Transientes Em Linhas De Transmissão Sem Perdas

J. E. Oliveira, V. R. Leonardi and L. R. d. Souza

***Abstract*—Esse relatório tem como objetivo explicar e detalhar o algoritmo de diferenças finitas no domínio do tempo(FDTD) e seu uso na análise computacional do comportamento de transientes de tensão e corrente em linhas de transmissão sem perdas, particularmente com o uso do programa Octave.**

***Keywords*— Linhas de transmissão sem perdas; FDTD; transientes; Octave; visualização de tensão e corrente; simulação.**

I. INTRODUÇÃO

LINHAS de transmissão são circuitos utilizados para a transmissão de sinais eletromagnéticos. Elas se tornam necessárias quando o comprimento das ondas transmitidas é curto o suficiente para que o comprimento da linha seja uma parcela relevante desse valor [1].

Transientes são eventos de curta duração causados por mudanças abruptas no estado do circuito. No caso das linhas de transmissão, transientes podem ocorrer quando uma nova carga é conectada ao circuito, quando um sinal aperiódico é injetado na ou se propaga através da linha, ou quando o gerador é ligado [2].

A análise feita nesse projeto se baseia no transiente gerado por sinais aperiódicos do tipo degrau. Para que seja computacionalmente possível tal análise, é necessário realizar uma discretização do sistema. O método FDTD permite transformar as equações do telegrafista (1) e (2) em diferenças finitas, criando uma malha de pontos discretos no espaço e no tempo que serão utilizados para os calculos de *v*(z,t) e *i*(z,t).

$$\frac{-\partial v(z,t)}{\partial z} = Ri(z,t) + L\frac{\partial i(z,t)}{\partial t} \tag{1}$$

$$\frac{-\partial i(z,t)}{\partial z} = Gv(z,t) + C\frac{\partial v(z,t)}{\partial t} \tag{2}$$

O programa criado permite o cálculo e a representação gráfica da tensão e corrente ao longo do comprimento da linha, para um dado tempo. Também é possível adquirir os valores para uma dada posição da linha ao longo do tempo. Finalmente, é possível uma visualização em forma de animação, destacando a mudança dos valores ao longo do tempo e do comprimento da linha.

Foi escolhido o programa Octave, que é open source, para realizar a simulação, devido à facilidade da representação gráfica da tensão e corrente, além da útil ferramenta de criação de vídeos para ilustrar o comportamento em relação ao tempo.

II. DADOS DA LINHA DE TRANSMISSÃO

Para a correta realização da simulação é preciso especificar as características da linha utilizada. Primeiramente, o sistema simulado é baseado no modelo de linha de transmissão na Fig. 1.

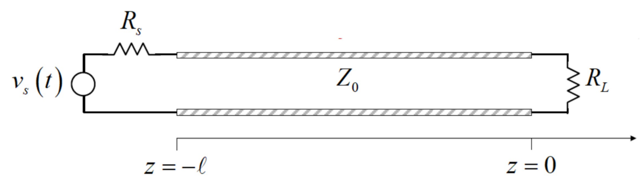

Figura 1. Modelo genérico de linha de transmissão[2].

Das especificações do projeto tem-se os seguintes valores:

$Z_0 = 50\Omega \quad R_S = 75\Omega$

A velocidade de propagação do sinal na linha é dada por:

$u_f = 0.9\mathrm{c}$, onde c é a velocidade da luz no vácuo.

Como é o caso de uma linha sem perdas, os parâmetros R e G podem ser considerados R = G = 0. Os parâmetros L e C da linha foram derivados de $Z_0 = \frac{1}{C u_f}$ e $Z_0 = \sqrt{\frac{L}{C}}$. Dessa forma, C =74,07 pF/m e L = 52,37 nH/m. O comprimento total da linha foi considerado z = 10m. Em questão dos valores da fonte e da carga acoplados, foram feitos testes para 2 fontes de tensão: $v_{s1} = 2u(t), v_{s2} = u(t) - u\left(t - \frac{z}{10\mathrm{u}_f}\right)$, onde *u(t)* é a função degrau. Para cada uma das fontes foram testados 3 valores de carga: $R_{L1} = 0$, $R_{L2} =$ e $R_{L3} = 100 \ \Omega$.

III. ALGORITMO FDTD

Para a linha de transmissão sem perdas, as equações do telegrafista (1) e (2) podem ser simplificadas com R = G = 0, resultando nas equações (3) e (4).

[1]J. E. Oliveira, Universidade de São Paulo, São Carlos, São Paulo, Brasil, jeosvici@usp.br, NUSP 10716491

V. R. Leonardi, Universidade de São Paulo, São Carlos, São Paulo, Brasil, vitorleonardi@usp.br, NUSP 10716567

L. R. de Souza, Universidade de São Paulo, São Carlos, São Paulo, Brasil, leo_souza@usp.br, NUSP

Corresponding author: Julio Eiji Oliveira

$$\frac{-\partial v(z,t)}{\partial z}=L\frac{\partial i(z,t)}{\partial t} \quad (3)$$

$$\frac{-\partial i(z,t)}{\partial z}=C\frac{\partial v(z,t)}{\partial t} \quad (4)$$

A partir dessas equações pode-se aplicar o método FDTD para transformá-las em diferenças finitas e criar a malha dos pontos necessários.

Defini-se então um espaçamento fixo Δz no espaço e Δt no tempo, de tal modo que os pontos da malha podem ser descritos como:

$z_k=k\Delta z$ e $t_n=n\Delta t$

Onde $1\leq k\leq K, 1\leq n\leq N$

Em seguida, aproxima-se o valor da derivada de tensão no espaço por uma diferença finita central:

$$\frac{\partial v(z,t)}{\partial z}\approx\frac{v(z+\Delta z,t)-v(z-\Delta z,t)}{2\Delta z} \quad (5)$$

Utilizando uma notação mais conveniente para a tensão, tal que $v(z_k,t_n)=v_k^n$, temos:

$$\frac{\partial}{\partial z}i_k^n\approx\frac{i_{k+1}^n-i_{k-1}^n}{2\Delta z} \quad (6)$$

De maneira similar, aplicanndo uma diferença finita no tempo, para a tensão temos:

$$\frac{\partial}{\partial t}v_k^n\approx\frac{v_k^{n+1}-v_k^{n-1}}{2\Delta t} \quad (7)$$

È possível melhorar a precisão sem custo adicional de computação ao tomar-se meio passos entre cada ponto. Assim:

$$\frac{\partial}{\partial z}i_k^n\approx\frac{i_{k+1/2}^n-v_{i-1/2}^n}{\Delta z} \quad (8)$$

$$\frac{\partial}{\partial t}v_k^n\approx\frac{v_k^{n+1/2}-v_k^{n-1/2}}{\Delta t} \quad (9)$$

Utilizando as equações (8) e (9) nas equações (3) e (4) tem-se a aproximação numérica das equações do telegrafista que pode ser usada para o cálculo computacional:

$$-\frac{i_{k+1/2}^n-i_{k-1/2}^n}{\Delta z}=C\frac{v_k^{n+1/2}-v_k^{n-1/2}}{\Delta t} \quad (10)$$

$$-\frac{v_{k+1}^{n+1/2}-v_k^{n+1/2}}{\Delta t}=L\frac{i_{k+1/2}^{n+1}-i_{k+1/2}^n}{\Delta z} \quad (11)$$

O algortimo FDTD consiste em encontrar o valor futuro de corrente e tensão a partir de valores passados conhecidos, então, rearranjando (10) e (11) tem-se como encontrar os valores futuros :

$$v_k^{n+1/2}=-\frac{\Delta t}{\Delta z C}(i_{k+1/2}^n-i_{k-1/2}^n)+v_k^{n-1/2} \quad (12)$$

$$i_{k+1/2}^{n+1}=-\frac{\Delta z}{\Delta t L}(v_{k+1}^{n+1/2}-v_k^{n+1/2})+i_{k+1/2}^n \quad (13)$$

Então, para um tempo n+1/2, o algoritmo calcula os valores v para em todas as posições k e em seguida faz o mesmo para os valores de i na posição k+1/2 e tempo n+1. No final de uma iteração o n é aumentado e o ciclo continua até que n>N. Neste caso, N é o valor do tempo para ser atingido o regime estácionario: $N=\frac{10z}{u_f}$.

IV. ESTABILIDADE

Na simulação do algoritmo é preciso tomar cuidado com a estabilidade do sistema: a escolha de valores impróprios de Δx e Δt podem levar os resultados a divergirem, criando erros. Sendo assim, é necessário definir os valores que permitirão manter o sistema estável.

Analisando um caso simples na linha de transmissão sem perda, onde a onda gerada não é refletida, tem-se a seguinte solução para as equaçoes do telegrafista, dadas no formato FDTD:

$$v_k^n=V_0^+\cos(\omega n\Delta t-\beta k\Delta z) \quad (14)$$

$$i_k^n=\frac{V_0^+}{Z_0}\cos(\omega n\Delta t-\beta k\Delta z) \quad (15)$$

Nota-se que o termo de fase é zero. Tomando V_0^+ = 1V e colocando as equações (14) e (15) em (12) chega-se em uma soma de cossenos. Assumindo um caso extremo onde $v_k^{n+1}=1$ e $v_k^{n-1/2}=-1$ tem-se:

$$2=-\frac{\Delta t}{\Delta z C Z_0}[\cos(A)-\cos(B)] \quad (16)$$

Onde:

$A=\omega(n)\Delta t-\beta(k+1/2)\Delta z$ e

$B=\omega(n)\Delta t-\beta(k-1/2)\Delta z$

Então, o caso máximo será aquele em que cos(A) – cos(B) = 2. A estabilidade do sistema dependerá apenas de $-\frac{\Delta t}{\Delta z C Z_0}$. Como é um caso máximo, deve-se definir um valor que garanta:

$$2 \leq -\frac{\Delta t}{\Delta z\, C\, Z_0} 2 \rightarrow 1 \leq -\frac{\Delta t}{\Delta z\, C\, Z_0} \qquad (17)$$

Substituindo $Z_0 = \frac{1}{C u_f}$, onde u_f é a velocidade de propagação do sinal e resolvendo para o espaçamento de tempo:

$$\Delta t \leq \frac{\Delta z}{u_f} \qquad (18)$$

Este é o valor máximo que o espaçamento de tempo pode ter para garantir a estabilidade do sistema.

V. CONDIÇÕES DE FRONTEIRA

A tensão das extremidades da linha, na fonte e na carga, não podem ser calculados pelas equações (12) e (13). Sendo K o máximo valor de k pertencente a malha, e 0 o valor mínimo, para calcular esses valores seria necessário valores não pertencentes à malha. Então os valores de $v_K^{n+1/2}$ e $v_0^{n+1/2}$ devem ser definidos separadamente.

No caso da carga em curto-circuito:

$$v_K^{n+1/2} = 0 \qquad (19)$$

No caso da carga em aberto:

$$v_K^{n+1/2} = v_{K-1}^{n+1/2} \qquad (20)$$

Para calcular a influência de uma carga com outro valor e a da fonte, pode-se analisar o primeiro "nódo" da linha de transmissão, aplicando seu equivalente de Norton, como mostra a Fig. 2.

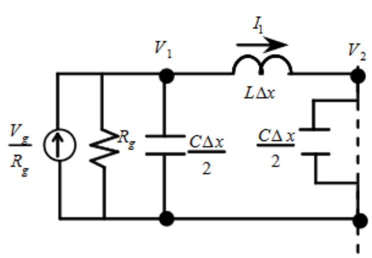

Figura 2. Equivalente de Norton do primeiro nódo da linha de transmissão[4].

Aplicando a LKC tem-se:

$$\frac{v_g^n}{R_g} = i_1^n + \frac{C\,\Delta x}{2}\frac{\partial v_l^n}{\partial t} + \frac{v_1^n}{R_g}$$

$$\frac{v_g^{n+1/2} + v_g^{n-1/2}}{2 R_g \Delta x} = \frac{i_1^n}{\Delta x} + C\,\frac{v_1^{n+1/2} - v_1^{n-1/2}}{2\,\Delta t} + \frac{v_1^{n+1/2} - v_1^{n-1}}{2}$$

VI. IMPLEMENTAÇÃO

O programa foi implementado como um script no Octave e é dividído em três partes. Primeiramente são definidas as constantes, como os dados da linha de transmissão e o passo de tempo e espaço utilizados. No caso da tensão da fonte e da carga no final da linha, o usuário deve indicar quais valores serão utilizados na caixa de diálogo inicial, como na Fig. 3.

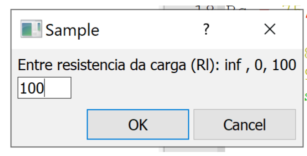

Figura 4. Caixa de escolha do valor da resistência.

A segunda parte consiste da loop principal da iteração. Aqui são calculados e atualizados os valores da corrente e tensão ao longo da linha.

```
for n = 2:Lt-1

  v(n,1) = (k1/k2)*v(n-1,1) - (Rs*i(n,1) - (Vs((n+1/2)*dt,m,uf,Z) + Vs((n-1/2)*dt,m,uf,Z))/2)/k2;

  #equacao de update

  #codigo vetorizado, muito mais rapido, linha sem perdas ,Gedney p43

  v(n,2:Lz-1) = v(n-1,2:Lz-1) - dt*(i(n,2:Lz-1) - i(n,1:Lz-2))/(C*dz);

  if (Rl == 0)

    v(n,Lz) = 0;

    i(n+1,Lz) = i(n+1,Lz-1);

  elseif (Rl == inf)

    v(n,Lz) = v(n,Lz-1);

    i(n+1,Lz) = 0;

  else

    v(n,Lz) = (k1/k2)*v(n-1,Lz) + Rl*i(n,Lz-1)/k2;

    i(n+1,Lz) = v(n,Lz)/Rl;

  endif

  i(n+1,1:Lz-1) = i(n,1:Lz-1) - dt*(v(n,2:Lz)-v(n,1:Lz-1))/(L*dz);

endfor
```

A terceira parte é responsável pela representação gráfica dos resultados. A janela priincipal apresenta os gráficos de tensão e corrente em cada posição da linha. Estes gráficos são atualizados no tempo e representam uma animação do comportamento das ondas. A janela secundária apresenta o tempo decorrido da simulação.

```
figure('Name',['Rl = ',num2str(Rl),', m = ', num2str(m)],'NumberTitle','off');

V = v(1,:);
```

```
I = i(1,:);

subplot(2,1,1);

plot (z, V,'ydatasource','V');

title('tensão');

xlabel("z (m)");

ylabel("v (V)");

axis([-Z 0 min(min(v)) max(max(v))]);

subplot(2,1,2);

plot (z, I,'ydatasource','I');

title('corrente');

xlabel("z (m)");

ylabel("i (A)");

axis([-Z 0 min(min(i)) max(max(i))]);

w = waitbar(0,'t = 0s');

M = round(p/100);#valor p teste

for n = 1:Lt/M#Lt/M para plotar 1 a cada M tempos,x = v(M*n,:),y = i(M*n,:)

  I = i(M*n,:);

  V = v(M*n,:);

  if (mod(n,10) == 0)

    s = num2str(M*n*dt);

    waitbar(M*n/Lt,w,['t = ',s,'s']);

  endif

  refreshdata

  drawnow

endfor

s = num2str(M*n*dt);

waitbar(1,w,['Finalizado ','t = ',s,'s']);

toc

clear
```

VII. RESULTADOS

AGRADECIMENTOS

Agradecimento especial para o Monitor PAE Luiz Felipe Machado Votto para o Prof. Dr. Leonardo André Ambrosio e para o colega de turma Alex Marcelino Santee e ao Prof. Dr. Stephen D. Gedney.

REFERÊNCIAS

[1] "Transmission line," *Wikipedia*. May 03, 2020, Accessed: May 19, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Transmission_line&oldid=954668391.

[2] F. Rana, "Time Domain Analysis of Transmission Lines," Cornell University, 2007, [Online]. Available: https://courses.cit.cornell.edu/ece303/Lectures/lecture24.pdf.

[3] L. Sevgi and C. Uluisik, "A MATLAB-based transmission-line virtual tool: finite-difference time-domain reflectometer," *IEEE Antennas Propagation Magazine*, vol. 48, no. 1, pp. 141–145, Feb. 2006, doi: 10.1109/MAP.2006.1645595.

[4] S. Gedney, *Computational Electromagnetics: The Finite-Difference Time-Domain*. University of Kentucky, 2003.

[5] "Lab 4 -- The FDTD Method." https://utah.instructure.com/courses/297816/assignments/1750259 (accessed May 23, 2020).