



**UNIVERSIDADE FEDERAL DO TOCANTINS
CAMPUS UNIVERSITÁRIO DE PALMAS
CURSO DE ENGENHARIA ELÉTRICA**

RENAN MACHADO ALVES VAZ

**OTIMIZAÇÃO DE PROJETOS DE ANTENAS *MICROSTRIP*
SIMULAÇÃO DE CÉLULAS METAMATERIAIS SRR COM *MACHINE LEARNING***

Palmas – TO
2019

RENAN MACHADO ALVES VAZ

OTIMIZAÇÃO DE PROJETOS DE ANTENAS *MICROSTRIP*
SIMULAÇÃO DE CÉLULAS METAMATERIAIS SRR COM *MACHINE LEARNING*

Monografia apresentada à disciplina de Trabalho de Conclusão de Curso de Engenharia Elétrica do Campus Palmas da Universidade Federal do Tocantins – UFT para obtenção do título de Bacharelado em Engenharia Elétrica.
Orientador: Dr. Humberto Xavier de Araújo

Palmas – TO

2019

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal do Tocantins

V393o Vaz, Renan Machado Alves.
Otimização de projetos de antenas microstrip: Simulação de células metamateriais SRR com machine learning . / Renan Machado Alves Vaz. – Palmas, TO, 2019.
45 f.

Monografia Graduação - Universidade Federal do Tocantins – Câmpus Universitário de Palmas - Curso de Engenharia Elétrica, 2019.
Orientador: Humberto Xavier de Araújo

1. Antenas. 2. Machine learning. 3. Otimização. 4. Metamateriais.
I. Título

CDD 621.3

TODOS OS DIREITOS RESERVADOS – A reprodução total ou parcial, de qualquer forma ou por qualquer meio deste documento é autorizado desde que citada a fonte. A violação dos direitos do autor (Lei nº 9.610/98) é crime estabelecido pelo artigo 184 do Código Penal.

Elaborado pelo sistema de geração automática de ficha catalográfica da UFT com os dados fornecidos pelo(a) autor(a).

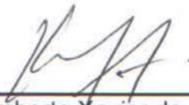
RENAN MACHADO ALVES VAZ

OTIMIZAÇÃO DE PROJETOS DE ANTENAS MICROSTRIP
SIMULAÇÃO DE CÉLULAS METAMATERIAIS SRR COM MACHINE LEARNING

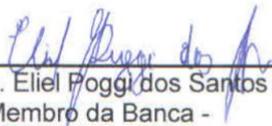
Monografia apresentada à disciplina de Trabalho de Conclusão de Curso de Engenharia Elétrica do Campus Palmas da Universidade Federal do Tocantins – UFT para obtenção do título de Bacharelado em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Data de Aprovação: 04/12/2019

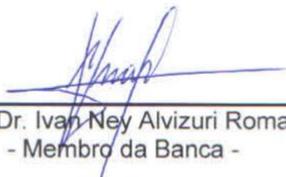
Banca examinadora:



Prof. Dr. Humberto Xavier de Araujo
- Orientador -



Prof. Dr. Eliel Poggi dos Santos
- Membro da Banca -



Prof. Dr. Ivan Ney Alvizuri Romani
- Membro da Banca -

RESUMO

O projeto de antenas de microfita é amplamente documentado pela literatura e oferece bons resultados quando a antena é fabricada. Entretanto, como essa arquitetura inclui baixa eficiência e largura de banda limitada, muitas vezes é preciso otimizar o projeto sem alterar as dimensões físicas da antena a fim de atender os parâmetros especificados da aplicação. Uma forma de conseguir isso é dada pela engenharia de materiais: a aplicação de metamateriais. Metamateriais são materiais com características elétricas notórias. Quando aplicados em antenas, eles fazem com que a radiação eletromagnética se comporte de uma maneira diferente do esperado, melhorando as especificações, e conseqüentemente o desempenho, sem alterar o tamanho. Todavia, a geometria das células metamateriais e o seu arranjo tornam o desenvolvimento analítico das equações dos campos eletromagnéticos resultantes muito complexo, se tornando impraticável. Engenheiros de telecomunicações tratam esse problema projetando a antena e os arranjos com base em simulações eletromagnéticas computacionalmente trabalhosas e demoradas. Com a popularização de bibliotecas de ciência de dados como *scikit-learn*, *pytorch*, *tensorflow*, dentre outras, esse trabalho visa desenvolver modelos preditivos de aprendizagem de máquina para contornar os custos computacionais que um simulador de dispositivos de micro-ondas consome na otimização de projetos de antenas de microfita através de arranjos metamateriais.

Palavras-chave: antenas de microfita; metamateriais; aprendizagem de máquina;

ABSTRACT

The design of microstrip antennas is widely documented in the literature and presents good results when an antenna is manufactured. However, since the technology includes low efficiency and limited bandwidth, it is often necessary to optimize the design without altering its physical dimensions to meet the requirements of the application. One way to achieve this is given by materials engineering: an application of metamaterials. Metamaterials are materials with remarkable electrical characteristics. When applied to antennas, the electromagnetic radiation behaves differently than expected, improving its specs, hence the performance, without altering the size. However, the geometry of the metamaterial cells and their array makes the analytical development of the resultant electromagnetic field equations more complex, becoming unworkable. Telecommunication engineers deal with this problem by designing an antenna with a metamaterial array based on computationally expensive and time-consuming electromagnetic simulations. With the popularization of data science libraries such as scikit-learn, pytorch, tensorflow, and others, this work aims to allow the optimization of antenna designs through development of predictive machine learning models to beat the computational costs of a microwave simulator optimization.

Keywords: microstrip antennas; metamaterials; machine learning;

LISTA DE FIGURAS

Figura 1 – Diagrama de radiação de potência (em dBi)	11
Figura 2 – Geometria de uma antena de microfita.....	13
Figura 3 – Célula SRR	16
Figura 4 – Fluxograma do aprendizado supervisionado	17
Figura 5 – Identificação de grupos organizados por seus recursos x_1 e x_2	18
Figura 6 – Sob ajuste e sobre ajuste	19
Figura 7 – Fluxograma de trabalho na criação de um modelo preditivo	20
Figura 8 – Geometria da antena base.....	22
Figura 9 – S_{11} da antena base	22
Figura 10 – Parametrização da célula SRR	23
Figura 11 – Parametrização do arranjo na aplicação lateral	24
Figura 12 – Prévia das variáveis independentes	25
Figura 13 – Prévia das previsões	26
Figura 14 – Informações detalhadas de cada campo do conjunto de dados.....	27
Figura 15 – Ganho pela quantidade de células no arranjo	28
Figura 16 – Variância dos recursos	28
Figura 17 – Verificação de peso dos recursos para o ganho	29
Figura 18 – Somatória dos erros na regressão linear.....	31
Figura 19 – Problema na somatória dos resíduos	31
Figura 20 – Antena utilizada na comparação.....	33
Figura 21 – S_{11} da antena de comparação.....	34
Figura 22 – Tempos de execução	34

LISTA DE TABELAS

Tabela 1 – Valores dos parâmetros da antena base (em milímetros).....	21
Tabela 2 – Resultados da antena base (para $f = 3,5 \text{ GHz}$).....	22
Tabela 3 – Dimensões da antena utilizada na comparação.....	32
Tabela 4 – Resultados da antena de comparação (para $f = 3,5 \text{ GHz}$)	33

LISTA DE ABREVIATURAS E SIGLAS

SRR	– <i>Split Ring Resonator</i>
dBi	– Decibel referente ao radiador isotrópico
dB	– Decibel
LHM	– <i>Left Handed Materials</i>
NIM	– <i>Negative Refractive Index Materials</i>
DNG	– <i>Double Negative</i>
TM	– <i>Transverse Mode</i>
ANATEL	– Agência Nacional de Telecomunicações
MWC	– <i>Mobile World Congress</i>

LISTA DE SÍMBOLOS

r	– Distância entre o observador e a antena
d	– Maior dimensão da antena
λ	– Comprimento de onda
U	– Intensidade de radiação
W_{rad}	– Densidade de radiação
P_{rad}	– Potência radiada
U_0	– Intensidade de radiação de uma antena isotrópica
η_r	– Eficiência de radiação
P_0	– Potência aceita pela antena
D	– Diretividade
G	– Ganho
S_{11}	– Perda por retorno
ϵ_r	– Permissividade relativa do substrato
h	– Altura do substrato
L	– Largura do <i>patch</i>
W	– Comprimento do <i>patch</i>
W_0	– Comprimento da linha de alimentação
Y_0	– Largura da entrada da linha de alimentação
g	– Comprimento da entrada da linha de alimentação
ϵ_{eff}	– Constante dielétrica efetiva
ΔL	– Diferença entre largura física e elétrica
L_{eff}	– Largura efetiva
$f_{rc_{010}}$	– Frequência ressonante do modo TM_{010}
μ_0	– Permeabilidade magnética do vácuo
ϵ_0	– Permissividade elétrica do vácuo

SUMÁRIO

1 INTRODUÇÃO	9
2 FUNDAMENTAÇÃO TEÓRICA	10
2.1 Antenas	10
2.2 Antenas de Microfita.....	12
2.2.1 Formulação Analítica.....	13
2.2.2 <i>Framework</i> para Projetos.....	14
2.3 Células Metamateriais	15
2.3.1 Metamateriais	15
2.3.2 <i>Split Ring Resonator</i>	15
2.4 <i>Machine Learning</i>	16
2.4.1 Tipos de aprendizado.....	17
2.4.2 Terminologia	18
2.4.3 Avaliação de Desempenho	19
2.4.4 <i>Framework</i> para Construção de Modelos.....	20
3 BASE DE DADOS.....	21
3.1 Metodologia	21
3.1.1 Antena Base	21
3.1.2 Parametrização.....	23
3.2 Concatenação.....	24
4 MODELAGEM	25
4.1 Seleção de Variáveis e Definição de Previsões.....	25
4.2 Limpeza e Pré-processamento dos Dados.....	26
4.3 Ganho	27
4.3.1 Variância	28
4.3.2 Seleção de Recursos	29
4.3.3 Treinamento do Modelo	29
4.4 Perda por Retorno e Largura de Banda	30
5 RESULTADOS.....	31
5.1 Modelos	32
5.2 Comparação	32
6 CONCLUSÃO.....	35
REFERÊNCIAS BIBLIOGRÁFICAS	36
ANEXO I – <i>LIVE SCRIPT</i> PARA APLICAÇÃO LATERAL DE CÉLULAS SRR	38

1 INTRODUÇÃO

Antenas fazem a interface entre o espaço livre e um dispositivo guia. É ela quem faz a interface entre o transmissor/receptor e o meio de propagação. Dentre os diversos tipos de antena, a antena de microfita¹ se destaca na telefonia celular, sendo empregada na maioria dos *smartphones* atuais como o dispositivo receptor de sinal. Essa arquitetura é ótima para essa aplicação devido a sua geometria planar e tamanho reduzido, entretanto, há alguns contrapontos característicos como o baixo ganho e eficiência de radiação.

O uso de metamateriais no projeto de antenas é uma importante área de pesquisa que ajuda a reduzir o tamanho e maximizar o desempenho dessas antenas através da alteração dos valores de permeabilidade magnética e permissividade elétrica (GIANFAGNA, SWAMINATHAN, *et al.*, 2015). Todavia, a geometria das células metamateriais, e conseqüentemente o seu arranjo, tornam o desenvolvimento analítico das equações dos campos eletromagnéticos resultantes impraticável. Dessa forma, o engenheiro de telecomunicações trabalha no projeto dessas antenas através de simulações eletromagnéticas trabalhosas e demoradas, utilizando o método de tentativa e erro.

Dada a recente tendência de pesquisa e mercado na área de ciência de dados, a sua aplicação na área de telecomunicações é recente e ainda pode ser considerada como imatura e em desenvolvimento. Materiais como os de (GIANFAGNA, SWAMINATHAN, *et al.*, 2015) e (KUSHWAH e TOMAR, 2017) são referências no assunto e mostram o quão nova é a aplicação de *machine learning* no projeto de antenas. Com isso, identificou-se a oportunidade de projetar um modelo de *machine learning* capaz de facilitar simulações de um projeto de antenas de microfita com células metamateriais do tipo *Split Ring Resonator* (SRR).

Inicialmente será feita uma definição das características/recursos (*features*)² e alvos (*targets*) a serem avaliados em uma antena, seguido pela criação de uma extensa base de dados com simulações eletromagnéticas heterogêneas, o pré-processamento desta base de dados, a escolha do modelo, treino, validação cruzada e aperfeiçoamento dos parâmetros para finalmente obter os resultados e compará-los com os resultados de um simulador.

¹ Vide seção 2.2

² A terminologia básica será explicada na seção 2.4.2

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção se resume a introduzir brevemente ao leitor conceitos que serão utilizados nas seções 3 em diante, como os diversos parâmetros de uma antena, exploração da arquitetura de microfita e *machine learning*.

2.1 Antenas

Uma antena é uma estrutura transitória entre o espaço livre e um dispositivo guia (BALANIS, 2005). Esse último dispositivo é comumente chamado de linha de transmissão e é amplamente estudado como guia de ondas na Engenharia de Telecomunicações. As linhas de transmissão podem ser da forma de um cabo coaxial ou um retângulo metálico oco e são usadas para transportar a energia eletromagnética da fonte para a antena, ou da antena para o receptor. Nesse escopo, na primeira situação temos uma antena transmitindo e na segunda temos uma antena recebendo o sinal.

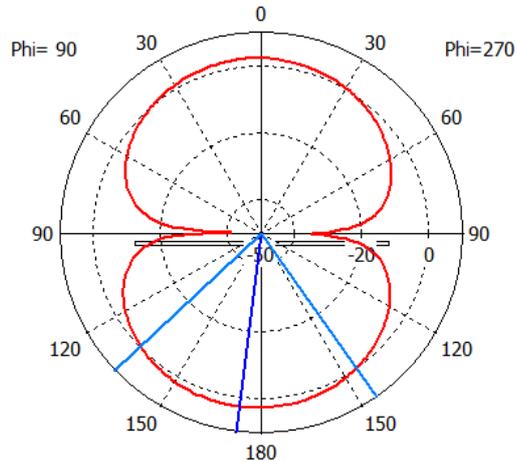
Para descrever o desempenho de uma antena, é necessário definir uma variedade de parâmetros. Alguns desse parâmetros são inter-relacionados e nem todos precisam ser necessariamente especificados para descrever completamente o desempenho de uma antena (BALANIS, 2005).

Segundo Balanis (2005), o diagrama de radiação de uma antena é definido como uma função matemática, ou representação gráfica, das propriedades da antena em um espaço de coordenadas, tal como mostrado na Figura 1. Essas propriedades incluem intensidade e densidade de radiação, força de campo, diretividade, fase ou polarização. Também é importante citar que essas propriedades são usualmente normalizadas com base no seu valor máximo, podendo estar em escala logarítmica ou em decibéis (dB). Nesse gráfico normalmente têm-se lóbulos, que são concentrações fechadas de campo, normalmente simétricos. Se uma antena só possui um lóbulo de raio constante em todas direções, se diz que essa antena é isotrópica, ou seja, ela irradia potência independente da direção.

O espaço a cerca de uma antena também é importante de ser estudado. Este é subdividido em três regiões: campo próximo reativo, campo próximo radiante e campo distante. A região de campo próximo reativo é definida como a porção de espaço mais próximo da antena onde o campo reativo predomina. Para a maioria das

antenas, essa região se estende a uma distância $r < 0,62\sqrt{d^3/\lambda}$ a partir da superfície da antena, onde λ é o comprimento de onda do sinal e D é a maior dimensão da antena (BALANIS, 2005).

Figura 1 – Diagrama de radiação de potência (em dBi)



Fonte: Autoria própria

A região de campo próximo radiante, também conhecida como região de Fresnel, fica situada entre as regiões de campo próximo reativo e de campo distante. Nessa região, a distribuição de campo depende da distância da antena (BALANIS, 2005). Se a dimensão d da antena não for muito maior do que o comprimento de onda λ , essa região pode não existir.

Por último, tem-se a região de campo distante também denominada como região de Fraunhofer. É definida como a região onde a distribuição de campo é independente da distância da antena (BALANIS, 2005). Normalmente essa região começa a partir de uma distância de $2d^2/\lambda$ metros.

Outro parâmetro importante é a intensidade de radiação, definida como a potência radiada pela antena por unidade de ângulo sólido em uma dada direção (BALANIS, 2005). Esse parâmetro só faz sentido na região de campo distante e pode ser obtido multiplicando o quadrado da distância (r) pela densidade de radiação (W_{rad}), conforme equação (1).

$$U = r^2 W_{rad} \quad (1)$$

Para uma antena isotrópica, a intensidade de radiação será independente da direção, já que a densidade de potência é constante ao longo de uma esfera. Se a potência radiada por essa antena for P_{rad} , a intensidade de radiação U_0 será dada conforme a equação (2).

$$U_0 = \frac{P_{rad}}{4\pi} \quad (2)$$

É importante citar que a antena possui uma eficiência de radiação (η_r) que associa a potência aceita pela antena (P_0) e a potência radiada (P_{rad}), como é mostrado na equação (3).

$$\eta_r = \frac{P_{rad}}{P_0} \quad (3)$$

Já a diretividade é uma medida da habilidade de uma antena concentrar a potência radiada em uma direção particular. Se a direção não for especificada, a diretividade se refere a direção de máxima intensidade de radiação ($U = U_{max}$) (BALANIS, 2005). É possível representar essa medida adimensional matematicamente através da equação (4).

$$D = \frac{U}{U_0} = \frac{4\pi U}{P_{rad}} \quad (4)$$

O ganho de uma antena é uma medida da habilidade de concentrar, em uma direção particular, a potência aceita pela antena. Essa medida também é adimensional e é mais utilizada do que a diretividade em projetos, já que é mais fácil medir a potência aceita pela antena do que a potência radiada pela mesma. Sendo G o ganho, podemos relacioná-lo com a diretividade D através da equação (5). Utopicamente, se a antena não apresentar perdas ($\eta_r = 1$), a diretividade e o ganho serão idênticos.

$$G = \eta_r D = \frac{4\pi\eta_r U}{P_{rad}} = \frac{4\pi U}{P_0} \quad (5)$$

A largura de banda de uma antena é um (ou mais de um) intervalo de frequência onde a antena consegue operar com condições aceitáveis. Esse intervalo é definido para frequências onde a perda por retorno (S_{11}) é menor do que -10 dB (BALANIS, 2005). Normalmente a largura de banda é descrita como:

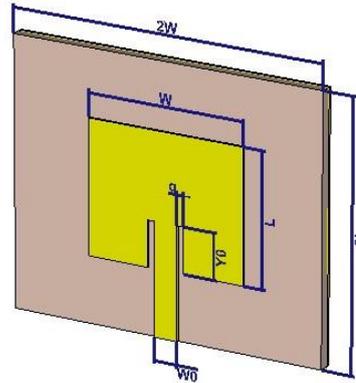
- Uma porcentagem a partir da menor frequência de operação;
- Uma margem acerca da frequência central.

2.2 Antenas de Microfita

Essas antenas também são chamadas de “antenas *microstrip*” por serem construídas com finas placas metálicas de circuito impresso, podendo ser planares ou não-planares. Elas são aplicadas em aeronaves de alta performance, veículos espaciais, satélites e mísseis, onde é preciso ter um dispositivo mais discreto devido

a restrições de peso, custo e desempenho. Em aplicações civis, podemos citar a telefonia móvel e comunicação *wireless* (BALANIS, 2005). As desvantagens dessa arquitetura incluem baixa eficiência, baixa potência, largura de banda limitada e baixa pureza de polarização.

Figura 2 – Geometria de uma antena de microfita



Fonte: Autoria própria

Como mostra a Figura 2, as antenas de microfita são constituídas por uma fita metálica muito fina denominada *patch*, um bloco dielétrico com permissividade relativa ϵ_r denominado substrato e uma outra fita metálica chamada de plano terra. O *patch* é projetado para que o valor máximo do diagrama de radiação seja normal a abertura radiante lateral (BALANIS, 2005). A escolha do substrato deve ser feita preferindo-se camadas finas com constantes dielétricas baixas, mas isso pode mudar de acordo com a aplicação. De forma geral, essa escolha leva em conta o melhor custo-benefício entre o desempenho da antena e o projeto do circuito.

De acordo com Pozar (1992), o substrato dielétrico deve ser, na maior parte dos casos, eletricamente fino e a constante dielétrica deve variar entre $2,2 \leq \epsilon_r \leq 12$. O *patch* entra em ressonância quando L é próximo de $\lambda/2$. A impedância de entrada é controlada ajustando a posição do ponto de alimentação, variando de 50Ω a 300Ω .

2.2.1 Formulação Analítica

Como algumas ondas viajam no substrato e algumas no ar, uma constante dielétrica efetiva ϵ_{reff} precisa ser introduzida de acordo com a equação (6).

$$\epsilon_{reff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left[1 + 12 \frac{h}{W} \right]^{-\frac{1}{2}} \quad (6)$$

para $W/h > 1$

Por causa dos efeitos de borda, o *patch* aparenta ser eletricamente maior do que realmente é, tendo uma diferença ΔL que é função de ϵ_{reff} e da razão W/h entre a largura real e a largura elétrica, de acordo com a equação (7).

$$\frac{\Delta L}{h} = 0,412 \frac{(\epsilon_{reff} + 0,3) \left(\frac{W}{h} + 0,264\right)}{(\epsilon_{reff} - 0,258) \left(\frac{W}{h} + 0,8\right)} \quad (7)$$

Com essa extensão da largura, é então definido uma “largura efetiva” conforme equação (8). Se desconsiderarmos o efeito de borda, $L = \lambda/2$ para o modo dominante TM_{010} .

$$L_{eff} = L + 2\Delta L \quad (8)$$

Para o modo dominante TM_{010} , a frequência ressonante é função desses últimos parâmetros definidos, como mostrado na equação (9).

$$f_{rc010} = \frac{1}{2L_{eff}\sqrt{\epsilon_{reff}}\sqrt{\mu_0\epsilon_0}} = \frac{1}{2(L + 2\Delta L)\sqrt{\epsilon_{reff}}\sqrt{\mu_0\epsilon_0}} \quad (9)$$

2.2.2 Framework para Projetos

A escolha da antena é feita de acordo com a aplicação e a frequência (THALLURI, GAYATHRI, *et al.*, 2016). O comprimento e a largura do *patch* são determinados de acordo com a frequência escolhida. A largura L impacta diretamente na frequência de operação da antena, enquanto que o comprimento W influencia a eficiência de radiação (BALANIS, 2005). Assim, em um projeto:

1. Determinar aplicação, conseqüentemente, a frequência de operação;
2. Especificar parâmetros satisfatórios mencionados na seção 2.1.
3. Especificar os materiais (metal da microfita, dielétrico) e as suas espessuras;
4. Calcular o comprimento e a largura do *patch*;
5. Simular e verificar se os resultados satisfazem os parâmetros definidos no item 2.
6. Caso não forem satisfeitos, retomar item 3 e verificar meios de otimização.

2.3 Células Metamateriais

Existem várias formas de minimizar as limitações das antenas de microfita e fazer com que as características da antena satisfaçam os parâmetros de projeto. É possível introduzir novos materiais às estruturas da antena a fim de reduzir os efeitos negativos ou aumentar os efeitos positivos (GUELBER, CARDOSO, *et al.*, 2016). Nesse escopo de novos materiais, um padrão que se destaca muito são os metamateriais.

2.3.1 Metamateriais

A permissividade elétrica (ϵ) e a permeabilidade magnética (μ) são ambas positivas para materiais presentes na natureza. Veselago, o pai dos metamateriais, trouxe a atenção da comunidade científica para materiais um tanto incomuns denominados *Left Handed Materials* (LHM). Com eles, é possível obter valores negativos para ϵ e μ usando geometrias específicas chamadas de metamateriais (OZBAY e AYDIN, 2009). Esses materiais também podem ser referenciados como *Negative Refractive Index Materials* (NIM).

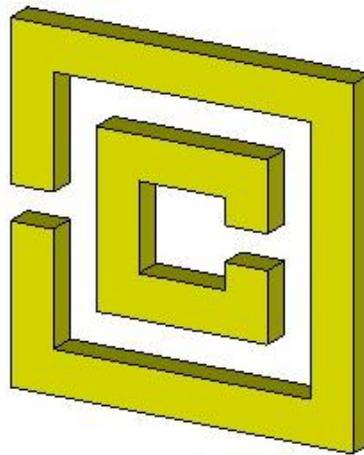
A combinação da permissividade negativa com a permeabilidade negativa resulta em um metamaterial com um índice de refração negativa. Isso faz com que a resposta à uma radiação eletromagnética emitida seja diferente da esperada (GUELBER, CARDOSO, *et al.*, 2016). Metamateriais oferecem propriedades notáveis como refração negativa, imagens de subcomprimento de onda (nano escala) e camuflagem.

Os metamateriais pertencem a classe *Double Negative* (DNG) por consistirem em um arranjo de estruturas periódicas onde ambos o arranjo e as células unitárias são pequenos quando comparados ao comprimento de onda de operação. Segundo Wiltshire (2009), a homogeneização das estruturas permite com que elas sejam descritas como um material único para com as constantes eletromagnéticas convencionais de permissividade e permeabilidade, mas com valores que não poderiam ser obtidos antes na natureza.

2.3.2 Split Ring Resonator

Um arranjo periódico de anéis metálicos fechados irá apresentar um comportamento diamagnético quando o comprimento de onda da radiação incidente for menor do que a periodicidade, apresentando uma polarização magnética negativa (MARQUÉS e MARTÍN, 2009). Essas células unitárias podem ser circulares ou quadradas, tendo diversas variações na geometria a fim de obter melhores parâmetros. A Figura 3 mostra uma célula SRR. Diante da variedade de células unitárias LHM, iremos utilizar somente células SRR nas seções 3 em diante.

Figura 3 – Célula SRR



Fonte: Autoria própria

2.4 Machine Learning

Esta é uma época em que dados são abundantes, sejam eles estruturados ou não. Segundo Raschka (2016), na segunda metade do século XX, *machine learning* evoluiu como um segmento da inteligência artificial, que envolveu o desenvolvimento de algoritmos autodidatas que ganhavam conhecimento a partir dos dados a fim de fazer previsões. É graças a esses modelos preditivos que podemos desfrutar de filtros robustos de spam, softwares de reconhecimento de texto e voz, mecanismos de buscas confiáveis, jogadores de xadrez desafiadores e, em breve, carros que dirigem por nós.

Quando se fala de *machine learning*, há três tipos distintos de aprendizado de máquina, sendo eles: aprendizado supervisionado, aprendizado não-supervisionado e aprendizado reforçado (RASCHKA, 2016).

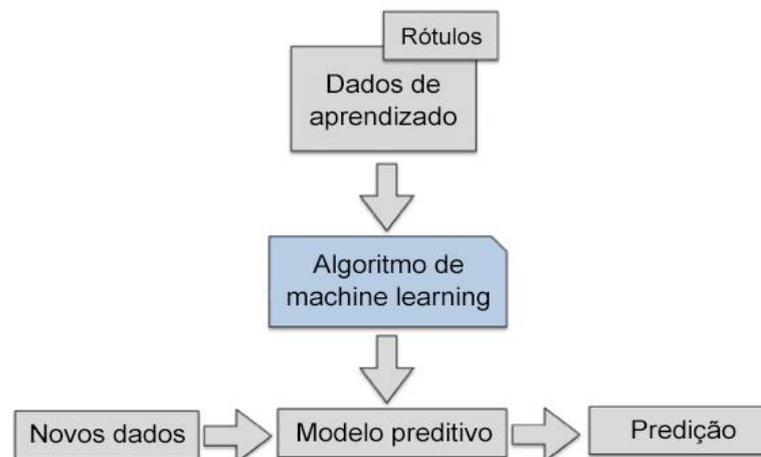
2.4.1 Tipos de aprendizado

No aprendizado supervisionado, o objetivo principal é fazer o modelo aprender com base em dados de treinamento rotulados para fazer previsões de dados futuros ou nunca antes vistos. Dados rotulados se referem a amostras onde a saída desejada já é conhecida.

Considere o exemplo de um filtro de *spam*. O modelo preditivo é treinado com base na metodologia de aprendizado supervisionado, onde as amostras de e-mails da base de dados de aprendizagem já estão corretamente rotuladas como *spam* ou *não-spam*. Assim, o modelo aprende o que é um *spam*, e o que não é um *spam*. Dessa forma, ele ganha a habilidade de prever se um novo e-mail pertence a uma das duas categorias. Esse processo é exemplificado na Figura 4.

Uma tarefa de aprendizado supervisionado com categorias discretas, como o exemplo anterior (*spam* ou *não-spam*) é denominada classificação. Outra tarefa de aprendizado supervisionado é denominada regressão, onde a saída desejada pode assumir valores contínuos.

Figura 4 – Fluxograma do aprendizado supervisionado



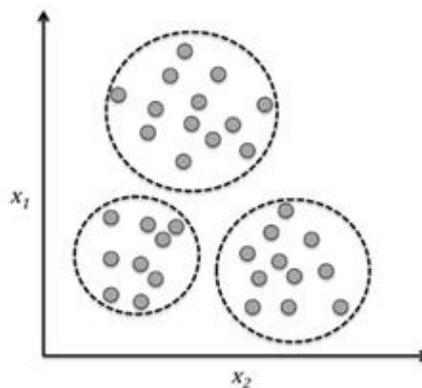
Fonte: (RASCHKA, 2016), adaptado

Enquanto que no aprendizado supervisionado sabe-se a resposta correta previamente, no aprendizado não-supervisionado os dados não são rotulados. Usando técnicas de aprendizado não-supervisionado, podemos explorar a estrutura dos nossos dados para extrair informações relevantes. O algoritmo aprende a correlação entre as *features* por si só.

Uma técnica bastante reconhecida se chama *clustering* e baseia-se na organização de subgrupos (*clusters*). Cada *cluster* que o algoritmo identifica durante

a análise define um grupo de objetos que compartilham um certo grau de similaridade, conforme mostra a Figura 5. Um exemplo de aplicação de *clusters* são algoritmos que descobrem grupos de clientes baseado nos seus respectivos interesses com a intenção de desenvolver programas de *marketing* direcionados.

Figura 5 – Identificação de grupos organizados por seus recursos x_1 e x_2



Fonte: (RASCHKA, 2016)

No aprendizado reforçado, o objetivo é desenvolver um sistema (*agent*) que aumenta seu desempenho com base nas interações com o ambiente. Nesse tipo de aprendizado, a saída é uma medida do quão boa a predição foi com base em uma função de recompensa (*reward*) (RASCHKA, 2016). Pode-se pensar que o aprendizado reforçado é fruto de uma mesclagem entre o aprendizado supervisionado e o não-supervisionado. Um bom exemplo desse tipo de algoritmo são os jogadores de xadrez, algoritmos que aprendem quais movimentos levam ao sucesso. O sistema decide em uma série de movimentos baseados no estado do tabuleiro (ambiente), e a recompensa pode ser a vitória ou a derrota no fim do jogo.

2.4.2 Terminologia

- Dados de aprendizagem (*training data*): dados que o algoritmo vai utilizar para aprender sobre as amostras;
- Dados de teste/validação (*test data*): dados que o algoritmo nunca observou antes;
- Características/recursos (*features*): são os dados de uma amostra, normalmente são as colunas da base de dados;

- Alvo (*target*): é a saída do algoritmo, o valor que se quer prever ou o valor da amostra que o algoritmo vai usar para aprender o que caracteriza uma categoria;
- Validação cruzada (*cross validation*): revezamento de partes dos dados de aprendizado para evitar viés (ou vício) no modelo preditivo.

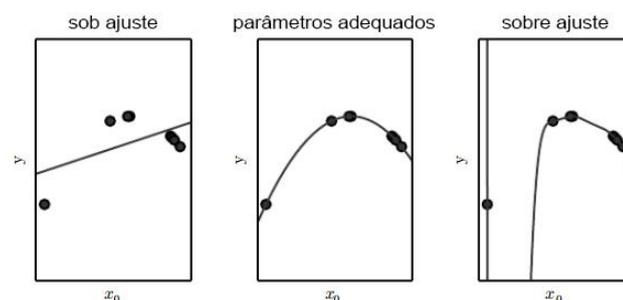
2.4.3 Avaliação de Desempenho

O principal desafio em *machine learning* é desempenhar bem não só naqueles dados usados no aprendizado do modelo, mas em dados nunca antes vistos, novas amostras (GOODFELLOW, BENGIO e COURVILLE, 2016). A habilidade de desempenhar bem em novas amostras é denominada generalização.

Uma base de dados é utilizada para treinar um modelo de *machine learning*. Durante a aprendizagem, é computado uma métrica denominada erro de treinamento e o algoritmo aprende minimizando esse erro, diminuindo-o significativamente. Com isso, deseja-se que o erro de generalização, muitas vezes chamado de erro de teste, também seja o menor possível. Assim, é comum haver uma distinção entre as bases de dados: uma para aprendizagem (*train set*) e outra para teste/validação (*test set*).

Sob ajuste (*underfitting*) acontece quando o modelo não consegue obter um erro suficientemente baixo na base de dados de aprendizagem, isso acontece quando o modelo é muito simples e não consegue separar bem o que caracteriza um alvo (*target*). Sobre ajuste (*overfitting*) ocorre quando a diferença entre o erro de treinamento e o erro de teste é muito grande, ou seja, o modelo desempenha bem nos dados de aprendizagem, mas não consegue generalizar o conhecimento para novas amostras. A Figura 6 mostra essas duas situações juntamente com o resultado almejado.

Figura 6 – Sob ajuste e sobre ajuste

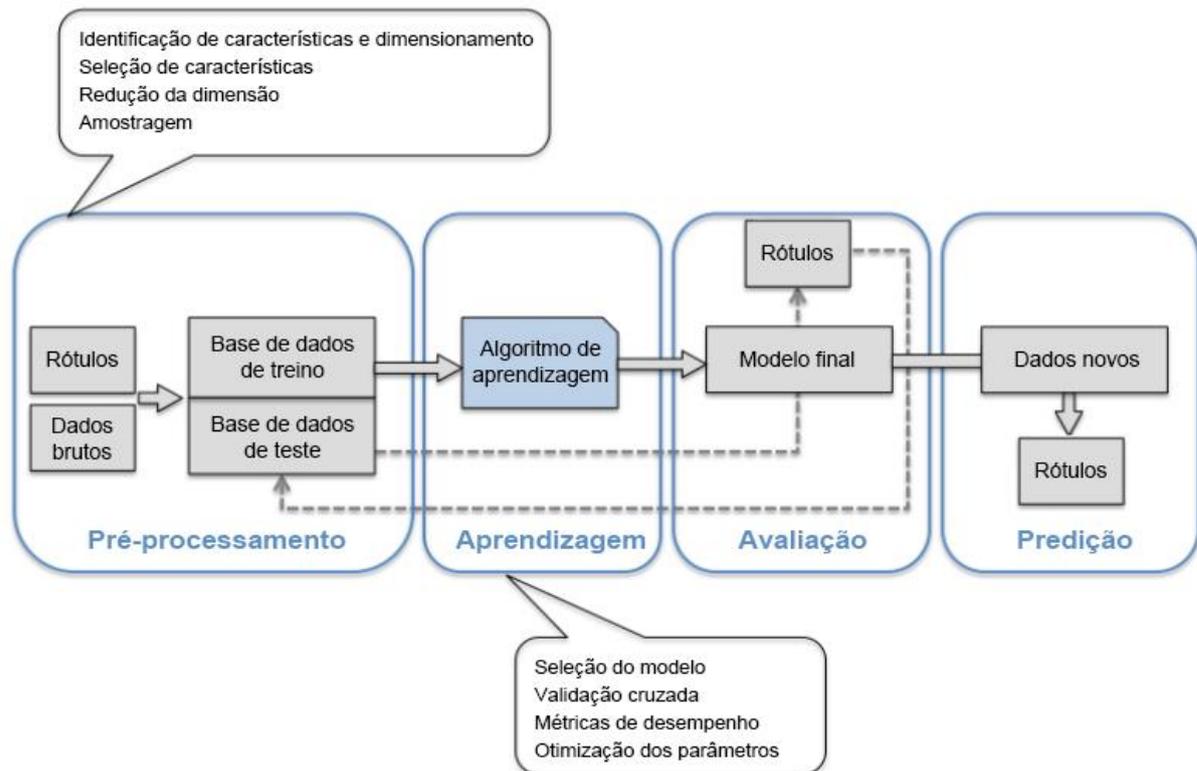


Fonte: (GOODFELLOW, BENGIO e COURVILLE, 2016), adaptado

2.4.4 Framework para Construção de Modelos

O diagrama da Figura 7 mostra um fluxograma típico utilizado na construção de modelos preditivos de *machine learning*.

Figura 7 – Fluxograma de trabalho na criação de um modelo preditivo



Fonte: (RASCHKA, 2016), adaptado

A etapa de pré-processamento é uma das mais importantes porque os dados brutos raramente estão bem formatados e estruturados. Muitos algoritmos de *machine learning* também necessitam que as características selecionadas estejam na mesma escala para fins de otimização de desempenho. Isso é comumente alcançado transformando as características em valores de 0 a 1, ou com uma distribuição com desvio padrão de média zero e variância unitária (RASCHKA, 2016).

Algumas das características selecionadas podem estar relacionadas entre si, causando um certo grau de redundância. Nesses casos, é executada uma redução da dimensão das amostras a fim de diminuir o espaço físico ocupado por elas, além de melhorar o tempo de execução do algoritmo.

3 BASE DE DADOS

A aplicação de algoritmos de *machine learning* se dá sobre um conjunto de dados, seja ele extenso (categorizando-se como *big data*) ou não (GOODFELLOW, BENGIO e COURVILLE, 2016). Nessa seção, serão apresentadas as metodologias na execução das simulações eletromagnéticas para elaboração do *dataset*. A base de dados utilizada nesse trabalho foi disponibilizada no Kaggle, uma plataforma de ciência de dados.

3.1 Metodologia

Na construção da base de dados, os parâmetros dos metamateriais e suas respectivas células unitárias foram variados em uma faixa de valores adequada, procurando respeitar os limites oferecidos pela literatura. O *software* utilizado para as simulações eletromagnéticas é o *CST Studio Lite*, utilizando o módulo *Microwave Studio*. Para o controle automatizado do *software*, utilizou-se a API³ de comunicação COM (exclusiva do *Windows*) escrita em *MATLAB*. Todo o código fonte está disponível no ANEXO I.

3.1.1 Antena Base

Para aplicação dos metamateriais, uma antena de microfita foi projetada seguindo o equacionamento apresentado na seção 2.2.1. Para isso, uma aplicação específica precisava ser escolhida para determinar a frequência de ressonância. No ano de realização desse trabalho, durante a MWC 2019 (*Mobile World Congress*), a ANATEL anunciou um leilão de 200 MHz na frequência de 3,5 GHz que será destinada para implantação da nova tecnologia de telefonia móvel denominada 5G. (HERNANDES, 2019). Na Tabela 1 são apresentados os valores obtidos de acordo com o equacionamento previamente apresentado.

Tabela 1 – Valores dos parâmetros da antena base (em milímetros)

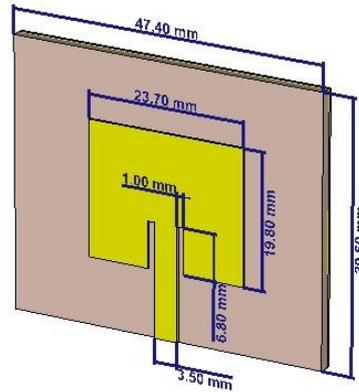
W	L	h	W₀	Y₀	g
23,7	19,8	1,7	3,5	6,8	1

³ Créditos à Symeon Symeonidis, Stefanos Tsantilas e Stelios Mitilineos.

Fonte: Autoria própria

A geometria da antena é reforçada na Figura 8. As folhas de cobre do *patch* e do plano terra são de 0,35 mm.

Figura 8 – Geometria da antena base



Fonte: Autoria própria

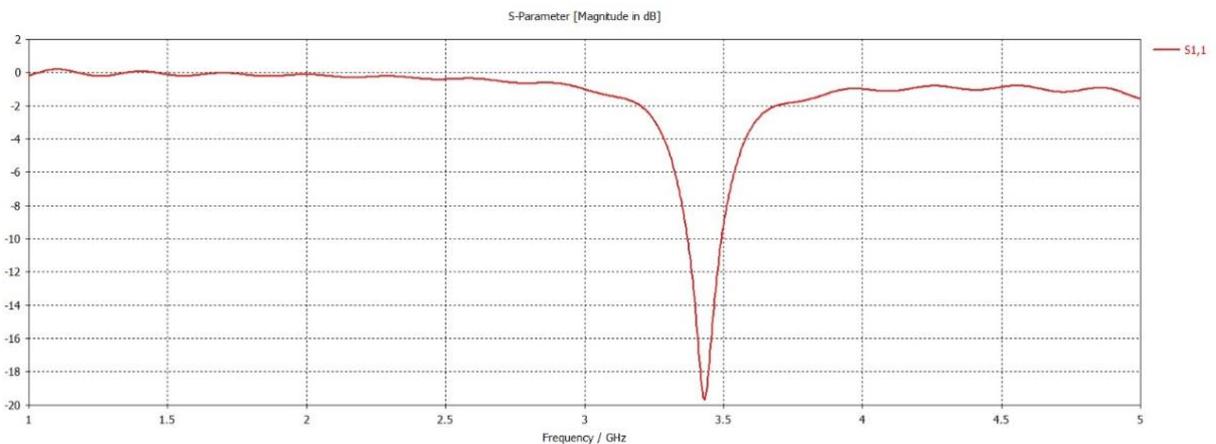
Os resultados obtidos na simulação são apresentados na Tabela 2, junto com a curva da perda por retorno na Figura 9. Esses valores irão servir como comparação às variáveis alvo (*target*) no aprendizado de máquina supervisionado.

Tabela 2 – Resultados da antena base (para $f = 3,5 \text{ GHz}$)

Resultado	Valor
Perda por retorno (S_{11})	-20 dB
Largura de banda	125,2 MHz
Ganho	3,27 dB

Fonte: Autoria própria

Figura 9 – S_{11} da antena base



Fonte: Autoria própria

3.1.2 Parametrização

Como explicado na seção 3, o modelo preditivo utiliza uma base de dados com uma grande quantidade de observações distintas. Para a construção dessa base de dados foram construídos algoritmos que automatizaram o processo de simulação das antenas metamateriais. Esses algoritmos funcionaram baseando-se na parametrização das geometrias da célula unitária e do arranjo das células, explicadas na seção 3.1.

As variáveis associadas à geometria de uma célula metamaterial SRR são mostradas na Figura 10 e detalhadas abaixo.

- W_m – Altura da célula SRR (em mm)
- W_{0m} – *Gap* dos anéis (em mm)
- d_m – Distância entre os anéis (em mm)
- t_m – Espessura dos anéis (em mm)
- X_a – Distância no eixo X do arranjo ao *patch* (em mm)
- Y_a – Distância no eixo Y entre as células do arranjo (em mm)
- *rows* – Quantidade de células no arranjo (de cada lado)

Depois de pesquisas em artigos como (SARKAR, 2017), identificou-se uma faixa de valores para essas variáveis apresentarem as características DNG das células. Esses intervalos são, em função do comprimento de onda de operação, aproximadamente:

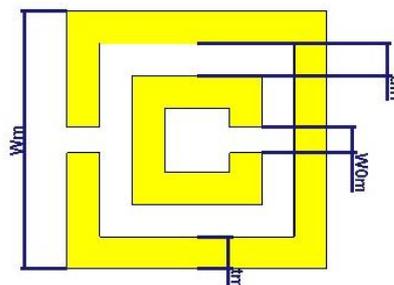
$$0,025\lambda \leq W_m \leq \frac{\lambda}{4} \quad (10)$$

$$t_m \approx 0,1W_m \quad (11)$$

$$0,0019\lambda \leq W_{0m} \leq 0,0076\lambda \quad (12)$$

$$0,0009\lambda \leq d_m \leq 0,0057\lambda \quad (13)$$

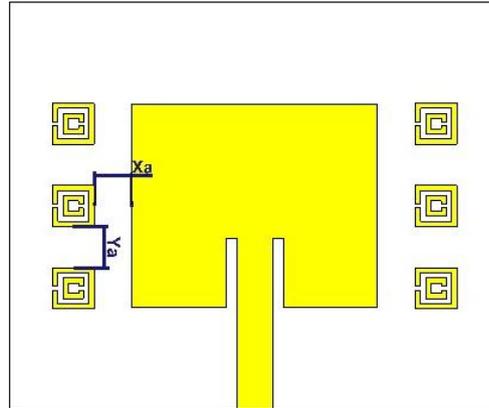
Figura 10 – Parametrização da célula SRR



Fonte: Autoria própria

De acordo com a seção 2.2.2, a alteração do comprimento do *patch* afeta diretamente a eficiência de radiação. Pensando nisso, adotou-se uma estratégia de aplicação onde o arranjo foi aplicado nas laterais do *patch*. Nessa estratégia, a referência da posição das células no arranjo se deu pelas bordas. A Figura 11 exemplifica isso.

Figura 11 – Parametrização do arranjo na aplicação lateral



Fonte: Autoria própria

De forma a abranger uma variedade de arranjos, os intervalos utilizados na disposição das células no arranjo são mostrados nas equações (14) e (15). Aqui, i representa o índice da célula, ou seja, qual posição a célula ocupa no arranjo de múltiplas linhas (nesse caso, foram utilizados arranjos de 3, 5 e 7 linhas), sempre tomando a célula central com índice 1, as superiores com índices pares e as inferiores com índices ímpares.

$$0 \leq X_a \leq \left(\frac{W}{2} - \frac{W_m}{2} \right) \quad (14)$$

$$W_m \leq Y_a \leq \left(\frac{4L}{i-1} - W_m \right) \quad (15)$$

3.2 Concatenação

Com a parametrização completa, foram programados os algoritmos para realização das simulações. O algoritmo para aplicação lateral está disponível no ANEXO I. Foi utilizado um servidor dedicado para coleta dos resultados. A concatenação desses dados foi feita utilizando *Python*, sendo que o código está disponível em github.com/renanmav/metamaterialsdataset. A base de dados final conta com 572 simulações e está disponível em kaggle.com/renanmav/metamaterial-antennas.

4 MODELAGEM

A modelagem de algoritmos de regressão é utilizada para prever resultados em uma escala contínua, tornando-a atrativa tanto na aplicação de diversas questões científicas quanto em aplicações industriais, como por exemplo, entender relacionamentos entre variáveis e identificação de tendências (RASCHKA, 2016).

A identificação de relações entre variáveis nos interessa neste estudo dado que esses *insights* ajudam na criação de um modelo matemático semelhante ao trabalho computacional de um simulador eletromagnético.

4.1 Seleção de Variáveis e Definição de Previsões

Com o conjunto de dados completo, precisamos separar as variáveis independentes que serão utilizadas no treinamento do modelo. Tendo em mente o escopo e como os dados foram obtidos, as variáveis independentes (seguindo a nomenclatura da Figura 10 e Figura 11) mostradas na Figura 12.

Figura 12 – Prévia das variáveis independentes

	Wm	W0m	dm	tm	Xa	Ya	rows
0	2142.9	162.86	77.143	214.29	0.0	2142.9	3.0
1	2142.9	162.86	77.143	214.29	3543.5	6964.3	3.0
2	2142.9	162.86	351.430	214.29	8365.0	6964.3	5.0
3	2142.9	162.86	351.430	214.29	8365.0	2142.9	3.0
4	2142.9	162.86	351.430	214.29	8365.0	6964.3	3.0
5	2142.9	162.86	351.430	214.29	8365.0	11786.0	3.0
6	2142.9	162.86	351.430	214.29	8365.0	11786.0	3.0

Fonte: Autoria própria

As previsões serão feitas com base nos *outputs* mais comuns de simuladores eletromagnéticos, sendo descritos baixo e na Figura 13.

- *gain* – Ganho da antena (em dB)
- *bandwidth* – Largura de banda (em MHz)
- *s* – Perda por retorno (em dB)

Figura 13 – Prévia das previsões

	gain	bandwidth	s
0	2.799175	110.6698	-16.090654
1	2.817305	120.7497	-12.328015
2	2.941413	123.6901	-19.313586
3	2.844078	122.4301	-19.317644
4	2.812315	124.3198	-18.459992
5	2.906383	123.4798	-14.597533
6	2.925063	114.8701	-11.407893

Fonte: Autoria própria

4.2 Limpeza e Pré-processamento dos Dados

Segundo Raschka (2016), a qualidade dos dados e a quantidade de informação útil são fatores chave que determinam o quão bem um modelo de *machine learning* pode aprender. Assim, é imprescindível examinar e pré-processar o conjunto de dados antes de alimentar o algoritmo de aprendizagem. As principais etapas desse processo são:

- Remoção e/ou inserção de valores vazios
- Transformação de dados categóricos (não-numéricos)

Há várias razões para a existência de dados vazios, como erros no processo de coleta ou algumas medições são inaplicáveis, por exemplo. Infelizmente a maioria dos algoritmos de aprendizagem não conseguem lidar com valores que estão faltando. Portanto, é crucial identificar e lidar com esses dados.

Quando verificado as informações do conjunto de dados com a função *info* do pacote *pandas*, obtemos o resultado mostrado na Figura 14. É possível ver que dentre as 572 simulações realizadas, 509 possuem *bandwidth*, ou seja, 63 simulações não tiveram uma largura de banda válida. Como a falta de largura de banda significa que a antena não ressoa (não funciona), esses 63 registros foram excluídos do conjunto de dados, representando uma perda percentual de cerca de 11%. Essa perda não é grande o suficiente para justificar a aplicação de uma segunda técnica.

Figura 14 – Informações detalhadas de cada campo do conjunto de dados

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 572 entries, 0 to 571
Data columns (total 13 columns):
Wm          572 non-null float64
W0m         572 non-null float64
dm          572 non-null float64
tm          572 non-null float64
rows        572 non-null float64
Xa          572 non-null float64
Ya          572 non-null float64
gain        572 non-null float64
vswr        572 non-null float64
bandwidth   509 non-null float64
s           572 non-null float64
pr          572 non-null float64
p0          572 non-null float64
dtypes: float64(13)
memory usage: 58.2 KB

```

Fonte: Autoria própria

Esta segunda técnica é chamada de inserção de valores vazios e consiste em adotar uma métrica para preencher os valores que estão faltando. As métricas mais adotadas são a média, a mediana ou outros. Normalmente procura-se inserir valores com média zero e desvio padrão unitário, garantindo assim a mínima (GOODFELLOW, BENGIO e COURVILLE, 2016).

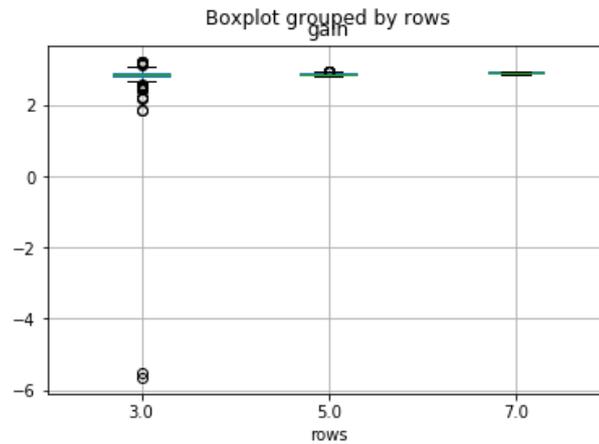
O conjunto de dados precisa ser dividido primariamente em duas partes: treino e teste. Isso é feito para evitar os *pitfalls* comentados na seção 2.4.3. O conjunto de treino é utilizado para ensinar ao algoritmo as relações entre as variáveis enquanto que o conjunto de teste é utilizado para validar o aprendizado. De acordo com Raschka (2016), esse último conjunto pode ser entendido como o último teste do modelo antes de liberá-lo para o mundo real. Foi usada a função *train_test_split* do pacote *sklearn* para criar um conjunto de treino e teste com razão 75:25, respectivamente.

4.3 Ganho

Para prever o ganho de uma antena, os recursos utilizados incluem aqueles mostrados na Figura 12, enquanto que a variável *target* será *gain*.

Plotando o ganho pelo recurso categórico *rows* é possível ver a existência de *outliers* (Figura 15), valores que destoam muito da média e possivelmente representam erros na coleta dos dados, atrapalhando a modelagem. A metodologia adotada foi a remoção desses valores do conjunto de dados.

Figura 15 – Ganho pela quantidade de células no arranjo



Fonte: Autoria própria

4.3.1 Variância

A variância dá uma medida de quanto os valores de uma função de uma variável qualquer x variam enquanto coletamos diferentes valores de x através da sua distribuição provável (GOODFELLOW, BENGIO e COURVILLE, 2016):

$$Var(f(x)) = \sum [(f(x) - \sum (f(x)))^2] \quad (16)$$

Quando a variância é baixa, os valores de $f(x)$ se aproximam dos valores esperados. A raiz quadrada da variância é conhecido como desvio padrão.

Verificando a variância dos recursos (Figura 16) é possível identificar uma dispersão de algumas variáveis, sinais claros da necessidade de sua normalização antes da modelagem.

Figura 16 – Variância dos recursos

```
Wm      3.171436e+05
W0m     3.427141e+04
dm      2.280619e+04
tm      3.171436e+03
Xa      9.805929e+06
Ya      2.821793e+07
rows    1.998020e+00
dtype: float64
```

Fonte: Autoria própria

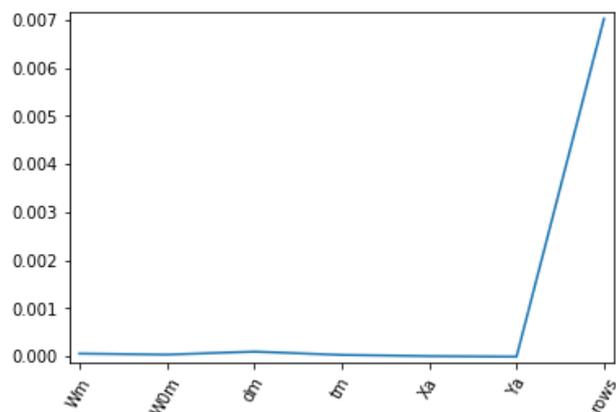
A normalização é o processo de dimensionar amostras individuais para ter uma norma de unidade. Em termos práticos, o processo de normalização está ligado com diferentes escalas de variância dos recursos. A normalização é importante em

modelos que predizem com base nos pesos dos relacionamentos formados entre os nós de dados.

4.3.2 Seleção de Recursos

Em 1996 Tibshirani introduziu um método interessante para diminuição e seleção de recursos chamado “*Least Absolute Shrinkage and Selection Operator*” (LASSO) (KIM e KIM, 2004). Este realiza previsões mais precisas por redução do que a regressão Ridge, ao mesmo tempo que realiza a seleção de recursos. A seleção de recursos nos mostra quais *features* têm mais peso na predição. Aplicando LASSO no conjunto de dados contendo os recursos pelo *target gain* obtemos a Figura 17.

Figura 17 – Verificação de peso dos recursos para o ganho



Fonte: Autoria própria

É possível identificar que o recurso *rows* é o que tem mais impacto sobre o ganho, constatando que essas duas variáveis possuem uma relação diretamente proporcional. Também é possível observar a mínima relação com W_m . Ratificando:

- Quanto mais células SRR no arranjo, maior o ganho.
- Quanto menor a altura da célula SRR, maior o ganho.

4.3.3 Treinamento do Modelo

É comum no aprendizado de máquina a construção de *pipelines*. Estes são a declaração de etapas sequenciais definidas no treinamento de um modelo. Construiu-se um *pipeline* com os seguintes passos, utilizando validação cruzada:

- Um normalizador para diminuir a variância
- Um escalonador para colocar os recursos na mesma escala

- Um estimador (ou modelo) para prever resultados

A validação cruzada é uma evolução do particionamento, consistindo na divisão do conjunto de dados em n partes, usando $n - 1$ partes para o treino e a parte remanescente para validação. Esse processo é repetido n vezes, de forma a garantir que o modelo não sofra de *overfitting* ou *underfitting*, métricas descritas na seção 2.4.3.

Os normalizadores e escalonadores são bem diretos, havendo pouca diferença entre os disponíveis da biblioteca *sklearn*, porém os modelos preditivos diferem bastante entre si de forma que a escolha do algoritmo é um dos pontos chave da modelagem. O modelo de regressão escolhido é baseado no *k-nearest neighbors* (ou KNN), tendo um modelo similar disponível para classificação. A estratégia utilizada em KNN é a atribuição de pesos baseado na distância euclidiana entre os seus pontos vizinhos. Essa estratégia pareceu adequada dado que o problema atual é espacial. (KELLER, GRAY e GIVENS, 1985) apresentam um algoritmo similar ao adotado por *KNeighborsRegressor*. O algoritmo completo escrito em *Python* com comentários do autor está disponível em <https://bit.ly/antennaml>.

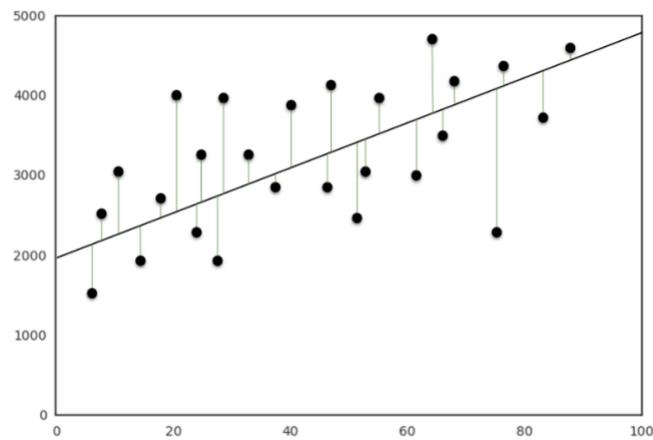
4.4 Perda por Retorno e Largura de Banda

A metodologia utilizada para prever a perda por retorno e a largura de banda não difere da apresentada na seção 4.3 que foi utilizada para prever o ganho. Em ambos foram utilizados o mesmo algoritmo KNN, com mesmas etapas nas *pipelines*, apenas diferenciando a variável *target* e os parâmetros do estimador.

5 RESULTADOS

Como a tarefa de regressão nada mais é que identificar uma função que melhor se adequa aos dados (na regressão linear, essa função é uma linha), a avaliação de desempenho do modelo é feita através do quanto os valores previstos destoam dos reais. Assim, o erro é a somatória da distância entre esses dois pontos, ilustrada na Figura 18. Essa distância é normalmente denominada “resíduos”.

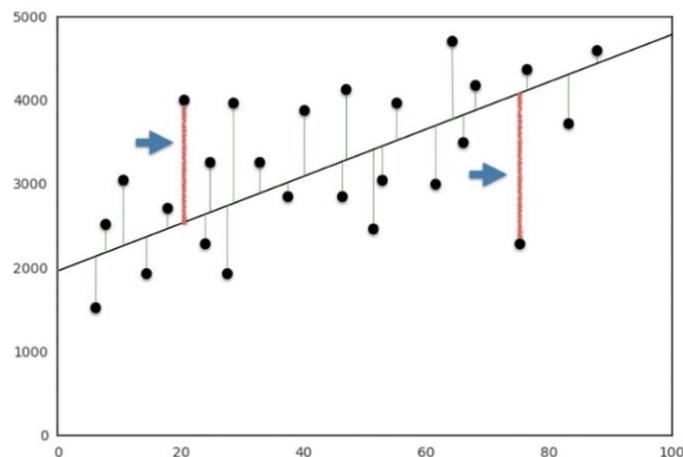
Figura 18 – Somatória dos erros na regressão linear



Fonte: Autoria própria

Entretanto, a métrica da somatória dos resíduos pode não ser a mais eficaz, já que um resíduo positivo pode cancelar um negativo, conforme mostrado na Figura 19. Dessa forma, a melhor métrica é a somatória dos quadrados dos resíduos. Essa é a função peso que queremos minimizar para obter máximo desempenho. De fato, o processo de aprendizagem nada mais é que a minimização de uma função peso.

Figura 19 – Problema na somatória dos resíduos



Fonte: Autoria própria

5.1 Modelos

A melhor precisão obtida com o modelo treinado para prever o ganho foi de 77,08% com somente um vizinho. A média dos quadrados dos resíduos é de $3,6 \times 10^{-3}$ dB, um erro baixíssimo.

A melhor precisão obtida com o modelo treinado para prever a perda por retorno foi de 94,38% com dois vizinhos. A média dos quadrados dos resíduos é de 2,45 MHz, um erro ainda sim baixo. Pode-se dizer que esse foi o melhor modelo construído, podendo concluir que esse modelo desempenha de forma excelente.

A melhor precisão obtida com o modelo treinado para prever a largura de banda foi de 87,81% com dois vizinhos. A média dos quadrados dos resíduos é de 15 dB, um erro consideravelmente alto. Conclui-se que o modelo possui boa precisão, mas um alto ruído.

5.2 Comparação

Afim de validar os resultados obtidos na seção anterior, vamos comparar a simulação de uma dada antena que está nos limites do escopo da modelagem, ou seja, uma antena de microfita de patch retangular com células SRR laterais, como mostrado na Figura 11. Serão comparados os resultados gerados pelos modelos com os resultados da simulação eletromagnética, sendo esta a referência. Também serão comparados os tempos de execução. A antena escolhida tem as dimensões dadas na Tabela 3, sendo ilustrada na Figura 20.

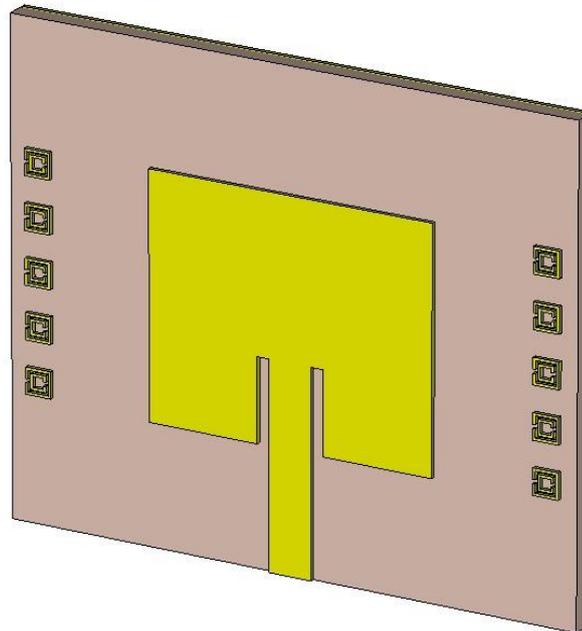
Tabela 3 – Dimensões da antena utilizada na comparação

Dimensão	Valor (em mm)
Comprimento do <i>patch</i> (W)	23,7
Largura do <i>patch</i> (L)	19,8
Altura do substrato (h)	1,7
Comprimento da linha de alimentação (W0)	3,5
Comprimento da entrada da linha (Y0)	6,8
Largura da entrada da linha (g)	1
Altura das folhas de cobre (t)	0,35
Largura e comprimento da célula SRR (Wm)	2,14

Gap entre os anéis (W0m)	0,33
Distância interna entre os anéis (dm)	0,21
Espessura dos anéis (tm)	0,21
Número de células SRR por lado ⁴	5
Distância entre o patch e o arranjo (Xa)	8,36
Distância entre as células SRR no arranjo (Ya)	2,14

Fonte: Autoria própria

Figura 20 – Antena utilizada na comparação



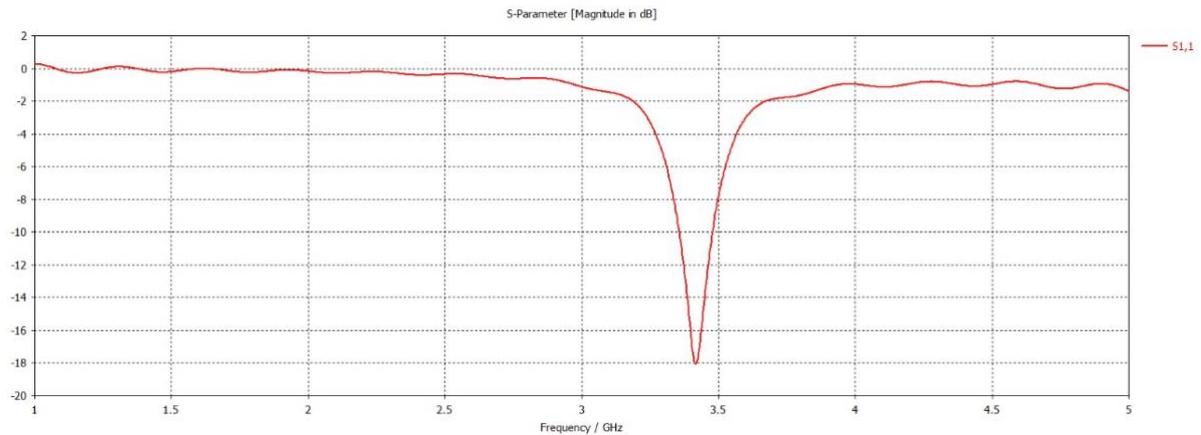
Fonte: Autoria própria

Os resultados obtidos na simulação eletromagnética são apresentados na Tabela 4, junto com a curva da perda por retorno na Figura 21. O tempo de execução é mostrado no gráfico da Figura 22, constando o tempo de simulação, de treino e de predição. Observe que o eixo y está em escala logarítmica.

Tabela 4 – Resultados da antena de comparação (para $f = 3,5 \text{ GHz}$)

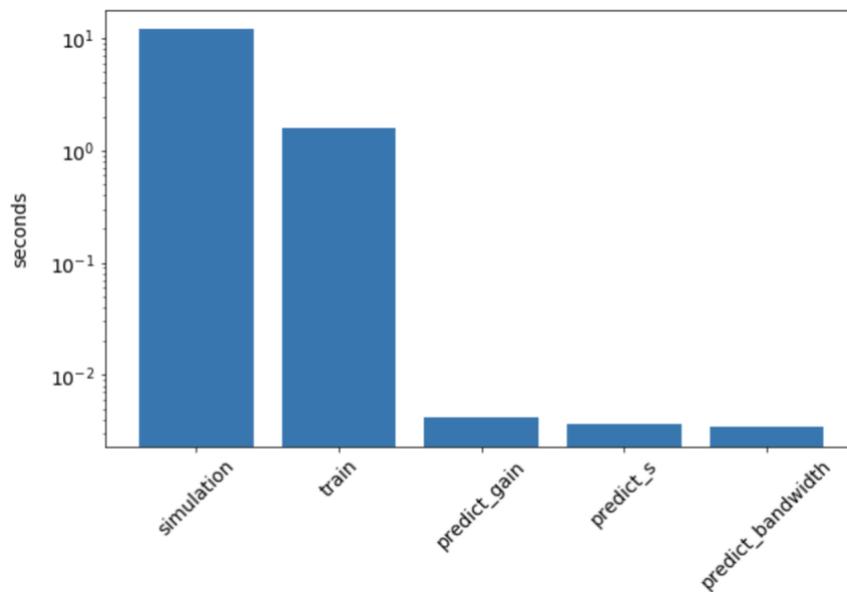
Resultado	Simulado	Modelado	Erro (%)
Perda por retorno (S_{11})	-18,1 dB	-17,7 dB	2,2
Largura de banda	130 MHz	121,9 MHz	6,23
Ganho	2,8 dB	2,81 dB	0,35

Fonte: Autoria própria

Figura 21 – S₁₁ da antena de comparação

Fonte: Autoria própria

Figura 22 – Tempos de execução



Fonte: Autoria própria

Pode-se observar que o tempo de execução de uma simulação é de cerca de 12 segundos, enquanto que o treinamento do modelo levou aproximadamente 1,6 segundos. Se forem utilizados modelos já treinados, o tempo de predição do ganho, perda por retorno e largura de banda são, respectivamente, 4,18, 3,71 e 3,48 milissegundos.

6 CONCLUSÃO

A partir dos resultados apresentados na Tabela 4 e na Figura 22, é possível concluir que o erro da metodologia de *design* com *machine learning* é menor do que 10%, se tornando adequada já que o tempo de execução de um modelo treinado é cerca de 3000 vezes menor. Isso representa uma maior quantidade de tentativas por tempo investido em otimização. Entretanto, a metodologia apresentada está limitada ao escopo da modelagem, ou seja, há a necessidade de se criar outros conjuntos de dados com simulações em outras frequências e/ou utilizando outras geometrias de células metamateriais. Isso deve ser feito a fim de aumentar a precisão dos algoritmos. Uma vez feita a modelagem, engenheiros podem basear-se nesses modelos para otimizar antenas ou desenvolver melhorias para os modelos existentes.

REFERÊNCIAS BIBLIOGRÁFICAS

BALANIS, C. A. **Antenna Theory: Analysis and Design**. 3^a. ed. Hoboken, New Jersey: John Wiley & Sons Inc., 2005.

GIANFAGNA, C. et al. **Enabling Antenna Design with Nano-Magnetic Materials using Machine Learning**. IEEE Nanotechnology Materials and Devices Conference. Anchorage, AK, USA: IEEE. 2015. p. 5.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Massachusetts: MIT Press, 2016.

GUELBER, E. F. et al. **A Modified Ultra Wide Band Antenna With Metamaterial Patterns**. PRZEGLĄD ELEKTROTECHNICZNY. Poland: Sławomir Tumański. 2016. p. 166-169.

HERNANDES, R. Leilão do 5G deve ser feito até março de 2020, diz presidente da Anatel. **Folha de São Paulo**, 2019. Disponível em: <<https://www1.folha.uol.com.br/tec/2019/02/leilao-do-5g-deve-ser-feito-ate-marco-de-2020-diz-presidente-da-anatel.shtml>>. Acesso em: 23 Maio 2019.

KELLER, J. M.; GRAY, M. R.; GIVENS, J. A. J. A Fuzzy K-Nearest Neighbor Algorithm. **IEEE Transactions on systems, man, and cybernetics**, SMC-15, Agosto 1985. 580-585.

KIM, Y.; KIM, J. **Gradient LASSO for feature selection**. Seoul National University. Seoul. 2004.

KUSHWAH, V. S.; TOMAR, G. S. Design and Analysis of Microstrip Patch Antennas Using Artificial Neural Network. In: CHATTOPADHYAY, S. **Trends in Research on Microstrip Antennas**. India: IntechOpen, 2017. Cap. 3, p. 55-75.

MARQUÉS, R.; MARTÍN, F. Split Ring Resonators and Related Topologies. In: CAPOLINO, F. **Theory and Phenomena of Metamaterials (Metamaterials Handbook)**. 1^a. ed. New York: CRC Press, v. III, 2009. Cap. 16, p. 353-370.

OZBAY, E.; AYDIN, K. Composite Metamaterials, Negative Refraction and Focusing. In: CAPOLINO, F. **Theory and Phenomena of Metamaterials (Metamaterials Handbook)**. 1^a. ed. New York: CRC Press, v. III, 2009. Cap. 18, p. 389-395.

POZAR, D. M. **Microstrip Antennas**. Proceedings of the IEEE. Massachusetts: IEEE. Janeiro 1992. p. 79-91.

RASCHKA, S. **Python Machine Learning**. 1^a. ed. Birmingham, UK: Packt Publishing Ltd., 2016.

SARKAR, S. B. **Design and Analysis of 5.2 GHz Rectangular Microstrip Patch Array Antenna using Split Ring Resonator**. 4th International Conference on Signal Processing, Communications and Networking. Chennai, India: IEEE. 2017.

THALLURI, L. N. et al. **Structural Analysis and Design of Microstrip Antenna for S-Band Applications**. International Conference on Control, Instrumentation, Communication and Computational Technologies. India: IEEE. 2016. p. 397-399.

WILTSHIRE, M. C. K. RF Metamaterials. In: CAPOLINO, F. **Theory and Phenomena of Metamaterials (Metamaterials Handbook)**. 1^a. ed. New York: CRC Press, v. II, 2009. Cap. 5, p. 300-327.

ANEXO I – LIVE SCRIPT PARA APLICAÇÃO LATERAL DE CÉLULAS SRR

Este é um *live script* em MATLAB que visa construir um *dataset* de simulações de antenas com metamateriais para treinar modelos preditivos de *machine learning*. Os parâmetros dos metamateriais, como comprimento da célula, disposição do arranjo, e outros, serão variados em uma faixa de valores adequados, procurando respeitar as referências oferecidas pela literatura. O *software* utilizado para as simulações eletromagnéticas é o **CST Studio Lite**®, utilizando o módulo **Microwave Studio**. Para o controle automatizado do software, será utilizada a API desenvolvida por Symeon Symeonidis, Stefanos Tsantilas e Stelios Mitilineos, tendo o código fonte e a documentação disponíveis [aqui](#). A API original foi modificada e está disponível no repositório.

1.1 Inicialização do CST

Para correta utilização da API, antes é necessário adicionar seu código fonte ao *path*, dessa forma:

```
addpath(genpath('CST-MATLAB-API-master'));
```

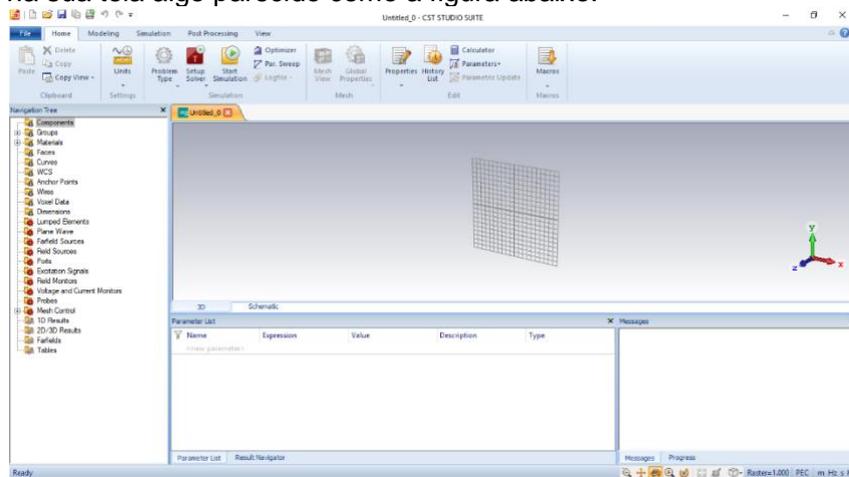
Para iniciar a aplicação do CST, utilizamos o seguinte comando:

```
cst = actxserver('CSTStudio.application');
```

Agora, com o CST aberto, podemos inicializar o Microwave Studio.

```
mws = cst.invoke('NewMWS');
```

Você deve ter na sua tela algo parecido como a figura abaixo.



1.2 Configurando o ambiente de simulação

Na MCW (*Mobile World Congress*) de 2019, o presidente da ANATEL, Leonardo de Moraes, anunciou o leilão de 200 MHz na frequência de 3,5 GHz que serão destinados para implantação do 5G no Brasil. Aqui, a frequência será esse domínio reservado na frequência de 3.5GHz que a ANATEL anunciou.

Inicialmente, precisamos definir as unidades padrão. É importante ressaltar que a **unidade de comprimento é micrômetros** (1e-6 metros). Com isso, também é necessário definir o intervalo de frequência que será simulado, aqui serão simulados 3% da frequência central como limites inferior e superior. Assim, é possível inicializar a *mesh* e definir as condições de contorno.

```
fcenter = 3.5;
```

```
fmin = fcenter * 0.97
```

```
fmin = 3.3950
```

```
fmax = fcenter * 1.03
```

```
fmax = 3.6050
```

Em código:

```
CstDefineUnits(mws, 'um', 'GHz', 's', 'Kelvin', 'V', 'A', 'Ohm', 'S', 'PikoF', 'NanoH');
```

```
CstDefineFrequencyRange(mws, fmin, fmax);
```

```
CstMeshInitiator(mws);
```

A antena estará no espaço livre, então:

```
%Condições de contorno
```

```
Xmin = 'expanded open';
```

```
Xmax = 'expanded open';
```

```
Ymin = 'expanded open';
```

```
Ymax = 'expanded open';
```

```
Zmin = 'expanded open';
```

```
Zmax = 'expanded open';
```

```
CstDefineOpenBoundary(mws, fmin, Xmin, Xmax, Ymin, Ymax, Zmin, Zmax);
```

Em seguida, precisamos definir o material do fundo:

```
XminSpace = 0;
```

```
XmaxSpace = 0;
```

```
YminSpace = 0;
```

```
YmaxSpace = 0;
```

```
ZminSpace = 0;
```

```
ZmaxSpace = 0;
```

```
CstDefineBackgroundMaterial(mws, XminSpace, XmaxSpace, YminSpace, YmaxSpace, ZminSpace, ZmaxSpace);
```

1.3 Definição de materiais

Para a definição de materiais que serão utilizados nos componentes, é preciso definir as propriedades do material de acordo com a biblioteca do CST e fazer uma função em MATLAB com todas essas propriedades, juntamente com o nome do material. Somente depois disso é que podemos utilizar o nome do material no código. Na API, já temos funções para definir materiais como cobre e FR4 com perdas.

```
metal = CstCopperAnnealedLossy(mws)
```

```
metal = 'Copper (annealed)'
```

```
[dielectric, epsilon_r] = CstFR4lossy(mws)
```

```
dielectric = 'FR-4 (lossy)'
```

```
epsilon_r = 4.3000
```

Além disso, precisamos definir a altura da folha de metal (nesse caso, cobre).

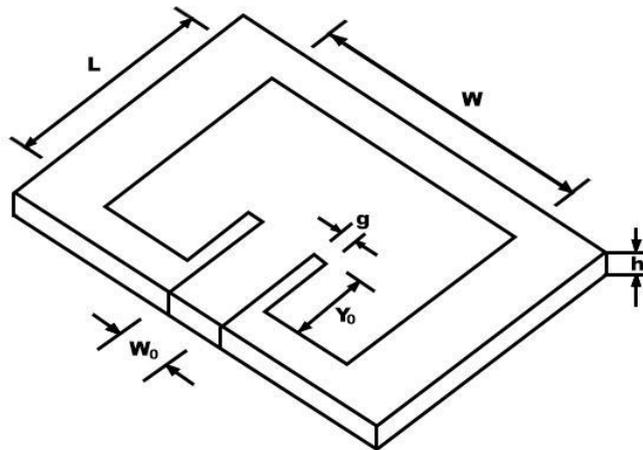
```
t = 0.035*1e3; % in um (t from thickness)
```

1.4 Design da antena base

Na criação do *dataset*, os metamateriais serão aplicados em uma antena base que possui bons parâmetros. Essa antena será projetada com base em toda a referência bibliográfica existente. No *design* de antenas, é necessário definir primariamente o material do substrato e a frequência de aplicação. A altura do substrato também precisa ser predefinida com antecedência. A bibliografia sugere uma escolha entre $0,03\lambda$ e $0,5\lambda$, tendo uma preferência por substratos mais finos por resultarem em uma melhor eficiência de radiação. Para esse projeto:

```
h = 1700; % in um (1e-6 meters)
```

As variáveis associadas a geometria de uma antena *microstrip* são essas mostradas na figura abaixo.



Para uma obtermos uma boa eficiência de radiação, (BALANIS, 2005) nos dá que:

$$W = \frac{c}{2f_r} \sqrt{\frac{2}{\epsilon_r + 1}}$$

$$W = 0.9 * 3e8 / (2 * f_{center} * 1e9) * \sqrt{2 / (\epsilon_r + 1)} * 1e6 \text{ \% in um}$$

$$W = 2.3694e+04$$

Antes de definirmos L , precisamos calcular a constante dielétrica efetiva da antena, já que algumas ondas viajam no substrato e algumas no ar. Segundo Balanis (2005):

$$\epsilon_{\text{reff}} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left(1 + 12 \frac{h}{W}\right)^{-\frac{1}{2}}$$

$$\epsilon_{\text{psilon_reff}} = (\epsilon_{\text{psilon_r}} + 1) / 2 + ((\epsilon_{\text{psilon_r}} - 1) / 2) * (1 + (12 / W) * h)^{(-1/2)}$$

$$\epsilon_{\text{psilon_reff}} = 3.8595$$

Agora que temos a constante dielétrica efetiva, calculamos o quão maior o patch aparenta ser do que realmente é através de:

$$\Delta L = 0,412h \frac{(\epsilon_{\text{reff}} + 0,3) \left(\frac{W}{h} + 0,264\right)}{(\epsilon_{\text{reff}} - 0,258) \left(\frac{W}{h} + 0,8\right)}$$

$$\text{delta_l} = 0.412 * h * ((\epsilon_{\text{psilon_reff}} + 0.3) * (W / h + 0.264)) / ((\epsilon_{\text{psilon_reff}} - 0.258) * (W / h + 0.8))$$

$$\text{delta_l} = 779.4965$$

Finalmente, temos que:

$$L = \frac{c}{2f_r \sqrt{\epsilon_{\text{reff}}}} - 2\Delta L$$

$$l = 0.98 * 3e5 / (2 * f_{center} * \sqrt{\epsilon_{\text{psilon_reff}}}) - 2 * \text{delta_l}$$

$$l = 1.9820e+04$$

Por último, os parâmetros restantes normalmente são:

$$Y_0 = 4h$$

$$g \approx 1000 \text{ um}$$

$$W_0 = 3,5g$$

$$Y_0 = 4 * h$$

$$Y_0 = 6800$$

$$g = 1000;$$

$$W_0 = 3.5 * g$$

$W_0 = 3500$

Para refletir essas medidas no CST, usamos:

```
DrawMicrostripAntenna(mws, W, l, t, h, Y0, W0, g, metal, dielectric);
DrawPort(mws, W, l, t, h, W0);
```

1.5 Simulação da antena base

Vamos agora definir o monitor de campo distante (*farfield*) na frequência de ressonância. Aqui um passo manual precisa ser dado, já que os *templates* de pós-processamento são uma *feature* relativamente nova, o controle VBA desse componente não está disponível na API.

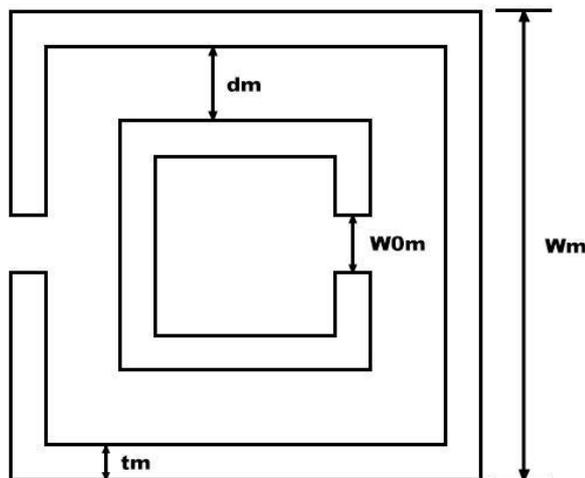
Para criar esse *template*, adicione um *breakpoint* (clique sobre o número da linha) na linha comentada para parar a execução do código. Abra o CST no menu superior, selecione a aba **Post Processing > Result Templates > Template Based Post Processing**. Na primeira caixa, selecione **Farfield and Antenna Properties** e na segunda **Farfield Result**. Selecione o *template* **Max Gain over Frequency** e renomeie-o para **Gain**, clicando duas vezes sobre o item. Após isso, clique em **Close** e continue com a execução do código.

Não se esqueça de definir o caminho onde serão exportados os resultados modificando a variável *base_path*.

```
base_path = 'C:\Users\Renan\results'
base_path = 'C:\Users\Renan\results'
CstDefineFarfieldMonitor(mws, strcat('farfield      (f=', num2str(fcenter), ')'),
fcenter);
%Create template for Gain vs Frequency (name it Gain)
CstDefineTimedomainSolver(mws, -40);
ExportResults(mws, base_path, 0, W, l, t, h, Y0, W0, g);
CstSaveProject(mws, strcat(base_path, '\0\simulacao.cst'));
CstQuitProject(mws);
```

1.6 Intervalo de valores para a célula metamaterial

As variáveis associadas a geometria de uma célula metamaterial quadrada são mostradas na figura abaixo.



De acordo com a literatura, as células metamateriais apresentam suas características DNG (Double Negative) para intervalos de:

$$0,025\lambda \leq W_m \leq \frac{\lambda}{4}$$

$$t_m \approx 0,1W_m$$

$$1,9 \times 10^{-3}\lambda \leq W_{0m} \leq 7,6 \times 10^{-3}\lambda$$

$$0,9 \times 10^{-3}\lambda \leq d_m \leq 5,7 \times 10^{-3}\lambda$$

Alguns intervalos estão em função do comprimento de onda λ , de forma que:

$$\lambda = \frac{c}{f}$$

```
lambda = 3e5./fcenter % in um (1e-6 m)
lambda = 8.5714e+04
n_Wm = 5;
Wm_min = 0.025*lambda;
Wm_max = 1/4*lambda;
Wm_step = (Wm_max-Wm_min)/(n_Wm-1);
Wm = Wm_min:Wm_step:Wm_max
Wm = 1x5
10^4 x
    0.2143    0.6964    1.1786    1.6607    2.1429
tm = 0.1.*Wm
tm = 1x5
10^3 x
    0.2143    0.6964    1.1786    1.6607    2.1429
n_W0m = 4;
W0m_min = 1.9e-3*lambda;
W0m_max = 7.6e-3*lambda;
W0m_step = (W0m_max-W0m_min)/(n_W0m-1);
W0m = W0m_min:W0m_step:W0m_max
W0m = 1x4
    162.8571    325.7143    488.5714    651.4286
n_dm = 4;
dm_min = 0.9e-3*lambda;
dm_max = 5.7e-3*lambda;
dm_step = (dm_max-dm_min)/(n_dm-1);
dm = dm_min:dm_step:dm_max
dm = 1x4
    77.1429    214.2857    351.4286    488.5714
Até agora, o número total de simulações é:
n_sim = n_Wm*n_W0m*n_dm
n_sim = 80
```

1.7 Intervalo de valores pro arranjo

Primeiramente, vamos trabalhar com arranjos ímpares de colunas, ou seja:

```
rows = 3:2:7;
```

A distância em X entre o arranjo e o *patch* X_a (contando da borda) será:

$$0 \leq X_a \leq \left(\frac{W}{2} - \frac{W_m}{2} \right)$$

```
n_Xa = 2;
Xa = zeros(length(Wm), n_Xa);
for i = 1:length(Wm)
    Xa_min = 0;
    Xa_max = W/2-Wm(i)/2;
    Xa_step = (Xa_max-Xa_min)/(n_Xa-1);
```

```

    Xa(i,:) = Xa_min:Xa_step:Xa_max;
end
Xa = Flat(Xa)
Xa = 6x1
104 x
    0
    0.1133
    0.3544
    0.5954
    0.8365
    1.0776

```

A distância em Y entre as células do arranjo Y_a (contando da borda) será:

$$W_m \leq Y_a \leq \left(\frac{4L}{\text{rows} - 1} - W_m \right)$$

```

n_Ya = 2;
Ya = zeros(length(Wm), n_Ya);
for i = 1:length(Wm)
    Ya_min = Wm(i);
    Ya_max = 4*l/(min(rows) - 1) - Wm(i);
    Ya_step = (Ya_max - Ya_min)/(n_Ya - 1);

    Ya(i,:) = Ya_min:Ya_step:Ya_max;
end
Ya = Flat(Ya);
Ya = Ya(Ya >= 0) % get only positive values
Ya = 10x1
104 x
    0.2143
    0.6964
    1.1786
    1.6607
    1.8211
    2.1429
    2.3032
    2.7854
    3.2675
    3.7497

```

Simulação

Para verificar quais combinações de geometria são possíveis, é necessário fazer uma verificação 1:1.

```

% Unit cell vars
% Wm and tm
% W0m
% dm
possible_geometry = [];
count = 1;
for wm = 1:length(Wm)
    for w0m = 1:length(W0m)
        for Dm = 1:length(dm)

                % Cells line-up vars

```



```

Xa = possible_geometry(i,6);
Ya = possible_geometry(i,7);

if cell == 1
    ya = 0;
elseif mod(cell,2) == 0
    ya = (cell/2)*(Ya+Wm);
else
    ya = (-1)*((cell-1)/2)*(Ya+Wm);
end

xa = (W/2+Xa+Wm/2)*side;

DrawSquareSRR(mws,Wm,W0m,dm,tm,hm,h0,metal,cell*side,xa,ya)
end
end

CstDefineFarfieldMonitor(mws,strcat('farfield (f=',num2str(fcenter),)'),
fcenter);

CstDefineTimedomainSolver(mws,-40);

ExportResultsMeta(mws, 'C:\Users\Renan\results', count,
Wm,W0m,dm,tm,hm,h0,cell,Xa,Ya);

CstSaveAsProject(mws,strcat('C:\Users\Renan\results\',num2str(count),'\simulac
ao.cst'));
CstQuitProject(mws);

count = count + 1;
end

```