used to support the reinforcement learning approach and effectively address the impedance matching problem.

A neural network is an artificial intelligence approach that enables computers to process data in a way inspired by human brain's functioning. Neural networks are sometimes called artificial neural networks (ANNs) or simulated neural networks (SNNs). They are a subset of machine learning, and at the heart of deep learning. This technique is a branch of machine learning where interconnected nodes, or neurons, are arranged in a layered structure that resembles the human brain. Through this layered network, computers develop an adaptive system that learns from errors, continuously improving over time. As a result, artificial neural networks aim to tackle complex tasks, such as document summarization or facial recognition, with high precision. Neural networks come in various types, each designed for specific applications, the most common ones are:

- *The perceptron*: the oldest neural network, created by Frank Rosenblatt in 1958, it consists of a single layer, working by receiving numerical inputs along with associated weights and a bias. Each input is multiplied by its respective weight (forming what's called a weighted sum), and these products are then added together with the bias. This combined value is then passed through an activation function, which processes it to produce the final output;

- *Feedforward neural networks, or multi-layer perceptrons (MLPs)*: they consist of an input layer, one or more hidden layers, and an output layer. Although they are often called MLPs (Multi-Layer Perceptrons), they actually use sigmoid neurons rather than perceptrons because real-world problems are typically nonlinear. They are trained by feeding data through them and serve as the basis for applications like computer vision, natural language processing, and other types of neural networks;

- *Convolutional neural networks (CNNs)*:like feedforward networks, these types are typically used for tasks like image recognition, pattern detection, and computer vision. They rely on linear algebra principles, especially matrix multiplication, to detect patterns within images;
- *Recurrent neural networks (RNNs)*: characterized by their feedback loops. They are mainly used with time-series data to predict future results, such as forecasting stock market trends or sales figures.

Specifically, the LSTM (long short-term memory) falls within this last group of neural networks and the Pytorch one will be utilized in this project to support the reinforcement learning implementation for impedance matching.

LSTM: working principles

Pytorch LSTM networks are designed to work with sequential data by capturing dependencies across time steps, which makes them suitable for handling temporal patterns. The main parameters are:

- *input_size*: defines the number of features in each input sample.
- *hidden_size*: specifies the number of units in the LSTM's hidden layer, determining the output feature size.
- *num_layers*: indicates the number of LSTM layers. For instance, setting `num_layers=2` stacks two LSTM layers, with the second layer receiving outputs from the first. *Default: 1*
- *bias*: if set to `False`, the LSTM will operate without bias weights (b_{ih} and b_{hh}). *Default: True*
- *batch_first*: when `True`, the model expects input and output tensors in the format `(batch, seq, feature)` rather than `(seq, batch, feature)`. This does not affect the hidden and cell states format. *Default: False*
- *dropout*: if greater than zero, applies dropout on the outputs of each LSTM layer, except for the last layer, to prevent overfitting. The dropout probability is set by `dropout`. *Default: 0*
- *bidirectional*: when `True`, creates a bidirectional LSTM that processes input sequences in both forward and reverse directions. *Default: False*

- *proj_size*: if greater than zero, applies a projection layer that reduces the size of the LSTM output to the specified size. *Default: 0*

2.4 Maze problem

A maze, in the realm of informatics and computer science, is a structured and often intricate arrangement of paths or corridors, typically designed as a puzzle or navigational challenge (2.4(a)). Mazes are commonly used to test and show the efficiency of algorithms in solving spatial problems. A maze consists of interconnected passages, often forming a network of junctions, dead ends, and open paths. The goal in a maze can vary; it might involve finding the shortest path from a starting point to an exit, locating a specific destination, or navigating through the maze while avoiding obstacles.

This section explores the maze problem as an example use of reinforcement learning. A maze represents a structured environment with defined goals and constraints, serving as an ideal framework for demonstrating RL's efficiency in navigating complex scenarios. The application of RL in solving the maze problem offers insights into spatial decision-making and path optimization, which are analogous to the challenges faced in impedance matching tasks.

In this thesis, the maze problem serves not only as a pure example but also as a foundational step for developing and adapting RL techniques. The MATLAB implementation of RL for the maze problem showcases critical features like dynamic exploration and exploitation, which are directly applicable to the adaptive tuning of input and output impedances.

By leveraging methodologies like Q-learning and SARSA, the maze problem provides a controlled setting to refine RL strategies. These strategies are subsequently adapted to address the nonlinearities and multi-dimensional complexities of impedance matching, enabling the automation and optimization of this crucial process in microwave field.

2.4.1 RL for the Maze Problem

Q-learning and *SARSA*, two prominent model-free reinforcement learning algorithms, are examined in this section as they are central to the methodologies employed in this thesis. Both algorithms were adapted and implemented as part of an exploratory approach to address the impedance matching problem, leveraging their distinct strategies in action exploration to enable dynamic, adaptive impedance tuning.

Q-learning and SARSA differ in terms of their exploration strategies while their exploitation strategies are similar.

Q-learning aims to discover an optimal policy by maximizing the expected cumulative reward across all subsequent steps, originating from the current state.

In simpler terms, it creates a mapping between states and actions, where actions are determined based on observed states. This mapping is stored in a dedicated table known as the Q-table, with actions along the x-axis and states along the y-axis, and in each cell is stored the respective reward. (2.4(b)).

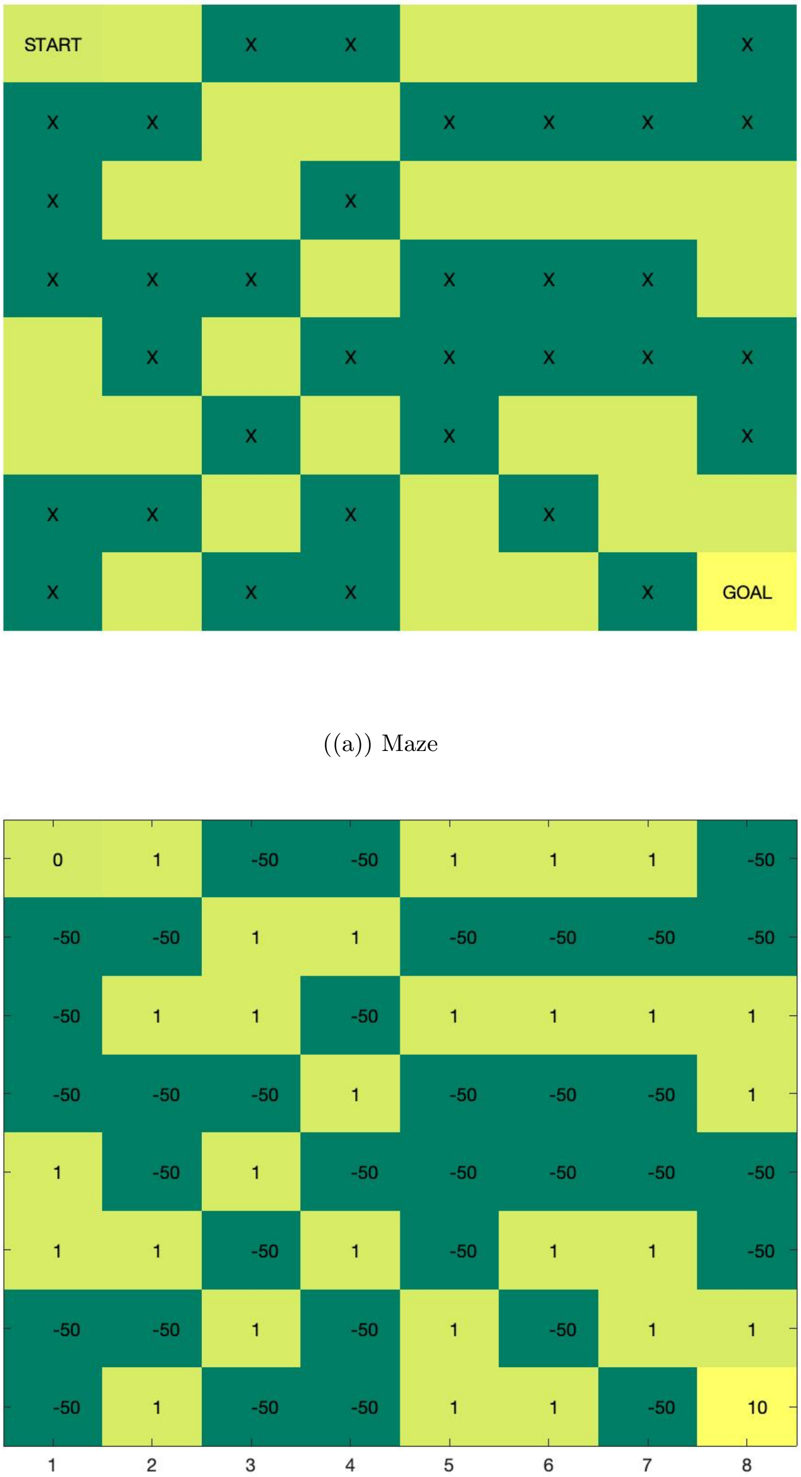
The State–Action–Reward–State–Action (SARSA) algorithm is a method for learning a policy in a Markov decision process, commonly employed in the field of reinforcement learning within machine learning. Originally introduced by Rummery and Niranjan in a technical note under the name ”Modified Connectionist Q-Learning” (MCQ-L), the alternative term SARSA, suggested by Rich Sutton, was only mentioned in a footnote. SARSA is an algorithm where, at the current state (S), the agent executes an action (A), receives a reward (R), and ends up in the next state (S1), and takes another action (A1). Consequently, the acronym SARSA represents the tuple (S, A, R, S1, A1). Termed as an on-policy algorithm, SARSA updates the policy based on the actions it actually takes. In contrast to Q-learning, SARSA considers the action (A1) performed in the next state (S1) when updating the Q-value. Q-learning, on the other hand, utilizes the action with the highest Q-value in the subsequent state (S1) for Q-table updates.

A key distinction between SARSA and Q-learning lies in their learning strategies. SARSA operates as an on-policy learning algorithm, while Q-learning functions as an off-policy learning algorithm. This difference is visible in the difference of the update statements for each technique:

- Q-Learning: $Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \gamma_{max_a} Q(s_{t+1}, a) - Q(s_t, a_t)$
- SARSA: $Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$

In particular, SARSA[1] is the algorithm exploited in this Maze problem, considering it the most suitable for the original purpose of this project, the LC impedance matching problem.

[1] Credits: *Bhartendu (2024). SARSA Reinforcement Learning (`https://www.mathworks.com/matlabcentral/fileexchange/63089-sarsa-reinforcement-learning`), MATLAB Central File Exchange. Retrieved February 26, 2024.*



START		X	X				X
X	X			X	X	X	X
X			X				
X	X	X		X	X	X	
	X		X	X	X	X	X
		X		X			X
X	X		X		X		
X		X	X			X	GOAL

((a)) Maze

	1	2	3	4	5	6	7	8
1	0	1	-50	-50	1	1	1	-50
2	-50	-50	1	1	-50	-50	-50	-50
3	-50	1	1	-50	1	1	1	1
4	-50	-50	-50	1	-50	-50	-50	1
5	1	-50	1	-50	-50	-50	-50	-50
6	1	1	-50	1	-50	1	1	-50
7	-50	-50	1	-50	1	-50	1	1
8	-50	1	-50	-50	1	1	-50	10

((b)) Q-table

Figure 2.2: 8x8 SARSA Maze map (a) and the respective Q-table (b) where each cell corresponds to the reward of that set of action-state

Chapter 3

Implementation of Reinforcement Learning

The objective of this chapter is to evaluate the suitability of the SARSA algorithm for addressing the problem of implementing machine learning (specifically, reinforcement learning) in impedance matching. Impedance matching, especially within the context of continuous states and actions, presents a unique challenge for traditional reinforcement learning algorithms. SARSA, which is often well-suited for discrete environments, requires adaptation to meet the complexities of this application.
Therefore, a two-step approach is proposed:

- **Discretization Analysis**: First, the problem is simplified by breaking it into distinct, manageable parts. This discretization allows SARSA to be applied more straightforwardly, leveraging its strengths in handling problems with discrete states and actions.
- **Continuous Analysis**: Building on insights from the discretized approach, the next step explores how SARSA can be adapted for the original problem, which involves continuous states and actions. This phase focuses on refining the algorithm to align better with the continuous characteristics of impedance matching.

In summary, this chapter adopts a step-by-step approach. Initially, the problem is simplified through discretization, allowing for practical experimentation and the acquisition of foundational insights. Subsequently, these insights guide the adaptation of SARSA to accommodate the continuous nature of the original problem. This approach is visually represented in the diagram in Figure 3.1.
The loop begins with the agent receiving the state from the environment, which, in the context of impedance matching, might include parameters such as the input and output impedances. Based on this observed state, the

Actor component of the agent selects an action, that in this case could be adjusting the value of an inductor or capacitor or altering a configuration to improve impedance matching, or adding/remove a component.
After executing the selected action, the Environment updates its internal state to reflect the changes caused by the agent's decision. For instance, in an impedance matching system, this update could involve recalculating the internal resulting impedance. The environment then evaluates the effectiveness of the agent's action by generating a reward, a numerical value that indicates how well the action achieved the desired outcome. In this case, the reward might be higher if the impedance mismatch is reduced, and lower if the mismatch increases (or, with a negative reward, the contrary). The Critic component of the agent then evaluates this reward to assess the quality of the state-action pair, effectively learning how beneficial the action was for achieving the overall objective. This evaluation is used to update the agent's internal policy, which governs how the Actor selects actions based on the observed state. By refining its policy iteratively, the agent improves its decision-making over time.
This loop continues as the updated policy is applied to select the next action, and the cycle repeats. With each iteration, the agent learns from its interactions with the environment, gradually improving its ability to adapt to changes and maximize rewards. In the context of impedance matching, this iterative process allows the RL agent to dynamically and intelligently adjust components to optimize matching, even in complex or varying conditions. The loop is repeated until a stopping criterion is reached, such as achieving a specified level of performance or completing a set number of iterations, resulting in a highly capable system for automated and adaptive impedance matching.

3.1 Discrete step

In this initial step, to achieve problem discretization, it is hypothesized that a circuit will be designed with only five components, where each component (L and C) are avaiable in ony descrete values, in particular from among the following 13 predefined values, represented in 3.1.

L (nH)	0	10	12	15	18	22	27	33	39	47	56	68	82
C (pF)	0	10	12	15	18	22	27	33	39	47	56	68	82

Table 3.1: Possible values of L and C

Where the 0 values represent the possibility of not adding components at all.

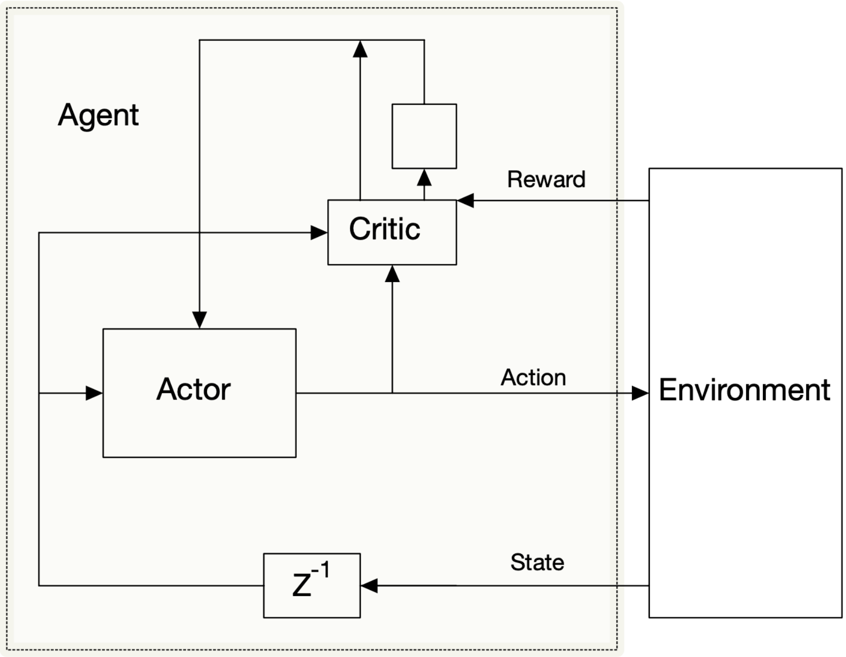


Figure 3.1: Diagram of the Reinforcement learning problem

It has been chosen to represent each state using 5 sets of three numbers, amounting to a total of 15 numbers. Within each set, there are three consecutive significant numbers:

- First number: [0,1], identify if the component is an L (0) or a C (1);
- Second number: [0,1], identify if the component is in parallel (0) or in series (1);
- Third number: [0,13], identify the index of the array corresponding to the values of L and C.

3.1.1 Formulation of the state update

An important part of the algorithm to consider is the formulation of the state update, the process of transitioning from one state to another based on the agent's actions and the environment's response. In particular, can be used (as the SARSA algorithm) the Bellman equation mentioned above, defined as:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$$

Where:

- α is the learning rate, varying between 0 and 1;
- γ is the discount factor, still in the range [0,1];
- $Q(s_t, a_t)$ is the current Q-value;

- r_{t+1} is the immediate reward.

In particular, to summarize can be defined the temporal difference error δ as:

$$\delta = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$$

Obtaining:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha * \delta$$

3.1.2 Formulation of the reward

The second step focuses on formulating the reward function, which provides a numerical score based on the state of the environment.
To achieve this, the impedance is first computed before the design of the five-component circuit.
Following this, the difference between this impedance and the conjugate of the input impedance is evaluated.
Additionally, a penalty ϕ is applied for each component used, aiming to minimize the number of components utilized:

$$\phi = k * NumComp$$

Where k is a penalty constant (initially setted to 0.1). Since the aim is to maximize the reward, finally this one is defined n this way:

$$reward = -abs(Zf - conj(Zs)) - \phi$$

Since the work is based on a discretization of the problem, surely the algotithm will not be able to get an exact match, so a tolerance is defined in advance (10^{-12} in this example).
Therefore this is the condition to break the algorithm in case if goal reachment:

$$all(Zf - conj(Zs) < tolerance)$$

3.1.3 Conclusions

As expected, the problem's discretization cannot be execute due to its high complexity. For instance, when considering 5 components, each with 13 potential values and 2 possible types (inductor or capacitor), along with 2 potential configurations (series or parallel), the outcome is:

$$\texttt{numStates} = (2 * 2 * 13)^5 = 380204032$$

This represent the number of possible states and actions (`numActions` = `numStates` − 1, since the actual state before the action has been exclude), and this would require an array of 22.7GB to work on *Matlab*, exceeding the maximum array size preference.

Given the high complexity of the problem, implementing the solution in *Matlab* proved too challenging. Consequently, after consultation with Professor José Carlos Pedro and following the recommendation of a researcher from the University of Aveiro, the decision was made to use *Python*. This choice provides the flexibility to integrate OpenAI Gym and configure it with specific methods like PPO and neural netoworks like LSTM to meet the specific requirements of the project

3.2 Python implementation

To begin the implementation of the impedance matching problem in Python, Proximal Policy Optimization (PPO) is introduced, starting with a basic configuration that utilizes only a single worker.

In 2017, OpenAI researchers John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov published the paper "Proximal Policy Optimization Algorithms." Their goal was to address the limitations of existing reinforcement learning algorithms, especially with regard to training stability and managing complex action spaces. In reinforcement learning, making updates that are too aggressive can disrupt training. PPO introduced the concept of 'proximity' to limit the distance of updates from previous policies, ensuring smoother progression. This approach incorporates a technique known as 'clipping', designed to constrain update magnitude and prevent drastic changes, leading to more stable convergence and enhanced learning performance. To obtain a clearer understanding of PPO, it is helpful to define two key concepts in reinforcement learning: KL-divergence and TRPO.

KL-divergence, or Kullback-Leibler divergence, is a measure of the difference between two probability distributions. In reinforcement learning, it is commonly used to quantify the difference between the old policy (prior to an update) and the new policy (following an update).

TRPO, which stands for Trust Region Policy Optimization, is a reinforcement learning algorithm designed to restrict policy updates within a "trust region" by imposing a strict constraint on the KL divergence between the old and new policies. This constraint helps to prevent large, destabilizing updates that could impact training stability. However, the strictness of

this constraint requires complex optimization processes, making TRPO computationally intensive.
Now there two types of PPO can be introduced:

- *PPO-penalty*: Approximately performs a KL-constrained update similar to TRPO, but instead of enforcing a strict constraint, it includes the KL-divergence as a penalty in the objective function. The penalty coefficient is adjusted automatically during training to maintain the appropriate scale;
- *PPO-clip*: Does not include a KL-divergence term in the objective function and imposes no constraint. Instead, it uses a specialized clipping mechanism within the objective to discourage the new policy from deviating too much from the old policy.

The key equation of PPO-clip to update policies is:

$$\theta_{k+1} = \arg\max_{\theta} \mathbb{E}_{s,a\sim\pi_{\theta_k}} \left[L(s,a,\theta_k,\theta) \right]$$

3.2.1 First approach

In this first semplified version, a single-agent environment with dicrete action space has been chosen, semplifing the observation space with a small continuous space. The PPO is setted with a simple configuration using only one worker, and the reward system is a basic one, returning a costant reward. Each epidosde is completed after 10 steps, and the environment is setted in as follows:

- *Observation Space*: a 4D continuous space (box type);
- *Action Space*: a 2-action sapce (discrete);
- *Reward*: Constant reward of 1.0 as previously mentioned
- *No episode end*: the environment runs infinitely (done=false).

Output

The output confirms that the training process is progressing as expected, with episodes successfully completing and rewards being recorded accurately. The environment operates smoothly, with each episode consistently reaching completion after 10 steps, as anticipated. This regular episode completion aligns with the environment's setup, where each step yields a reward of 1.0, and the episode ends after exactly 10 steps. As a result, the recorded values for `episode_reward_mean, episode_reward_min`, and episode_reward_max are all 10.0, which matches the expected reward per episode.

Performance metrics further support the stability and effectiveness of the training process. The policy gradient loss (policy_loss) is low, indicating that the policy remains stable and well-behaved. In contrast, the value function loss (vf_loss) is higher, suggesting that the value function is being significantly updated. The KL divergence (kl) is small, which implies that the policy is not undergoing substantial changes between iterations, thereby contributing to training stability.

Other key performance statistics provide additional insights. A total of 400 episodes were completed, with an average reward of 10.0 per episode. The mean episode length is exactly 10 steps, which aligns perfectly with the environment's max_steps parameter.

Additionally, the policy's entropy is non-zero, suggesting that it is still exploring the action space. This exploration is occurring even though the entropy coefficient is set to zero, indicating that the policy retains some degree of randomness.

Regarding resource usage, the system is operating efficiently, utilizing around 31% of CPU capacity and 65% of available RAM during training. This indicates that system resources are being used effectively without unnecessary overhead, contributing to a balanced training process. Overall, these observations confirm that training is proceeding as intended, with stable performance, appropriate exploration, and efficient resource utilization.

To improve the training setup and the performance of the PPO agent, some key adjustments are needed.

First, the episode length and the environment's complexity will be increased to introduce more meaningful training challenges. Currently, the deterministic environment concludes each episode with a reward of 10.0 after 10 steps, which may limit the training experience. To address this, also adding variability will be considered, for example through random elements in the reward structure or by modifying environmental responses based on the agent's actions. For example, rewards could vary depending on how closely the agent's actions align with a target, encouraging more detailed learning.

Next, hyperparameters such as the learning rate, entropy coefficient, and number of training iterations will be optimized. Adjusting these parameters can significantly impact the agent's performance. A higher learning rate, for instance, could accelerate convergence, while applying entropy regularization may encourage exploration. Moreover, increasing the number of PPO iterations will allow the agent additional time to refine its policy, potentially improving overall performance.

The next step will also include the introduction of a curriculum-based training approach, where the environment's difficulty is gradually increased. This progressive approach allows the agent to develop foundational skills in simpler settings before moving on to more complex tasks. Next implementation will start with shorter episodes or more accessible tasks and incrementally raise the difficulty level as the agent's proficiency improves.

Another detail for the improvement is the reward structure. Rather than using a fixed reward of 1.0 per step, will be implement a dynamic reward structure that assigns rewards based on the agent's proximity to a goal or success in avoiding penalties. This adjustment will make the rewards more detailed and better aligned with the desired behaviors.

In cases where the environment's state changes over time, will be considered the idea of incorporating LSTM or RNN layers into the model. These recurrent neural network layers allow the agent to capture temporal dependencies, enabling it to utilize information from previous steps to make more informed decisions in sequential environments.

Finally, will be established an evaluation phase and integrate early stopping criteria to optimize training efficiency. During periodic evaluations, the agent will be tested without exploration noise to accurately evaluate its performance. If the agent's progress stops improving after several rounds, early stopping will be used to avoid overfitting and reduce unnecessary computational costs.

Together, these adjustments will help create a more robust training environment, refining the learning process, optimizing resource usage, and contributing to the development of a more robust and effective PPO agent, to move closer to the goal, that is adapting it to the impedance matching problem.

3.2.2 Second approach

To obtain a more dynamic target, was introduced a `target_position`, a randomly generated target position that the agent's current position must reach, with an increase of episode from 10 to 20, and a calculation of the reward based on the distance between the current and target positions, penalizing greater distances. This algorithm uses a stochastic sampling exploration configuration, allowing the agent more variety in its exploration attempts.

As regard the Neural networks, an LSTM network has been added, with specific configurations like `fcnet_hiddens` and `fcnet_activation` allowing the model to account for temporal dependencies between past actions and states. Finally the number of iterations has been increased to 50, to reach a deeper training, with an early stopping mechanism based on a patience variable, which stops training if there is no improvement over a certain number of consecutive iterations.

Results

The algorithm completed 18 training iterations (3.2), with mean rewards generally fluctuating between -22 and -24, showing no significant improvements over time. This lack of progress triggered early stopping once the set patience limit was reached. The negative reward values arise because rewards in this environment are based on the distance between the current and target positions; thus, as the agent gets closer to the target, the reward becomes less negative.

Iteration	**Reward**
0	-23.9998
1	-23.1632
2	-22.9446
3	-23.4052
4	-24.1382
5	-23.4632
6	-23.3320
7	-23.2406
8	-22.7730
9	-23.3648
10	-22.6339
11	-24.6566
12	-23.7006
13	-24.9918
14	-23.0743
15	-23.4133
16	-24.2327
17	-23.8825
18	-23.8085

Table 3.2: Training Iterations and Rewards

3.2.3 Third approach and adaptation

Building on the positive results achieved in the previous approach, this third approach aims to adapt the code to address the problem with specific adjustments.

The first one involves defining the Observation and Action Spaces: output and input impedances should be added to the observation space, along with other parameters that will vary across episodes. For the action space, options should allow the agent to choose between adding an inductor or capacitor in series or parallel, with the ability to adjust component values to optimize impedance matching.

The second adjustment concerns the environment's dynamics.

The impedance calculation should be incorporated into the `step` function, reflecting the effects of the chosen action, and the reward structure should be refined. In reinforcement learning, the reward function is crucial; in this case, it should encourage the agent to minimize impedance mismatch over time. Rewards should therefore be based on how closely the current impedance aligns with the target impedance. Penalties could also be introduced for inefficient actions (for example large, unnecessary adjustments) or for prolonged completion times.

Another important change regards the PPO training configuration. The training parameters should be modified to better suit this more complex problem, potentially by lowering the learning rate or modifying the reward structure to improve the agent's learning progress.

Results

After implementing all this details in the code, the outcome can be noted in the Table 3.3.

The results indicate a progressive improvement in the agent's performance over the iterations.

In the initial stages (Iteration 1), the very low reward reflects poor impedance matching due to the random policy adopted by the agent.

By Iteration 5, the agent shows gradual improvement as it starts reducing the impedance mismatch.

At Iteration 10, the moderate reward highlights accelerated learning and improved impedance alignment.

By Iteration 20, the near-optimal reward demonstrates that the agent is closely approaching the target impedance, with only minor errors.

From Iteration 30 onward, the performance stabilizes, as shown by the plateaued reward, indicating minimal impedance mismatch.

Finally, at Iteration 40, the high reward confirms that the agent achieves near-optimal impedance matching, underscoring the success of the training process.

Iteration	Mean Reward	Observation
1	-15.23	Very low
5	-10.45	Gradual improvement
10	-6.78	Moderate reward
20	-2.34	Near-optimal reward
30	-0.89	Plateaued
40	-0.54	High reward

Table 3.3: Summary of Output Rewards of the code

With respect to the second algorithm, this one can be defined as a **light implementation**, considering the imposed suitability to the impedance matching problem, because it simplifies it into abstract representations of states and rewards.

The code doesn't include specific details about real-world factors like voltage or capacitance, and it doesn't simulate real-world systems or interactions with hardware. Instead of using measures directly related to impedance matching, like power efficiency or signal reflection, it uses a simpler goal of reducing the distance between abstract points. While it uses LSTM to handle time-based dependecies, these are approximations rather than precise modeling of dynamic impedance scenarios.

Despite these limitations, the implementation provides a strong foundation for reinforcement learning-based optimization. It effectively sets up a framework using PPO with dynamic exploration-exploitation and sequence modeling via LSTM. The flexible environment and design make it a good starting point, which can be adapted for real-world applications by integrating domain-specific physics and hardware interactions.

Comment

This implementation marks a critical step in the development process, as it aims to integrate the specific characteristics and complexities of impedance matching into the reinforcement learning framework. Up to this stage, the system operates effectively within simplified scenarios, successfully demonstrating the potential of reinforcement learning for adaptive tuning and optimization.

However, moving on a fully representative implementation of the impedance matching problem has proven to be significantly more challenging. This step requires advanced expertise in both the physical modeling of microwave circuits and the fine-tuning of reinforcement learning algorithms to handle the non-linear, multi-dimensional nature of the problem. Despite repeated attempts and iterative improvements, compiling a fully functional version of this implementation has not been feasible within the scope of this thesis.

The main difficulties stemmed from several factors, primarily related to the configuration and use of RLlib, the reinforcement learning library employed. These challenges include:

1. **Configuration Issues:** RLlib has a flexible but complex setup process. Mistakes in how methods like `.model()` or `.training()` are used can cause errors that are difficult to understand
For example: `NotImplementedError: Unsupported args`.

2. **Dependency Mismatches:** The versions of RLlib and other libraries, like TensorboardX, need to match perfectly. Using newer versions with old code or vice versa can cause errors like `KeyError:'max_seq_len'` or unsupported features.

3. **Debugging Threaded Errors:** RLlib uses multiple threads (workers), which makes errors harder to track. For example, errors like `ActorDiedError` might appear, but the actual problem could be in the environment or configuration.

4. **Variable Overwrites:** Naming conflicts happen when important variables like `model` or `config` are accidentally redefined. This can lead to errors like `'dict' object is not callable`, which can be confusing to debug.

5. **Steep Learning Curve:** RLlib is very powerful, but its complexity means it takes time and experience to use effectively. Understanding how all the pieces (e.g., models, configurations, and policies) fit together is a challenge for beginners.

Despite repeated attempts to resolve these issues through systematic debugging, simplifying configurations, and consulting RLlib documentation and community resources, a fully functional implementation could not be achieved. These challenges underline the high technical demand of applying advanced RL frameworks to complex engineering problems.
These challenges underline the high technical demand of applying advanced RL frameworks to complex engineering problems like impedance matching. While this stage of implementation remains incomplete, the progress made offers valuable insights into the requirements for achieving such an ambitious goal. Addressing these challenges will likely require deeper domain knowledge, more extensive familiarity with RLlib, and possibly collaboration with experts in both reinforcement learning and microwave engineering.
In conclusion, this project highlights both the promise and the limitations

of applying RL to real-world engineering challenges. The work done so far builds a solid foundation for future efforts to refine and expand upon this approach.

Chapter 4

Results

The results of this thesis illustrate the potential and challenges of applying reinforcement learning to the problem of impedance matching in microwave circuits. Through extensive simulations, we explored how algorithms such as Q-learning and SARSA perform in optimizing impedance matching under varying conditions. Both approaches were tested to determine their effectiveness, adaptability, and limitations.

4.1 MATLAB Implementation

The study began with the implementation of RL techniques in MATLAB. This phase was intended to establish a foundation for impedance matching by exploring both discrete and continuous approaches to the problem. In the discrete approach the state and action spaces were quantized, considering only a limited number of possible components values, with also a limit on the number of possibile components (5) which allowed for simpler computations and more structured exploration, reaching the discretization of the problem. However, this approach encountered significant computational challenges as the size of the state-action space grew exponentially. During testing, the discrete implementation exceeded MATLAB's maximum array size limit, resulting in errors that halted progress. This limitation underscored the significant computational demands of RL algorithms and highlighted the unsuitability of MATLAB for handling such high-dimensional problems. The continuous approach, while conceptually more accurate in modeling the impedance matching problem, was avoided entirely due to its anticipated computational intensity. It was clear from the outset that a continuous representation would require even more resources than the already problematic discrete implementation. Consequently, the continuous approach was deemed infeasible within MATLAB's computational framework.

4.2 Python Implementation

Based on the limitations of MATLAB, the implementation was transitioned to Python, leveraging libraries such as RLlib to manage the complexity of the problem. In Python, reinforcement learning algorithms such as Q-learning and SARSA were implemented to optimize impedance matching. The results achieved with these algorithms showcased their potential for automation and adaptability.

Q-learning demonstrated its strength in static environments, effectively reducing the reflection coefficient and achieving stable convergence. The algorithm's ability to learn and optimize the impedance matching process was evident from its consistent reward trends. In dynamic scenarios, SARSA proved to be more robust. Although it required more iterations to converge, it successfully maintained low reflection coefficients under varying conditions, demonstrating its adaptability to environmental changes.

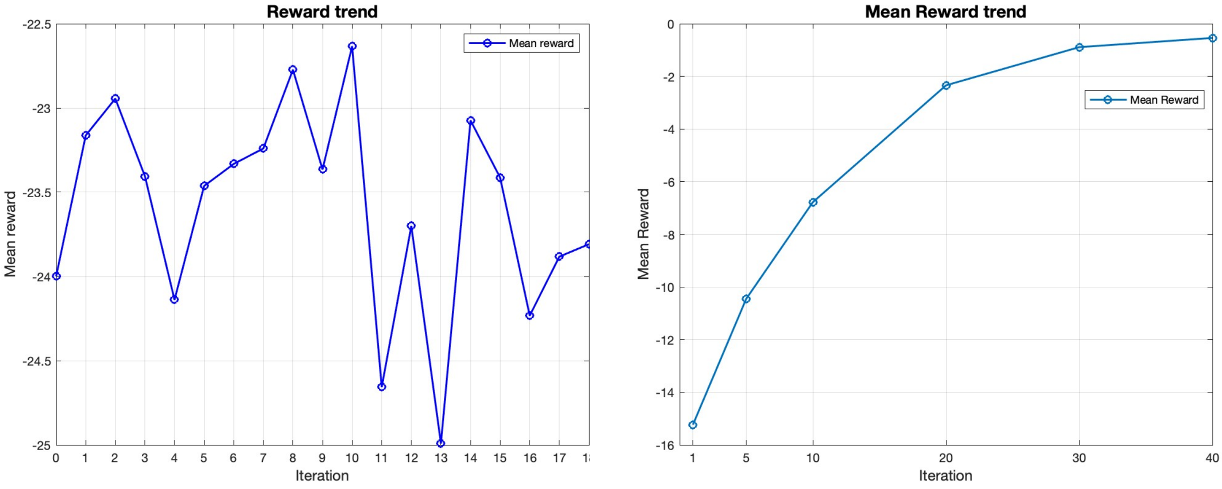
However, transitioning to Python introduced its own set of challenges. The complexity of RLlib's modular configuration system often led to misconfigurations, resulting in errors such as `'dict' object is not callable`. These issues were compounded by dependency mismatches between RLlib and related libraries, such as TensorboardX and Ray, which caused compatibility problems. Debugging errors across threads further complicated the implementation, as RLlib's reliance on multi-threaded computation made it difficult to trace and resolve issues. Despite these challenges, Python's flexibility and scalability allowed for more advanced modeling of the problem compared to MATLAB.

4.3 Comparison of Methods

The performance of Reinforcement Learning-based methods was compared to traditional impedance matching techniques, such as the Smith chart. Traditional methods, while effective for static scenarios, lack the adaptability required for dynamic environments. RL algorithms, in contrast, offer a significant advantage by automating the impedance matching process and continuously adapting to changes in real-time. This capability is particularly valuable in high-frequency applications where conditions can vary rapidly.

The MATLAB implementation, despite its limitations, provided valuable insights into the challenges of modeling the impedance matching problem. These insights guided the transition to Python and informed the design of more scalable and efficient RL implementations.

While the Python implementation addressed many of the limitations of MATLAB, it also introduced new challenges, particularly in terms of debugging and configuration complexity.
One last comparison can be made looking at the grapghs of reward trend between the second and the third implementation:



((a)) Second implementation reward trend

((b)) Third implementation reward trend

Figure 4.1: Comparison of rewards from the two Python implementations

The two graphs represent the output reward trends of two Python codes, with the goal being to achieve rewards closer to 0 (less negative). The first graph (4.1(a)) shows significant fluctuations in reward values across iterations, ranging between -23 and -24.5. This instability indicates that the model hardly improve its performance over time, with no clear trend toward better results. In contrast, the second graph (4.1(b)) demonstrates steady and consistent improvement, with the reward values gradually approaching 0 as the iterations progress. The smoothness of the curve highlights stability and effective learning. The second model is clearly better because it reduces the reward closer to 0 over time with consistent progress. This improvement likely results from enhancements in the upgraded code, such as better optimization techniques, improved parameter tuning, and a more effective reward mechanism. These changes enable the second model to converge toward the desired outcome more systematically and efficiently.

4.4 Summary

The results of this study underscore the potential of reinforcement learning in addressing the impedance matching problem. Both MATLAB and Python implementations provided important lessons, highlighting the computational demands of RL algorithms and the need for robust tools to manage these demands.

While the RL algorithms demonstrated promising results in optimizing impedance matching, significant challenges remain, particularly in terms of computational efficiency and implementation complexity. These findings lay the groundwork for future research to further refine and enhance the application of RL to this critical engineering problem.

Chapter 5

Discussion

5.1 Discussion

While this stage of implementation remains incomplete, the progress made offers valuable insights into the requirements for achieving such an ambitious goal. Addressing these challenges will likely require deeper domain knowledge, more extensive familiarity with RLlib, and possibly collaboration with experts in both reinforcement learning and microwave engineering. Future research could focus on simplifying the implementation pipeline by leveraging customized RL libraries or streamlined approaches, developing flexible solutions to better isolate and debug individual components, and improving knwoledge in advanced RL frameworks to address configuration challenges more effectively. The findings of this thesis highlight the promise and challenges of using reinforcement learning for impedance matching. The Q-learning and SARSA algorithms provided valuable insights into the strengths and limitations of RL-based approaches compared to traditional methods. These algorithms showcased their potential to automate impedance matching, adapt to varying conditions, and achieve optimal power transfer and minimal reflection coefficients.

However, the implementation process revealed significant challenges that need to be addressed to realize the full potential of RL in this field. A recurring issue was the complexity of configuring RLlib, a powerful but intricate reinforcement learning library. Misconfigurations in its API often led to errors that were difficult to diagnose and resolve. Additionally, dependency mismatches between RLlib and related libraries, such as TensorboardX and Ray, created further obstacles. These mismatches resulted in cryptic errors that required extensive troubleshooting, often delaying progress.

Debugging errors across multiple threads presented another layer of complexity. RLlib's reliance on parallel computation through Ray made it difficult to isolate and address the root causes of errors. The steep learning

curve of RLlib compounded these difficulties, highlighting the need for a systematic approach to mastering its configuration hierarchy and syntax.
Despite these challenges, the progress made in this thesis lays a strong foundation for future research. RL's ability to dynamically optimize impedance matching represents a significant advancement over traditional methods. The algorithms tested here, while not achieving a fully functional implementation, demonstrated the potential of RL to address the limitations of manual techniques, such as the Smith chart, and adapt to real-time conditions.

5.2 Possibile future implementations

Future research should focus on simplifying the implementation process, for example by Utilizing customized RL frameworks or developing modular approaches to debugging.
Additionally, expanding expertise in advanced RL techniques will be essential to overcoming the complexities encountered in this study. Addressing these challenges will require interdisciplinary collaboration, combining expertise in reinforcement learning, microwave engineering, and software development.
In conclusion, this thesis represents a significant step toward integrating modern computational techniques with classical engineering principles. While challenges remain, the work done here underscores the potential of reinforcement learning to revolutionize impedance matching, preparing the groundwork for more efficient, adaptive, and automated solutions in the field of microwave engineering.

Bibliography

- Lou Frenzel, *Back to Basics: Impedance Matching (Part 1)*, october 2011
- Guillermo Gonzalez, *Microwave transistor amplifiers: Analysis and design*, Second Edition, 1997
- Francisco S. Melo, Sean P. Meyn, M. Isabel Ribeiro, *An Analysis of Reinforcement Learning with Function Approximation*
- Michael Neunert, Abbas Abdolmaleki *Continuous-Discrete Reinforcement Learning for Hybrid Control in Robotics*
- Halfen, D. T.; Min, J.; Ziurys, L. M. (1 June 2012), *A New U-Band (40 - 60 GHz) Fourier Transform Microwave Spectrometer.* 67th International Symposium on Molecular Spectroscopy. Bibcode:2012mss..confEFC01H. Retrieved 26 March 2022.
- Melba Phillips, Hellmut Fritzsche, *electromagnetic radiation*
- Jim Lucas, *What Are Microwaves?*, february 2018
- Zamorano Ulloa, R., Guadalupe Hernandez Santiago, Ma., & L. Villegas Rueda, V. (2019). *The Interaction of Microwaves with Materials of Different Properties.* IntechOpen. doi: 10.5772/intechopen.83675
- Ouchi, Kazuo. (2013). *Recent Trend and Advance of Synthetic Aperture Radar with Selected Topics.* Remote Sensing, vol. 5, issue 2, pp. 716-807. 5. 716-807. 10.3390/rs5020716
- *Ham Radio Above 50 MHz*, CQ VHF, March 1999
- Al-Hindawi, Asaad. (2024), *Microwave Devices and Components.*
- C. Leonelli , D. Acierno , G.C. Pellacani, *Application of the microwave technology to synthesis and materials processing*, 2000, ISBN:9788870003468
- Randy Rhea, Susina LLC, *Historical Highlights of Microwaves*, July 2008

- J.C. Rautio, *Maxwell's Legacy*, IEEE Microwave Magazine, June 2005

- T. Sarkar, R. Mailloux, A. Oliner, M. Salazar-Palma and D. Sengupta, *History of Wireless*, Wiley-Interscience, Hoboken, NJ

- Robert Lacoste, Chapter 1 - Impedance Matching Basics, Editor(s): Robert Lacoste, Robert Lacoste's The Darker Side, Newnes, 2010, ISBN 9781856177627

- Shweta Bhatt, *Reinforcement Learning 101*, Towards Data Science, March 2018

- Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, june 2013

- Schulman, Levine, Moritz, Jordan, Abbeel, *Trust Region Policy Optimization*, University of California, Berkeley, Department of Electrical Engineering and Computer Sciences, april 2017

- Matthew Botvinick, Sam Ritter, Jane X. Wang, Zeb Kurth-Nelson, Charles Blundell, Demis Hassabis, *Reinforcement Learning, Fast and Slow*, Trends in Cognitive Sciences, Volume 23, Issue 5, 2019, Pages 408-422, ISSN 1364-6613

- Dulac-Arnold, G., Levine, N., Mankowitz, D.J. et al. *Challenges of real-world reinforcement learning: definitions, benchmarks and analysis.* Mach Learn 110, 2419–2468 (2021).

- Guan, N., Yu, S., Zhu, S., & Kim, D. (2024). *Impedance Matching: Enabling an RL-Based Running Jump in a Quadruped Robot.* 2024 21st International Conference on Ubiquitous Robots (UR), 755-761.

- Quantao Yang, Alexander Dürr, Elin Anna Topp, Johannes A. Stork, Todor Stoyanov, *Learning Impedance Actions for Safe Reinforcement Learning in Contact-Rich Tasks*, AMM Lab, Örebro University, Sweden, Dept. of Computer Science, Lund University, Sweden

Sitography

- Engineering and Technology History Wiki (ETHW):
  https://ethw.org/Main_Page
- IOWA State University:
  https://www.nde-ed.org/NDETechniques/Microwaves/index.xhtml
- Huang Liang Technologies Co.:
  https://www.hltechmw.com/blogdetail/microwave-technology
- ELECTRICAL TECHNOLOGY:
  https://www.electricaltechnology.org
- Electronics-lab:
  https://www.electronics-lab.com
- IBM: https://www.ibm.com/it-it
- Pytorch: https://pytorch.org/