



Instituto Tecnológico de Aeronáutica

Aplicativo web para casamento de toco simples

André Marcello Soto Riva Figueira
Daniel Prince Cerneiro
Dylan N. Sugimoto
Gabriel Adriano de Melo

São José dos Campos
23/05/2017

Introdução

Este projeto tem como objetivo a elaboração de um programa que realize os cálculos para casamento de uma linha de transmissão com uma carga pelo método do toco simples e que tenha uma interface amigável tendo em vista a grande dificuldade que é realizar esses cálculos manualmente, e o fato de existir poucos aplicativos que desempenham a mesma funcionalidade.

O objetivo final do casamento do toco simples é obter a partir da impedância intrínseca do circuito, da impedância, da frequência de casamento e das propriedades eletromagnéticas do material (permissividade elétrica e permeabilidade magnética) o comprimento do toco e a distância que esse deve ser colocado em paralelo para que não haja ondas refletidas. Para isso, primeiramente calcula-se a impedância normalizada de carga:

$$z_L = \frac{Z_L}{Z_0} \quad (1),$$

em que Z_L é a impedância de carga, Z_0 é impedância intrínseca do circuito e z_L é impedância normalizada de carga.

Em seguida, calcula-se a admitância normalizada de carga:

$$y_L = \frac{1}{z_L} \quad (2),$$

em que z_L é a impedância normalizada de carga e y_L é a admitância normalizada de carga.

Calculada a admitância de carga, substitui-se o seu valor na equação (3) e calcula-se a expressão da admitância da linha em função da posição e realiza-se a manipulação algébrica necessária para deixar a expressão na sua forma complexa algébrica.

$$y(x) = \frac{y_L + j \tan(\beta x)}{1 + y_L \tan(\beta x)} \quad (3),$$

em que $y(x)$ é admitância da linha em função de x , que é a posição; y_L é a admitância de carga e β é número de onda.

Após o cálculo da admitância da linha, calcula-se para qual valor de posição que a parte real da admitância da linha assume valor unitário. Esse valor de posição é a posição em que o toco deve ser inserido em paralelo ao circuito. Em seguida, calcula-se o valor da parte imaginária da admitância da linha para o valor de posição encontrado anteriormente que é o valor da admitância que o toco deve subtrair para que haja o casamento; logo, a partir desse valor calcula-se o

comprimento do toco utilizando a equação (4), se o toco estiver em aberto ou a equação (5), se o toco estiver em curto. Após esses cálculos é possível calcular o V_{swr} para outras frequências calculando, para a posição do toco, a posição elétrica correspondente para a nova frequência e com a posição elétrica calcular a nova admitância da linha usando a equação (3). Calculado a admitância da linha, soma-se esse valor ao novo valor da admitância do toco que foi calculado para o valor de comprimento elétrico do toco para a nova frequência. Com isso, calcula-se o valor do coeficiente de reflexão pela equação (6) a partir do qual se obtém, o valor do V_{swr} usando a equação (7).

$$y_{toco} = j\tan(\beta l) \quad (4),$$

em que y_{toco} é a admitância do toco, l é o comprimento do toco e β é o número de onda.

$$y_{toco} = -j\cotan(\beta l) \quad (5),$$

em que y_{toco} é a admitância do toco, l é o comprimento do toco e β é o número de onda.

$$|\Gamma| = \frac{|1 - y(x)|}{|1 + y(x)|} \quad (6),$$

em que Γ é o coeficiente de reflexão e y é admitância da linha em função de x .

$$V_{swr} = \frac{1 + |\Gamma|}{1 - |\Gamma|} \quad (7),$$

em que Γ é o coeficiente de reflexão e V_{swr} é taxa de voltagem da onda estacionário (Voltage Standing Wave Ratio).

Esse são os passos teóricos que o programa desse projeto automatizou na forma de um aplicativo *Web Browser* para ter uma interface mais amigável para o usuário, de forma a se tornar uma ferramenta interessante na área de ensino de engenharia eletrônica, em especial, no estudo das linhas de transmissão.

A escolha da plataforma web, sugestão inicial do nosso grupo, se deu pela acessibilidade e pela compatibilidade de tal plataforma, que pode ser acessada livremente e por qualquer sistema operacional, inclusive em tablets e smartphones que possuam navegador web.

Descrição do Algoritmo

Da linha 115 até 126 do código 3 no Anexo, os valores de inputs de entrada são transferidos para variáveis. Em seguida é calculado a impedância normalizada da carga na linha 128, e admitância normalizada da carga na linha 130. Da linha 177 até a linha 198, os cálculos necessários para se obter o valor da posição do toco, que é obtido nas linhas 197 e 198, são realizados. Após calculado a posição do toco, é calculado o comprimento do toco da linha 191 até a linha 208.

Após calculado o valor do comprimento do toco e da posição do toco, são realizados os cálculos para se obter o valor de V_{SWR} para a frequência máxima e mínima definido pela banda de largura de frequência (input BW), da linha 211 até 235, em que é utilizado as funções calcularNovoyToco (linha 30 até 35), que calcula o valor da admitância para um dado valor de comprimento do toco e frequência de onda; calcularModR (linha 51 até 63), que calcula o valor do módulo do coeficiente de reflexão para um dado valor de admitância do toco, posição do toco, frequência de onda e admitância de carga; e calcular V_{SWR} , que calcula o valor de V_{SWR} para um dado valor de módulo do coeficiente de reflexão. Por fim, nas linhas 238 até 274, os resultados encontrados são formatados para serem exibidos na tela.

Para a exibição da carta de Smith, utiliza-se primeiro uma imagem de fundo fixa, e com posições do seu centro e raio conhecidas, e faz-se uma transformação de coordenadas para que o ponto a ser desenhado na carta seja correspondente às marcações da imagem.

Desta forma, determinou-se a transformação de coordenadas a partir do cálculo do coeficiente de reflexão, uma vez que sistemas de coordenadas para o coeficiente de reflexão é o cartesiano, com o centro na Carta de Smith, sendo apenas necessário a computação de um fator de escala para atender as dimensões da figura.

Para facilitar o seu uso para os próximos laboratórios, utilizou-se o paradigma da orientação a objeto para encapsular o código da Carta de Smith, facilitando a sua reutilização e a verificação de erros. Assim, ao criar uma nova Carta de Smith, o programador deve passar como parâmetros o centro da imagem de fundo da Carta de Smith, o seu raio, a referência à imagem e a referência da área de desenho.

O algoritmo realiza sucessivas iterações para o cálculo de valores de SWR relacionados a frequências da banda de operação especificada. Para a exibição do gráfico de SWR utilizou-se a biblioteca plotly.js, que utiliza a lista de pontos de SWR gerados anteriormente.

Utilização

O aplicativo web é de fácil utilização e de grande portabilidade. Para que o usuário utilize-o, basta inserir os valores nas caixas de texto e pressionar o botão de “Calcular”. Os resultados são adicionados na página, representados pelo texto, pelas imagens da Carta de Smith (de impedância) e pelos gráficos.

Para refazer o cálculo, basta pressionar novamente o botão de “Calcular” e todos os resultados serão atualizados.

Como observado na Figura 1, o usuário deverá digitar a impedância da linha de transmissão “ Z_0 ”, a impedância da carga “ Z_L ”, com a sua parte real e imaginária. É importante notar que as unidades de impedância, apesar de serem em Ohms por padrão, podem ser qualquer unidade (desde que ambos na mesma unidade), uma vez que é a impedância normalizada da carga que será utilizada para os cálculos.

No campo de “Frequência de Casamento (MHz)”, o usuário deverá digitar a frequência para o qual o casamento será realizado, e no campo de “BW de operação (MHz)”, a largura de banda para o cálculo do SWR. Nos campos ϵ_{rel} e U_{rel} , o usuário deverá inserir a permissividade elétrica relativa e a permeabilidade magnética relativa da linha de transmissão.

Lab 1 - ELE-12 - Casamento de impedância com toco simples em paralelo

Especifique as entradas para o cálculo do casamento com toco simples em paralelo

Z_0	<input type="text" value="50"/>	
Z_L	<input type="text" value="60"/>	+ j <input type="text" value="30"/>
Frequência de Casamento (MHz)	<input type="text" value="2000"/>	
U_{rel}	<input type="text" value="1"/>	
ϵ_{rel}	<input type="text" value="1"/>	
BW de operação (MHz)	<input type="text" value="100"/>	

Clique em calcular para obter as respostas.

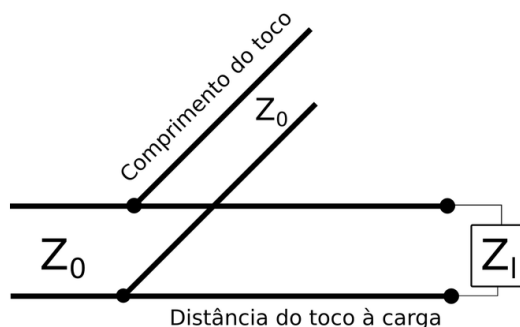


Figura 1 – Interface de entrada dos valores para o casamento.

Assim, após pressionar o botão de calcular, os resultados serão inseridos na mesma página e o usuário poderá rolar a tela para poder visualizá-los. Em primeiro lugar, é disposto as informações da primeira solução do casamento, com o resultado textual, conforme indicado na Figura 2, e com resultado de imagem, na Carta de Smith observado na Figura 4.

Em segundo lugar aparece a outra solução, com suas respectivas informações textuais e Carta de Smith, assim observados nas Figuras 3 e 5.

Primeira Solução

Comprimento de Onda: 0.1499 m
 Impedância Normalizada: $1.2000 + j\,0.6000$
 Admitância Normalizada: $0.6667 + j\,-0.3333$
 Distância do Toco à carga: 0.0646 m
 Comprimento do Toco em aberto: 0.0625 m
 Comprimento do Toco em curto: 0.0250 m
 V_{SWR} em aberto para frequência máxima: 1.1020
 V_{SWR} em aberto para frequência mínima: 1.1119
 V_{SWR} em curto para frequência máxima: 1.0799
 V_{SWR} em curto para frequência mínima: 1.0847

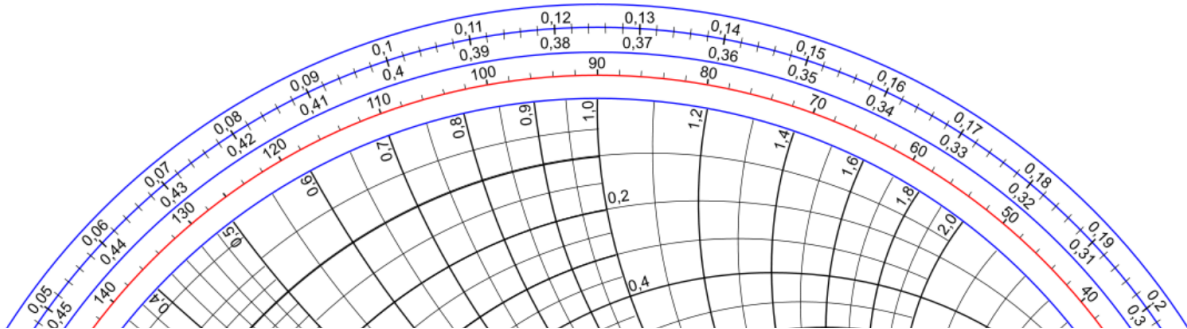


Figura 2 – Indicação textual da primeira solução do casamento.

Segunda Solução

Comprimento de Onda: 0.1499 m
 Impedância Normalizada: $1.2000 + j\,0.6000$
 Admitância Normalizada: $0.6667 + j\,-0.3333$
 Distância do Toco à carga: 0.0338 m
 Comprimento do Toco em aberto: 0.0125 m
 Comprimento do Toco em curto: 0.0500 m
 V_{SWR} em aberto para frequência máxima: 1.0428
 V_{SWR} em aberto para frequência mínima: 1.0416
 V_{SWR} em curto para frequência máxima: 1.0758
 V_{SWR} em curto para frequência mínima: 1.0715

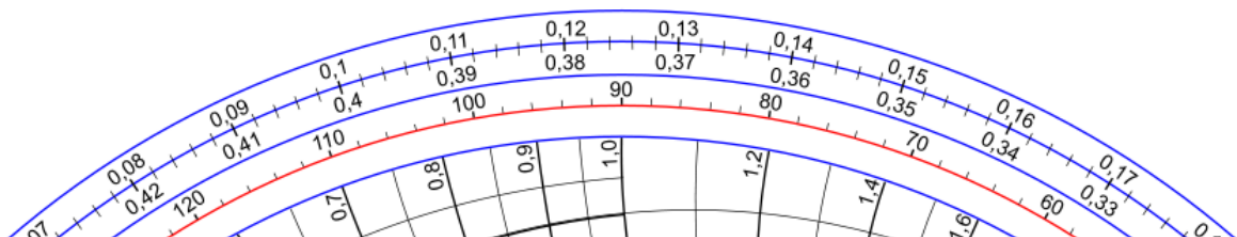


Figura 3 – Indicação textual da segunda solução do casamento.

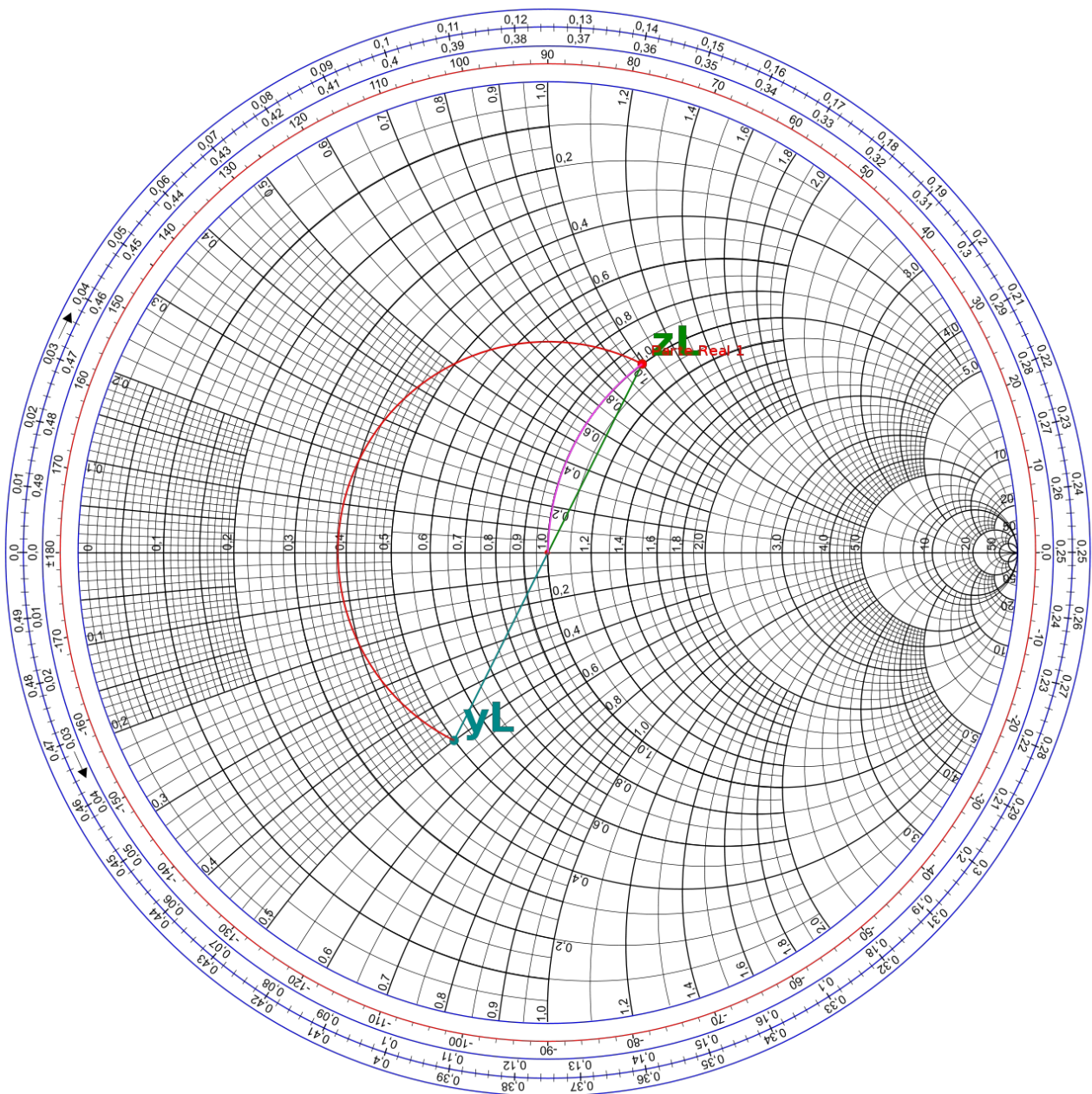


Figura 4 – Carta de Smith da primeira solução do casamento.

Na Carta de Smith, é importante notar os pontos z_L (impedância da carga normalizada) e y_L (admitância da carga normalizada), apresentados em verde e verde marinho, respectivamente. O círculo de p constante (módulo do coeficiente de reflexão) também é tracado, em segmento, em vermelho. Esse segmento representa o comprimento elétrico percorrido até que seja atingido o círculo de parte real unitária, cujo ponto também é representado em vermelho.

Dessa foma, com a adição do toco simples, percorre-se a curva em magenta, atingindo o centro da Carta de Smith, o que caracteriza um casamento perfeito. A primeira solução para essa entrada deu-se no segundo cruzamento do círculo de parte real unitária, já a segunda solução, no primeiro cruzamento.

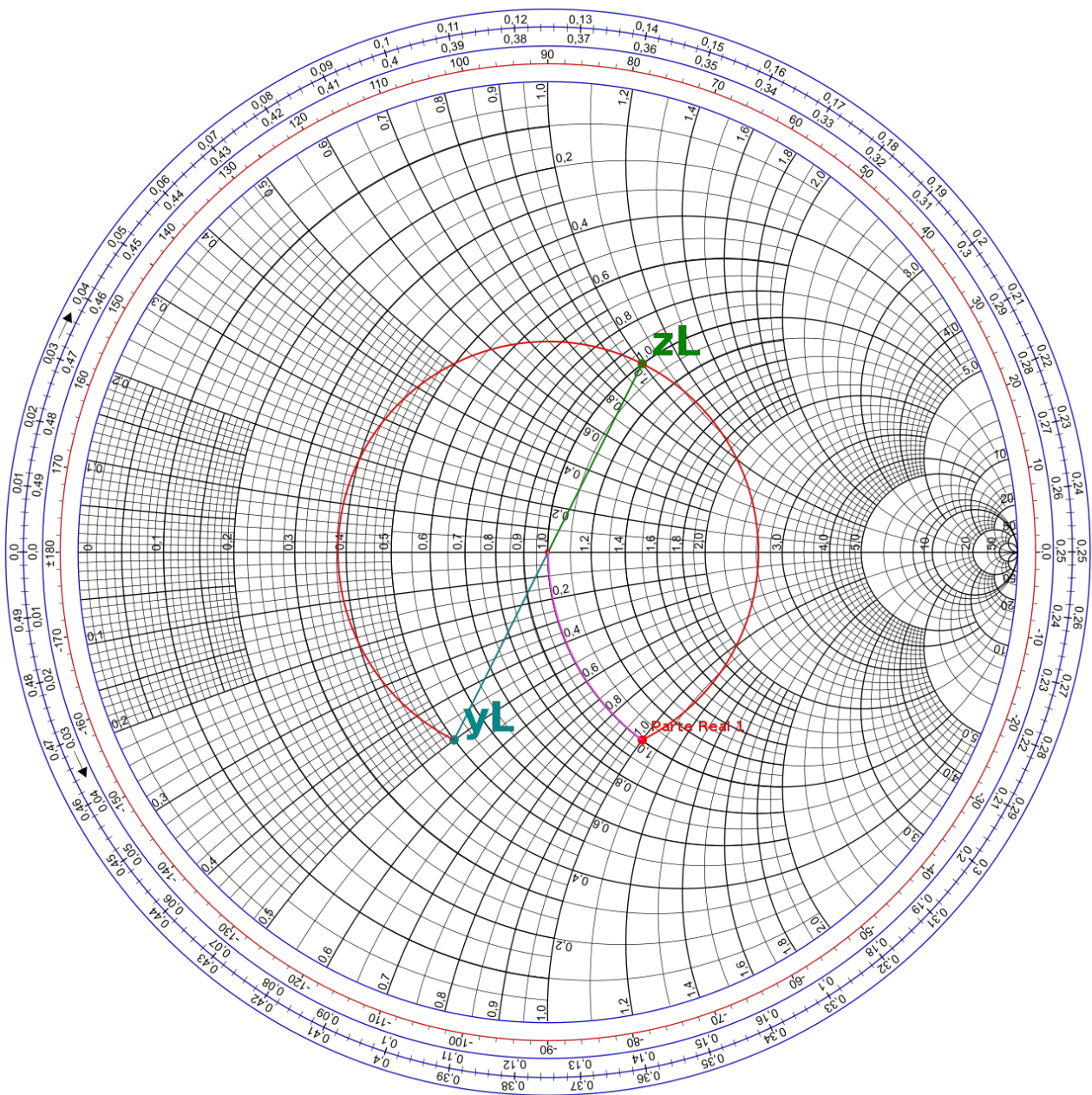


Figura 5 - Carta de Smith da segunda solução do casamento.

O usuário também pode clicar com o botão direito do mouse nas figuras e selecionar a opção “Salvar imagem como” para poder baixá-las na sua resolução máxima de 1300 por 1300 pixels de largura e altura.

Finalmente, tem-se o gráfico com a análise da largura de faixa do casamento, que traça as curvas de SWR em função da frequência de operação. São traçadas 4 curvas referentes aos casamentos com toco em aberto e em curto para a primeira solução e também para o casamento da segunda solução com toco em aberto e em curto. O gráfico gerado para as entradas expostas na Figura 1 pode ser observado na Figura 6. É importante notar que a escala dos valores aparecem representados por B, que indica bilhão. Assim a frequência de 2B Hz é de 2 MHz.

Análise de Largura de faixa

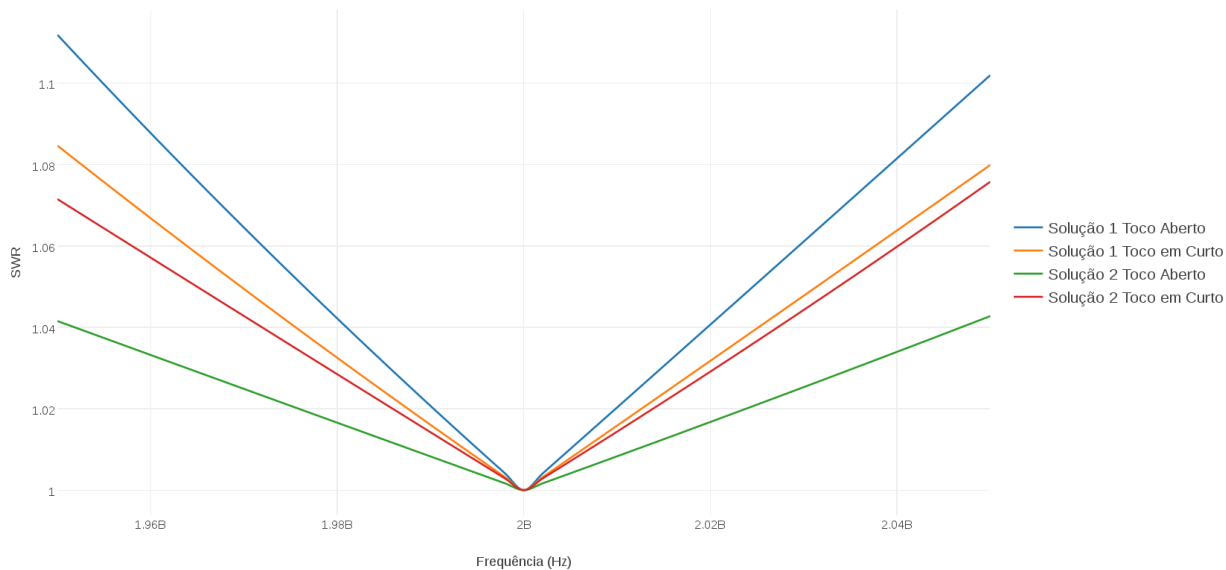


Figura 6 – Gráfico gerado para a análise de largura de faixa.

A Carta de Smith pode ser observada junto com o texto que descreve a sua solução, como indicado na Figura 7, uma vez que o zoom da página pode ser ajustado.

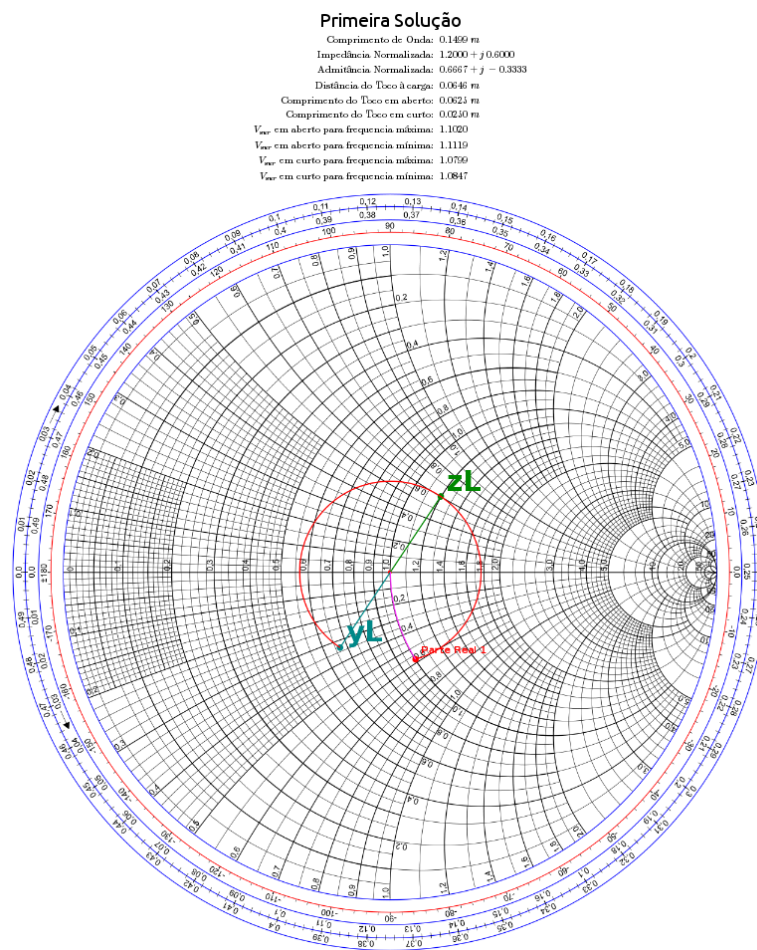


Figura 7 – Conjunto do texto com a Carta de Smith.

Conclusão

O presente trabalho, acompanhado de seus arquivos web, realiza o cálculo do comprimento do toco (em curto ou em aberto) e da distância do toco à carga, indicando as duas soluções. O algoritmo foi testado para diversos casos, e comparados com o material de referência. O resultado final foi um código robusto, de precisão maior que 8 algarismos significativos e que observa os casos nos quais poderiam ocorrer divisão por zero.

Em um primeiro momento, foi observado uma instabilidade computacional quando o denominador de uma divisão se aproximava de zero, o que gerava erros nos cálculos ou resultados que não podiam ser representados numericamente. Depois de extensiva análise e testes, o erro foi sanado e o aplicativo se tornou estável para todas as entradas testadas.

Além de sua robustez, esse aplicativo web possui, portanto, grande potencial didático, uma vez que diferentemente de uma planilha, a sua interface gráfica é amigável, é possível visualizar o desenho da Carta de Smith em alta resolução com os traçados e contornos dos pontos e também há o gráfico com a análise de banda por meio do SWR.

O aplicativo cumpriu com sucesso os seus requisitos de projetos, mas também, como forma de continuidade, ele pode ser aprimorado para poder realizar outros tipos casamentos e ainda permitir que o usuário faça, manualmente, um casamento, com a possibilidade de inserir um número arbitrário de tocos em posições arbitrárias.

Anexo

Código 1:

```

/home/gabruí/arquivos/ITA/5S/ELE-12/Lab1/lab1.html
1 <!DOCTYPE html>
2
3 <html>
4
5 <head>
6     <title>Lab1</title>
7     <meta charset="UTF-8">
8     <meta name="viewport" content="width=device-width, initial-scale=1.0">
9     <script src="math.js" type="text/javascript"></script>
10    <script src="src/plotly-latest.min.js"></script>
11    <script src="codigo.js" type="text/javascript"></script>
12    <script src="cartaSmith.js" type="text/javascript"></script>
13    <script type="text/x-mathjax-config">
14        MathJax.Hub.Config({
15            tex2jax: {inlineMath: [['$', '$'], ['\\(', '\\)']]}
16        });
17    </script>
18    <script type="text/javascript" async src="path-to-mathjax/MathJax.js?
config=TeX-AMS_CHTML"></script>
19    <script type="text/javascript" async
20        src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?
config=TeX-MML-AM_CHTML">
21    </script>
22    <link rel="stylesheet" href="estilo.css">
23    <link href="https://fonts.googleapis.com/css?family=Ubuntu"
rel="stylesheet">
24 </head>
25
26
27 <body>
28 <h1> Lab 1 - ELE-12 - Casamento de impedância com toco simples em
paralelo</h1>
29 <p> Especifique as entradas para o cálculo do casamento com toco simples em
paralelo:</p>
30 <form id="entradas" action="">
31     <table>
32         <tr>
33             <td>$Z_0$</td>
34             <td><input type="number" name="ReZ0" step="any" value="50"> <!--+ j
<input type="number" name="ImZ0" step="any" value="0">--></td>
35         </tr>
36         <tr>
37             <td>$Z_L$</td>
38             <td><input type="number" name="ReZL" step="any" value="80"> + $j$
<input type="number" name="ImZL" step="any" value="30"></td>
39         </tr>
40         <tr>
41             <td>$\text{Frequência de Casamento}\; \; \text{(MHz)}$</td>
42             <td><input type="number" name="freq" step="any" value="2000"></td>
43         </tr>
44         <tr>
45             <td>$U_{rel}$</td>
46             <td><input type="number" name="urel" step="any" value="1"></td>
47         </tr>
48         <tr>
```

```

49         <td>\epsilon_{rel}</td>
50         <td><input type="number" name="erel" step="any" value="1"></td>
51     </tr>
52     <tr>
53         <td>\text{BW de operação}\;; (MHz)</td>
54         <td><input type="number" name="bw" step="any" value="300"></td>
55     </tr>
56 </table>
57 </form>
58
59 <p>Clique em calcular para obter as respostas.</p>
60
61 <button onclick="calcular()">Calcular</button>
62 <p></p>
63 
64 <p></p><p></p>
65 <div id="respostas"></div>
66
67 <canvas id="desenho" width="1300" height="1300">
68     
69 </canvas>
70
71 <div id="respostas2"></div>
72
73 <canvas id="desenho2" width="1300" height="1300">
74 </canvas>
75
76 <div id="swr" style="width: 1300px; height: 700px; margin: auto"></div>
77
78 </body>
79 </html>
80

```

Código 2:

```

/home/gabru/arquivos/ITA/5S/ELE-12/Lab1/estilo.css
1 * {
2     text-align: center;
3 }
4
5 td:nth-child(1) {
6     text-align: right;
7 }
8
9 td:nth-child(2) {
10     text-align: left;
11 }
12
13 table {
14     margin: auto;
15 }
16 h1, p {
17 font-family: 'Ubuntu', sans-serif;
18 }
19
20 caption {
21     font-size: 2em;
22     font-family: 'Ubuntu', sans-serif;
23 }

```

Código 3:

```
C:\Users\Dylan N. Sugimoto\Desktop\ITA\COMP\ELE-12\LAB_1_ELE_12\ELE12Lab1\codigo.js
1 //UTILIZA A BIBLIOTECA math.js para o cálculo com números complexos
2
3 /* global math, MathJax, Plotly */
4
5 var epsilon = math.pow(10, -8);
6
7 /**
8  * @function somarSeNegativo Soma um incremento a um número se este for negativo
9  * @param {Number} incremento Soma esse incremento ao número
10 * @param {Number} numero Número a ser analisado se negativo
11 * @returns {Number} Resultado da operação de soma, se numero for negativo
12 */
13 function somarSeNegativo(incremento, numero) {
14     if (numero < 0) {
15         return numero + incremento;
16     }
17     return numero;
18 }
19
20
21 /**
22 * @function calcularNovoyToco calcula a admitancia do toco para um valor de frequencia
23 * @param {Number} aberto variavel booleana que diz se o toco esta em aberto ou nao
24 * @param {Number} l comprimento do toco
25 * @param {Number} f valor de frequencia
26 * @param {Number} u valor da permeabilidade magnetica
27 * @param {Number} e valor de permissividade eletrica
28 * @returns {Number} Retorna o valor da admitancia do toco
29 */
30 function calcularNovoyToco(aberto, l, f, u, e) {
31     if (aberto){
32         return math.tan(2*math.pi*l*f*math.sqrt(u*e));
33     }
34     else{
35         return -1*math.cot(2*math.pi*l*f*math.sqrt(u*e));
36     }
37 }
38
39
40 /**
41 * @function calcularModR Calcula o modulo do coeficiente de reflexao
42 * @param {Number} a parte real da admitancia de carga
43 * @param {Number} b parte imaginaria da admitancia de carga
44 * @param {Number} NyT valor da admitancia do toco
45 * @param {Number} x posicao do toco
46 * @param {Number} f frequencia
47 * @param {Number} u permeabilidade magnetica
48 * @param {Number} e permissividade eletrica
49 * @returns {Number}
50 */
51 function calcularModR(a,b,NyT,x,f,u,e){
52     var t = math.tan(2*math.pi*x*f*math.sqrt(u*e));
53     var t2 = t*t;
54     var t3 = t2*t;
55     var t4 = t2*t2;
56     var Rey = (a - a*b*t + t*a*(b+t))/((1-b*t)*(1-b*t)+(t*a)*(t*a));
57     var Imy = (-t*a*a+(b+t)*(1-b*t))/((1-b*t)*(1-b*t)+(t*a)*(t*a));
58     var ReyLf = (Rey);
59     var ImyLf = (-NyT+Imy);
```

```

60 var num2 = (1-ReyLf)*(1-ReyLf) + (ImyLf)*(ImyLf);
61 var dem2 = (1+ReyLf)*(1+ReyLf) + (ImyLf)*(ImyLf);
62 return math.sqrt(num2/dem2);
63 }
64
65
66 /**
67 * @function calcularVswr Retorna o VSWR a partir do coeficiente de reflexão
68 * @param {Number} R Módulo do coeficiente de Reflexão
69 * @returns {Number} VSWR calculado
70 */
71 function calcularVswr(R) {
72     return (1+R)/(1-R);
73 }
74
75
76 /**
77 *
78 * @param {math.complex} zComplexo A impedância da carga normalizada.
79 * @returns {math.complex} O coeficiente de reflexão complexo
80 */
81 function calcularReflexao(zComplexo) {
82     return math.divide(math.subtract(zComplexo, 1), math.add(1, zComplexo));
83 }
84
85
86
87 /**
88 * Gira no sentido horário para o cálculo da diferença de ângulo
89 * @param {number} inicio Ângulo inicial entre -pi e pi
90 * @param {number} fim Ângulo final entre -pi e pi
91 * @returns {number} Distância angular a ser percorrida no sentido horário
92 */
93 function calcDifAngRad(inicio, fim) {
94     // Tem que passar pela descontinuidade
95     if (inicio < fim) {
96         return 2*math.pi - (fim - inicio);
97     } else {
98         return inicio - fim;
99     }
100 }
101
102 function reveal(){
103     $("#desenho").show();
104     $("#desenho2").hide();
105 }
106 }
107
108 function reveal2(){
109     $("#desenho2").show();
110     $("#desenho").hide();
111 }
112
113 function calcular() {
114
115     var ReZ0 = document.getElementById("ReZ0")[0].value;
116     var ImZ0 = 0;//document.getElementById("ImZ0")[0].value;
117     var Z0 = math.complex(ReZ0, ImZ0);
118     var ReZL = document.getElementById("ReZL")[0].value;
119     var ImZL = document.getElementById("ImZL")[0].value;
120     var ZL = math.complex(ReZL, ImZL);
121     var freq = document.getElementById("freq")[0].value * math.pow(10, 6);
122     var urel = document.getElementById("urel")[0].value;

```



```

123 var erel = document.getElementsByName("erel")[0].value;
124 var bw = document.getElementsByName("bw") [0].value * math.pow(10,6);
125 var u = urel * 4 * math.pi * math.pow(10, -7);
126 var e = erel * 8.8541878176 * math.pow(10, -12);
127 // Impedância Normalizada da carga
128 var ZLNorm = math.divide(ZL, Z0);
129 // Admitância Normalizada da carga
130 var YLNorm = math.divide(1, ZLNorm);
131 // Comprimento de onda
132 var lambda = 1/(math.sqrt(u*e)*freq);
133 // Coeficiente de reflexão do toco em curto normalizada
134 var rTocoCurto = math.complex(-1, 0);
135 // Admitância do toco em aberto normalizada
136 var rTocoAberto = math.complex(1, 0);
137 // Frequência máxima da banda de operação
138 var fmax = freq + bw/2;
139 // Frequência mínima da banda de operação
140 var fmin = freq - bw/2;
141 var freqs = [fmin, fmax];
142
143
144 /* CÓDIGO PARA CALCULAR PELA CARTA DE SMITH
145 for (var numSol = 1; numSol <=2; numSol++) {
146     // Coeficiente de Reflexão da carga, complexo
147     var reflexaoCarga = calcularReflexao(ZLNorm);
148     // O seu módulo
149     var reflexaoCargaModulo = reflexaoCarga.toPolar().r;
150     // O coeficiente de reflexão no ponto de parte real da admitância igual
151     // a 1. Ele é simétrico com relação a outra solução
152     var reflexaoParteReal1 = math.complex.fromPolar(reflexaoCargaModulo,
153         math.acos(reflexaoCargaModulo) * ((numSol==1)?1:-1));
154     // A admitância desse ponto de reflexão
155     var admitanciaParteReal1 = math.divide(math.add(1, reflexaoParteReal1),
156         math.subtract(1, reflexaoParteReal1));
157     // Um coeficiente de reflexão associada à admitância da carta de smith
158     var reflexaoAdmitancia = calcularReflexao(YLNorm);
159     // A distância elétrica
160     var xEletrico = calcDifAngRad(reflexaoAdmitancia.toPolar().phi,
161         reflexaoParteReal1.toPolar().phi)*0.5/(2*math.pi);
162     // A distância real
163     var x = xEletrico * lambda;
164     // Comprimento do Toco em curto
165     var compTocoCurto = calcDifAngRad(rTocoCurto.toPolar().phi,
166         -reflexaoParteReal1.toPolar().phi)*lambda*0.5/(2*math.pi);
167     // Comprimento do Toco em aberto
168     var compTocoAberto = calcDifAngRad(rTocoAberto.toPolar().phi,
169         -reflexaoParteReal1.toPolar().phi)*lambda*0.5/(2*math.pi);
170
171     // Descasamento de
172     for (var tam = freqs.length - 1; tam >= 0; tam--) {
173     }
174 }
175 */
176
177 var a = YLNorm.re;
178 var b = YLNorm.im;
179
180
181 var A = a*a + b*b - a;
182 var B = -2*b;
183 var C = 1 - a;
184 var delta = B*B - 4*A*C;
185 if (math.abs(A) < epsilon) {

```

```

186     var t2 = -C/B;
187
188     var x1 = lambda/4;
189     var x2 = somarSeNegativo(lambda/2, lambda * math.atan(t2)/(2*math.pi));
190
191     var lmy1 = ZLNorm.im;
192     var lmy2 = -ZLNorm.im;
193 } else {
194     var t1 = (-B + math.sqrt(delta))/(2*A);
195     var t2 = (-B - math.sqrt(delta))/(2*A);
196
197     var x1 = somarSeNegativo(lambda/2, lambda * math.atan(t1)/(2*math.pi));
198     var x2 = somarSeNegativo(lambda/2, lambda * math.atan(t2)/(2*math.pi));
199
200     var lmy1 = ((b+t1)*(1-b*t1)-t1*a*a)/((1-b*t1)*(1-b*t1)+(t1*a)*(t1*a));
201     var lmy2 = ((b+t2)*(1-b*t2)-t2*a*a)/((1-b*t2)*(1-b*t2)+(t2*a)*(t2*a));
202 }
203
204
205 var comp1aberto = somarSeNegativo(lambda/2, math.atan(-lmy1)*lambda/(2*math.pi));
206 var comp1curto = somarSeNegativo(lambda/2, math.atan(1/lmy1)*lambda/(2*math.pi));
207 var comp2aberto = somarSeNegativo(lambda/2, math.atan(-lmy2)*lambda/(2*math.pi));
208 var comp2curto = somarSeNegativo(lambda/2, math.atan(1/lmy2)*lambda/(2*math.pi));
209
210
211 var f = fmin;
212 var quantIter = 50;
213 var df = bw/quantIter;
214 var xGrafico = [];
215 var Vswr1aberto = [];
216 var Vswr2aberto = [];
217 var Vswr1curto = [];
218 var Vswr2curto = [];
219 for (var quant = 0; quant <= quantIter; quant++, f+=df) {
220     var NyTaberto1 = calcularNovoyToco(true,comp1aberto,f,u,e);
221     var NyTaberto2 = calcularNovoyToco(true,comp2aberto,f,u,e);
222     var NyTcurto1 = calcularNovoyToco(false,comp1curto,f,u,e);
223     var NyTcurto2 = calcularNovoyToco(false,comp2curto,f,u,e);
224
225     var R1aberto = calcularModR(a,b,NyTaberto1,x1,f,u,e);
226     var R2aberto = calcularModR(a,b,NyTaberto2,x2,f,u,e);
227
228     var R1curto = calcularModR(a,b,NyTcurto1,x1,f,u,e);
229     var R2curto = calcularModR(a,b,NyTcurto2,x2,f,u,e);
230
231     xGrafico[quant] = f;
232     Vswr1aberto[quant] = calcularVswr(R1aberto);
233     Vswr2aberto[quant] = calcularVswr(R2aberto);
234     Vswr1curto[quant] = calcularVswr(R1curto);
235     Vswr2curto[quant] = calcularVswr(R2curto);
236 }
237
238 var texto = "";
239 var carta1 = new CartaSmith(649, 649, 1196-648, "desenho", "fundo");
240
241 //texto += "<table>";
242 //texto += "<caption>Primeira Solução</caption>";
243 var jtextz = "+j\\;";
244 if(ZLNorm.im < 0) {
245     jtextz = "-j\\;";
246 }
247 var jtexty = "+j\\;";
248 if(YLNorm.im < 0) {

```

```

249     jtexty = "-j\\;";
250 }/**/
251 texto += "<tr><td> </td><th> Primeira Solução</th><th width = '300'> Segunda Solução</th></tr>";
252 texto += "<tr><td>\\text{Comprimento de Onda(m): }$</td><td>$" + lambda.toFixed(4) + "\\; $</td><td>$" +
lambda.toFixed(4) + "$</td></tr>";
253 texto += "<tr><td>\\text{Impedância Normalizada($\\Omega$): }$</td><td>$" + ZLNorm.re.toFixed(4)
+jtextz + math.abs(ZLNorm.im).toFixed(4) + "$</td><td>$" + ZLNorm.re.toFixed(4) + jtextz +
math.abs(ZLNorm.im).toFixed(4) + "$</td></tr>";
254 carta1.desenharRetaZNorm(ZLNorm);
255 carta1.desenharPontoZNorm(ZLNorm, "zL");
256 texto += "<tr><td>\\text{Admitância Normalizada($\\Omega$): }$</td><td>$" + YLNorm.re.toFixed(4)
+jtexty + math.abs(YLNorm.im).toFixed(4) + "$</td><td>$" + YLNorm.re.toFixed(4) + jtexty +
math.abs(YLNorm.im).toFixed(4) + "$</td></tr>";
257 carta1.setCor("#008888");
258 carta1.desenharRetaZNorm(YLNorm);
259 carta1.desenharPontoZNorm(YLNorm, "yL");
260 carta1.setCor("#FF0000");
261 carta1.curvaRConst(calcularReflexao(YLNorm), calcularReflexao(math.complex(1, Imy1)));
262 carta1.setFonte("1em");
263 carta1.desenharPontoZNorm(math.complex(1, Imy1), "Parte Real 1");
264 carta1.setCor("#FF33FF");
265 carta1.interpolarZ(math.complex(1, Imy1), math.complex(1, 0));
266 texto += "<tr><td>\\text{Distância do Toco à carga(m): }$</td><td>$" + x1.toFixed(4) + "$</td><td>$" +
x2.toFixed(4) + "$</td></tr>";
267 texto += "<tr><td>\\text{Comprimento do Toco em aberto(m): }$</td><td>$" + comp1aberto.toFixed(4) + "$</td><td>$" +
comp2aberto.toFixed(4) + "$</td></tr>";
268 texto += "<tr><td>\\text{Comprimento do Toco em curto(m): }$</td><td>$" + comp1curto.toFixed(4) + "$</td><td>$" +
comp2curto.toFixed(4) + "$</td></tr>";
269 texto += "<tr><td>$V_{swr}$ \\text{ em aberto para frequencia máxima: }$</td><td>$" + Vswr1aberto[quantIter].toFixed(4) + "\\; $</td><td>$" + Vswr2aberto[quantIter].toFixed(4) +
"$</td></tr>";
270 texto += "<tr><td>$V_{swr}$ \\text{ em aberto para frequencia mínima: }$</td><td>$" + Vswr1aberto[0].toFixed(4) + "\\; $</td><td>$" + Vswr2aberto[0].toFixed(4) + "$</td></tr>";
271 texto += "<tr><td>$V_{swr}$ \\text{ em curto para frequencia máxima: }$</td><td>$" + Vswr1curto[quantIter].toFixed(4) + "\\; $</td><td>$" + Vswr2curto[quantIter].toFixed(4) +
"$</td></tr>";
272 texto += "<tr><td>$V_{swr}$ \\text{ em curto para frequencia mínima: }$</td><td>$" + Vswr1curto[0].toFixed(4) + "\\; $</td><td>$" + Vswr2curto[0].toFixed(4) + "$</td></tr>";
273 texto += "<tr><td>Carta de Smith</td><td><button onclick='reveal()'>Mostrar</button></td><td><button
onclick='reveal2()'>Mostrar</button></td></tr>";
274
275
276 var rtable = $("#respostas").find("table");
277 rtable.html(texto);
278
279
280
281
282 var texto2 = "";
283 var carta2 = new CartaSmith(649, 649, 1196-648, "desenho2", "fundo");
284 carta2.desenharRetaZNorm(ZLNorm);
285 carta2.desenharPontoZNorm(ZLNorm, "zL");
286 texto2 += "<tr><td>\\text{Admitância Normalizada: }$</td><td>$" + YLNorm.re.toFixed(4) + " + j\\;" +
YLNorm.im.toFixed(4) + "$</td></tr>";
287 carta2.setCor("#008888");
288 carta2.desenharRetaZNorm(YLNorm);
289 carta2.desenharPontoZNorm(YLNorm, "yL");
290 carta2.setCor("#FF0000");
291 carta2.curvaRConst(calcularReflexao(YLNorm), calcularReflexao(math.complex(1, Imy2)));
292 carta2.setFonte("1em");
293 carta2.desenharPontoZNorm(math.complex(1, Imy2), "Parte Real 1");
294 carta2.setCor("#FF33FF");
295 carta2.interpolarZ(math.complex(1, Imy2), math.complex(1, 0));

```

```

296
297 MathJax.Hub.Queue(["Typeset",MathJax.Hub,"respostas"]);
298
299
300 var dadosVswr1a = {
301     x: xGrafico,
302     y: Vswr1aberto,
303     mode: 'lines',
304     name: 'Solução 1 Toco Aberto',
305     line: {shape: 'spline'},
306     type: 'scatter'
307 };
308 var dadosVswr2a = {
309     x: xGrafico,
310     y: Vswr2aberto,
311     mode: 'lines',
312     name: 'Solução 2 Toco Aberto',
313     line: {shape: 'spline'},
314     type: 'scatter'
315 };
316 var dadosVswr1c = {
317     x: xGrafico,
318     y: Vswr1curto,
319     mode: 'lines',
320     name: 'Solução 1 Toco em Curto',
321     line: {shape: 'spline'},
322     type: 'scatter'
323 };
324 var dadosVswr2c = {
325     x: xGrafico,
326     y: Vswr2curto,
327     mode: 'lines',
328     name: 'Solução 2 Toco em Curto',
329     line: {shape: 'spline'},
330     type: 'scatter'
331 };
332
333 var dados = [dadosVswr1a, dadosVswr1c, dadosVswr2a, dadosVswr2c];
334
335 var estilo = {
336     legend: {
337         y: 0.5,
338         font: {size: 16},
339         yref: 'paper'
340     },
341     title: 'Análise de Largura de faixa',
342     xaxis: {
343         title: 'Frequência (Hz)'
344     },
345     yaxis: {
346         title: 'SWR'
347     }
348 };
349
350 Plotly.newPlot('swr', dados, estilo);
351
352
353 }
354
355

```

Código 4:

```

    /home/gabruí/arquivos/ITA/5S/ELE-12/Lab1/cartaSmith.js
1  /* global math */
2
3  //UTILIZA FUNÇÕES DO ARQUIVO codigo.js
4
5  /**
6   * @class Representação gráfica da Carta de Smith
7   * @param {Number} cx A coordenada x do centro da carta de smith da imagem
8   * @param {Number} cy A coordenada y do centro da carta de smith da imagem
9   * @param {Number} r O raio da carta de smith da imagem
10  * @param {string} desenho O ID do canvas no html
11  * @param {string} fundo O ID da imagem de fundo do html
12  * @returns {CartaSmith} Um objeto do tipo carta de smith
13  */
14  CartaSmith = function (cx, cy, r, desenho, fundo) {
15
16      /**
17       *
18       * @param {Number} cx A coordenada x do centro da carta de smith da
imagem
19       * @param {Number} cy A coordenada y do centro da carta de smith da
imagem
20       * @param {Number} r O raio da carta de smith da imagem
21       * @param {string} desenho O ID do canvas no html
22       * @param {string} fundo O ID da imagem de fundo do html
23       */
24      this.iniciar = function(cx, cy, r, desenho, fundo) {
25          this.cx = cx; ///< Posição x do centro da figura
26          this.cy = cy; ///< Posição y do centro da figura
27          this.r = r;    ///< Raio da carta de smith da figura
28          this.canvas = document.getElementById(desenho);
29          this.desenho = this.canvas.getContext("2d");
30          this.fundo = document.getElementById(fundo);
31          this.desenho.restore();
32          this.apagar();
33          this.desenho.moveTo(0, 0);
34          this.setCor("#008800");
35          this.desenho.lineWidth = 2;
36          this.fonte = "'Open Sans', sans-serif";
37          this.desenho.font = "bold 3em 'Open Sans', sans-serif";
38      };
39
40      /**
41       * Muda a cor de desenho dos pontos e retas da carta
42       * @param {string} cor A representação hexadecimal ou de nome da cor
43       */
44      this.setCor = function(cor) {
45          this.desenho.fillStyle = cor;
46          this.desenho.strokeStyle = cor;
47      };
48
49      /**
50       *
51       * Muda a fonte
52       * @param {type} fonte A representação da fonte
53       */
54      this.setFonte = function(fonte) {
55          this.desenho.font = "bold " + fonte + this.fonte;
56      };
57
58  }
```

```

59  /**
60   * @function apagar Apaga que foi desenhado
61   */
62  this.apagar = function() {
63      this.desenho.clearRect(0, 0, this.canvas.width, this.canvas.height);
64      this.desenho.drawImage(this.fundo, 0, 0);
65  };
66
67
68  /**
69   * @function desenharPontoZNorm Desenha um ponto de impedância
normalizada
70   * na carta de smith.
71   * @param {math.complex} zNormComplexo Impedância normalizada
72   * @param {string} texto Comentário sobre o ponto
73   */
74  this.desenharPontoZNorm = function(zNormComplexo, texto) {
75      this.desenharR(calcularReflexao(zNormComplexo), texto);
76  };
77
78
79  /**
80   * @function desenharRetaR Desenha uma reta que liga o centro da carta
de
81   * smith até o ponto que representa uma impedância normalizada.
82   * @param {math.complex} zNormComplexo Impedância normalizada
83   */
84  this.desenharRetaZNorm = function(zNormComplexo) {
85      this.desenharRetaR(calcularReflexao(zNormComplexo));
86  };
87
88
89  /**
90   * @function desenharR Desenha um ponto na carta de Smith referente a o
91   * coeficiente de reflexão dado
92   * @param {math.complex} reflexaoComplexo Coeficiente de reflexão
93   * @param {string} texto Texto a ser legenda do ponto
94   */
95  this.desenharR = function(reflexaoComplexo, texto) {
96      var x = reflexaoComplexo.re*this.r + this.cx; // Posição x na
figura
97      var y = -reflexaoComplexo.im*this.r + this.cy; // Posição y na
figura
98      this.desenho.beginPath();
99      this.desenho.arc(x, y, this.r/100, 0, 2*Math.PI);
100     this.desenho.fill();
101     if (texto) {
102         this.desenho.fillText(texto, x + 10, y - 10);
103     }
104 };
105
106
107  /**
108   * @function desenharRetaR Desenha uma reta que liga o centro da carta
de
109   * smith até o ponto de coeficiente de reflexão especificado.
110   * @param {math.complex} reflexaoComplexo Coeficiente de reflexão
111   */
112  this.desenharRetaR = function(reflexaoComplexo) {
113      var x = reflexaoComplexo.re*this.r + this.cx; // Posição x na
figura
114      var y = -reflexaoComplexo.im*this.r + this.cy; // Posição y na
figura
115      this.desenho.beginPath();

```



```

116         this.desenho.moveTo(this.cx, this.cy);
117         this.desenho.lineTo(x, y);
118         this.desenho.stroke();
119     };
120
121
122     /**
123     *
124     * @param {math.complex} zNormInicial Z Normalizado inicial
125     * @param {math.complex} zNormFinal Z Normalizado final
126     */
127     this.interpolarZ = function(zNormInicial, zNormFinal) {
128         var i = 0;
129         var iteracoes = 100;
130         var delta = math.divide(math.subtract(zNormFinal, zNormInicial),
131 iteracoes);
132         var temp = zNormInicial.clone();
133         var rTemp = calcularReflexao(temp);
134         var x = rTemp.re*this.r + this.cx; // Posição x na figura
135         var y = -rTemp.im*this.r + this.cy; // Posição y na figura
136         this.desenho.beginPath();
137         this.desenho.moveTo(x, y);
138         for(i = 0; i<iteracoes; i++) {
139             temp = math.add(temp, delta);
140             rTemp = calcularReflexao(temp);
141             x = rTemp.re*this.r + this.cx; // Posição x na figura
142             y = -rTemp.im*this.r + this.cy; // Posição y na figura
143             this.desenho.lineTo(x, y);
144         }
145         this.desenho.stroke();
146     };
147
148     /**
149     * Desenha uma curva, segmento de circunferência de Coeficiente de
150 Reflexão
151     * constante.
152     * @param {math.complex} rInicial Marca o ponto de início, é o seu raio
153     * que
154     * é utilizado
155     * @param {math.complex} rFinal Marca o ponto de final
156     */
157     this.curvaRConst = function(rInicial, rFinal) {
158         this.desenho.beginPath();
159         this.desenho.arc(this.cx, this.cy, rInicial.toPolar().r * this.r,
160 -rInicial.toPolar().phi, -rFinal.toPolar().phi);
161         this.desenho.stroke();
162     };
163
164     this.iniciar(cx, cy, r, desenho, fundo);
165 }

```