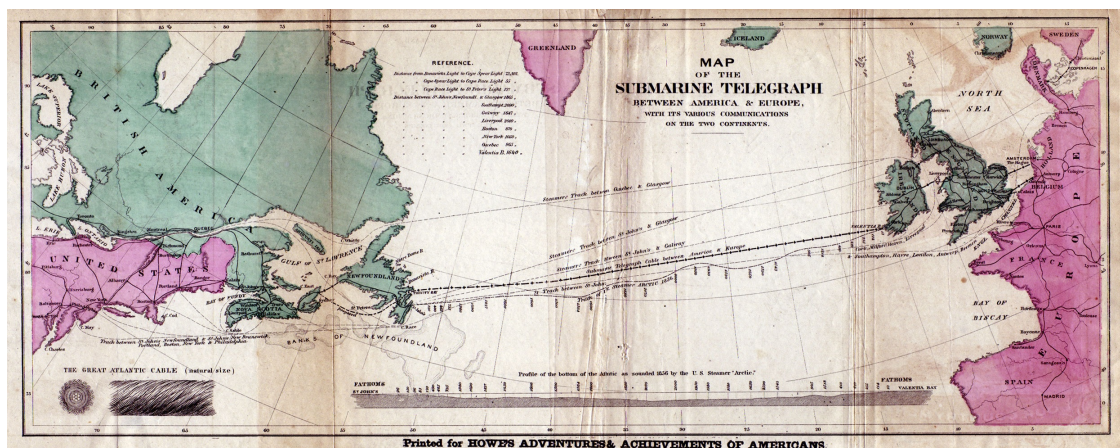


Sistemas Lineares-AtlanticCable

October 3, 2024

1 Linhas de Transmissão e o Primeiro Cabo Transatlântico de Comunicação

Neste Tutorial veremos como a teoria de sistemas lineares pode ser aplicada na análise de linhas de transmissão por meio da Transformada de Laplace. Como aplicação específica será discutido o primeiro cabo telegráfico transatlântico. O primeiro cabo telegráfico submarino foi instalado no leito do oceano atlântico norte por uma extensão de aproximadamente 3700km conectando Newfoundland no Canadá à Irlanda, em 1858. Tendo falhado após poucas semanas, o segundo e definitivo cabo foi instalado em 1865.



O primeiro estudo teórico da propagação de sinais elétricos em um cabo de grande extensão foi realizado por William Thompson, posteriormente elevado a nobreza e hoje conhecido como Lord Kelvin, em 1854. Thompson modelou uma linha telegráfica submersa como um longo fio condutor coberto por um isolante elétrico perfeito, e finalmente por outra camada condutora, em um arranjo conhecido hoje como cabo coaxial. O isolante utilizado foi a goma de gutta-percha, uma goma natural asiática com propriedades isolantes superiores a da borracha. A camada externa condutora era a própria água salgada do oceano. Devido a capacidade isolante do material utilizado, Thompson desprezou os efeitos de fuga de corrente, e portanto, a condutância de fuga no seu modelo matemático à parâmetros distribuídos do cabo. Além disso, Thompson também considerou o cabo como um elemento não indutivo (nas baixas velocidades então utilizadas na comunicação de sinais telegráficos o efeitos indutivos que dependem da frequência de chaveamento eram de fato desprezíveis), e portanto, em sua análise, $G = L = 0$, e a clássica equação da onda para modelos de linhas de transmissão e também conhecida como equação do telegrafista degenerava em um equação de difusão análoga a utilizada para modelagem de transporte de calor. No estudo a seguir, ao invés

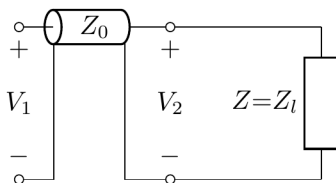
de lidar com estas equações no domínio do tempo, trabalharemos com a análise por meio da transformada de Laplace, já que o modelo de uma linha de transmissão baseada em elementos passivos lineares é também um sistema linear. A modelagem inicial será feita com o pacote de análise de circuitos lineares Lcapy, com o qual obteremos os principais resultados relativos a propagação de sinais no cabo como modelado por William Thompson.

Inicialmente importamos Lcapy:

```
[1]: from lcapy import *
```

Uma linha de transmissão geral pode ser descrita como mostrado abaixo. Aqui a entrada é aberta, e a linha alimenta uma carga genérica Z_l . Além disso, a impedância característica da linha Z_0 , a constante complexa de propagação γ e o comprimento da linha x são deixados sem ser especificados.

```
[2]: clt=Circuit("""
TL 3 0_3 2 0_2,Z_0, gamma, x; right
W 3 4; right=0.5
Z 4 0_4, Z_l; down
W 0_3 0_4; right
P2 3 0_3; down, v=V_2
P1 2 0_2; down, v=V_1
; draw_nodes=connections, label_nodes=None""")
clt.draw()
```



A impedância de entrada - *driving-point impedance* - nos terminais de entrada pode então ser obtida:

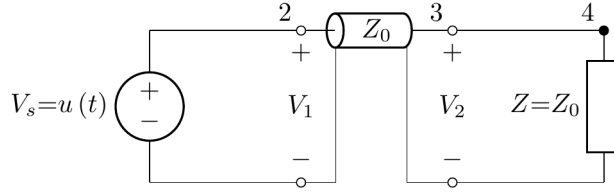
```
[3]: clt.P1.dpZ.simplify()
```

```
[3]:
```

$$\frac{Z_0 (Z_0 \sinh(\gamma x) + Z_l \cosh(\gamma x))}{Z_0 \cosh(\gamma x) + Z_l \sinh(\gamma x)}$$

Para o cabo transatlântico consideramos uma linha infinita e para este fim fazemos $Z_l = Z_0$. Além disso, inserimos uma fonte de degrau de tensão $u(t)$ e a conectamos com o circuito descrito acima.

```
[4]: clt.add('W 1 2;right')
clt.add('W 0 0_2;right')
clt.add('Vs 1 0 {u(t)};down ')
clt=clt.subs({'Z_l':'Z_0'})
clt.draw();
```



A tensão em um ponto x em uma linha de transmissão no domínio de Laplace é

$$V(x, s) = K_1 e^{-\gamma x} + K_2 e^{\gamma x}$$

onde $\gamma = \sqrt{(R + sL)(G + sC)}$, a constante complexa de propagação. Por outro lado, a linha é infinita e portanto a componente refletida é nula, ou seja

$$V(x, s) = K_1 e^{-\gamma x}$$

Desde que em $x=0$ temos $V(0, s) = K_1 = \mathcal{L}(u(t)) = \frac{1}{s}$,

$$V(x, s) = \frac{e^{-\gamma x}}{s}$$

Este resultado é confirmado tomando a tensão nos terminais da carga como mostrado abaixo:

[5]: `clt.P2.V`

[5]:

$$\frac{e^{-\gamma x}}{s}$$

No modelo de Thompson $\gamma = \sqrt{RCs}$, portanto

[6]: `Vs=clt.P2.V.subs({'gamma':'sqrt(R*C*s)'}
Vs`

[6]:

$$\frac{e^{-\sqrt{C}\sqrt{R}\sqrt{s}x}}{s}$$

cuja transformada inversa é (para obter a transformada inversa é suficiente acrescentar “(t)” ao final da expressão no domínio s):

[7]: `v=Vs(t).simplify()
v`

[7]:

$$\left(1 - \operatorname{erf}\left(\frac{\sqrt{C}\sqrt{R}x}{2\sqrt{t}}\right)\right) u(t)$$

Onde $erf(x)$ é a função erro de x . A grandeza de interesse aqui é a corrente no receptor, e assim tensão V_s é dividida pela impedância $Z(s)$

```
[8]: Z=impedance('sqrt(R/(C*s))')
Vs/Z
```

[8]:

$$\frac{\sqrt{C}e^{-\sqrt{C}\sqrt{R}\sqrt{s}x}}{\sqrt{R}s\sqrt{\frac{1}{s}}}$$

Para a calcular transformada inversa de laplace da corrente simplificamos o denominador da equação acima como mostrado na próxima célula:

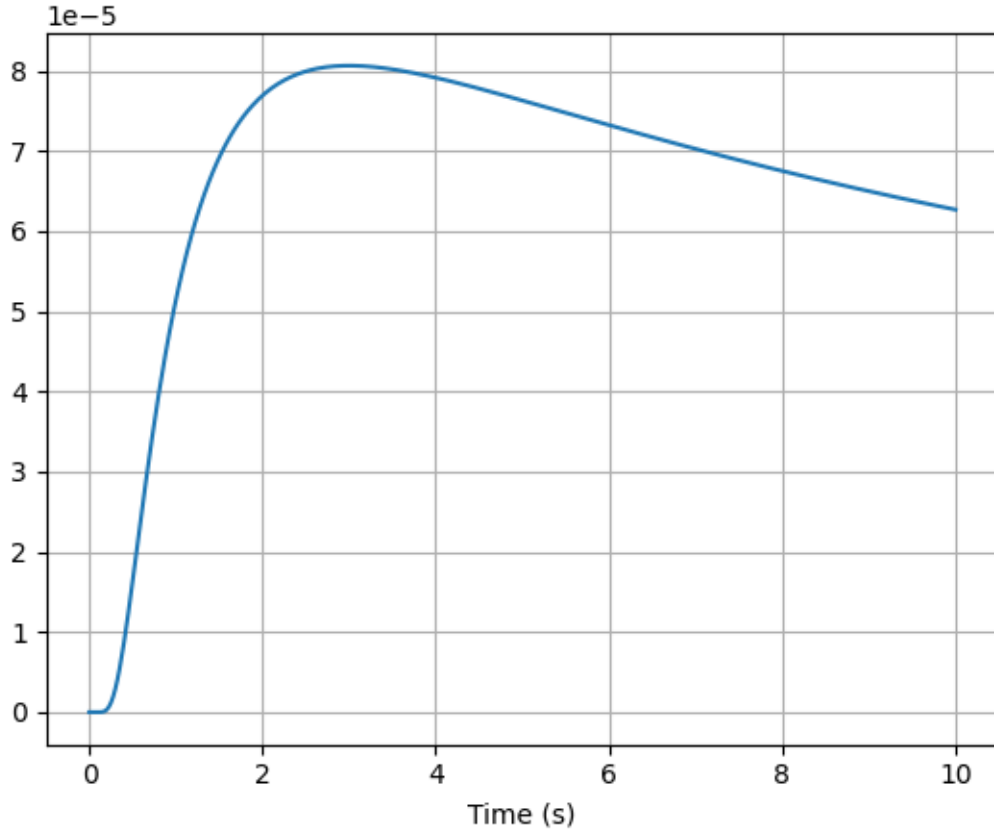
```
[9]: R,C,x = symbols("R, C,x")
i=(expr(sqrt(C/R)*exp((-sqrt(R*C))*sqrt(s)*x))/sqrt(s))(t)
i
```

[9]:

$$\frac{\sqrt{C}e^{-\frac{C R x^2}{4t}}u(t)}{\sqrt{\pi}\sqrt{R}\sqrt{t}} \text{ for } t \geq 0$$

Os valores padrão de resistência e capacitância por quilômetro eram $R = 1,62 \text{ } \Omega/Km$ e $C = 2,7 \cdot 10^{-7} \text{ } F/Km$. Substituindo estes valores em $i(t)$ podemos calcular a corrente que seria detectada em um galvanômetro no receptor. A corrente como função do tempo no terminal receptor, 3704 Km do transmissor é então:

```
[10]: i.subs({'x':3704,'R':1.62,'C':2.7e-7}).plot((0.001,10));
```



Esta é a resposta em corrente devido à um degrau unitário de tensão, na qual pode ser observado um atraso de aproximadamente 3 segundos para a corrente atingir seu pico no receptor, em relação ao instante em que ocorre o degrau. Este atraso era responsável por impor limites severos na velocidade com que sinais poderiam ser enviados através da linha, como será discutido mais adiante. Para uma tensão aplicada de 1 V, o pico da corrente recebido atingia um valor em torno dos 80 Miliampères, e na realidade as tensões utilizadas na entrada eram de algumas dezenas de Volts. Assim, a corrente recebida poderia ter seus valores de pico identificados por um galvanômetro de alta sensibilidade desenvolvido por Thompson, conhecido como galvanômetro de espelho, o qual detectava uma corrente elétrica refletindo um feixe de luz com um espelho. O feixe de luz projetado em uma escala atuava como um ponteiro longo sem massa, portanto, com alta sensibilidade (ver figura mais abaixo). Além disso, ao invés do código Morse padrão baseado nos símbolos representados pelos sinais ponto-traço comumente utilizados em telegrafia, o cabo transatlântico era operado com pulsos de polaridades invertidas para facilitar a identificação de símbolos diferentes na recepção.

Uma característica importante do cabo atlântico tal como modelado por Thompson era a distorção de fase produzida durante a propagação. Sabemos que uma entrada arbitrária pode ser representada por uma soma infinita de harmônicas senoidais. Considere uma harmônica com determinada frequência ω_0 . Para esta harmônica, $\gamma = \sqrt{(R + j\omega_0 L)(C + j\omega_0 C)} = \sqrt{j}\sqrt{\omega_0 RC} = \frac{1+j}{\sqrt{2}}\sqrt{\omega_0 RC}$.

Assim, $\alpha = \beta = \sqrt{\frac{\omega_0 RC}{2}}$ e substituindo na solução da equação da onda, $E_0 e^{-\alpha x} \cos(\omega_0 t - \beta x)$, onde

E_0 é a amplitude da harmônica com frequência ω_0 , temos

$$E(x, t) = E_0 e^{-\sqrt{\frac{\omega_0 RC}{2}} x} \cos(\omega_0 t - \sqrt{\frac{\omega_0 RC}{2}} x)$$

Como pode ser visto, a atenuação da tensão é dependente da frequência, da mesma forma que a velocidade de propagação, a qual é dada por $v_f = \frac{\omega_0}{\beta} = \sqrt{\frac{2\omega_0}{RC}}$, ou seja, harmônicas diferentes propagam-se com velocidades diferentes espalhando e distorcendo o sinal. Como resultado, dois pulsos diferentes podem ser sobrepostos, causando o que é conhecido como interferência intersimbólica. Para prevenir este efeito a velocidade de chaveamento deveria ser baixa o suficiente para que pulsos diferentes fossem claramente identificáveis na recepção.

Outra característica presente na modelagem realizada por Thompson é discutida a seguir. Precisamos obter uma forma de calcular a resposta de corrente para uma entrada arbitrária, a qual pode ser obtida com a integral de convolução entre a tensão arbitrária de entrada e a resposta ao impulso para corrente $h(t)$. Sabendo que i acima é a resposta de corrente à um degrau unitário, e que a derivada temporal do degrau é o impulso $\delta(t)$, conseguimos a resposta ao impulso com a derivada temporal de i , ou seja:

```
[11]: h=diff(i,t)
      h
```

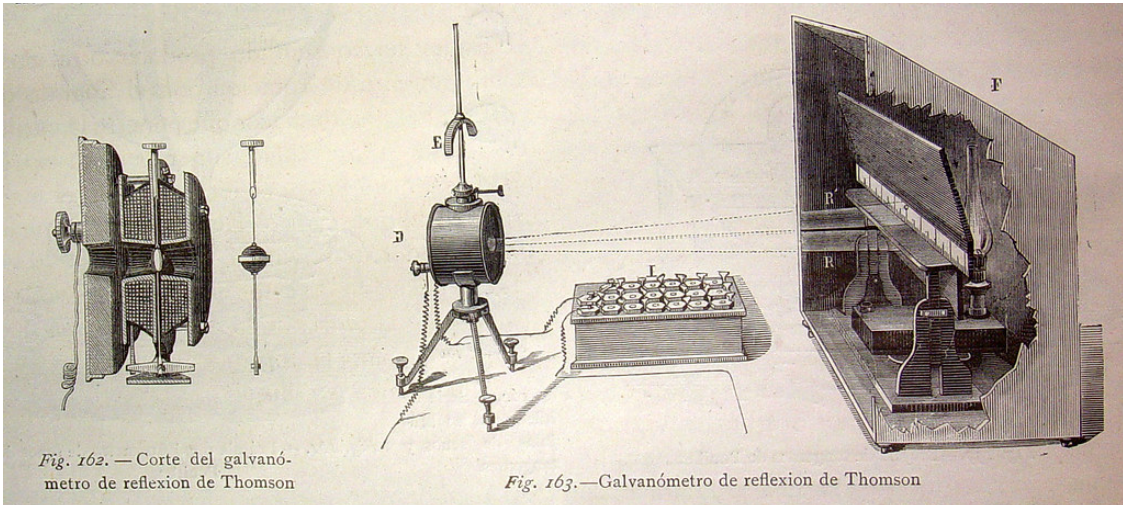
[11]:

$$\frac{C^{\frac{3}{2}} \sqrt{R} x^2 e^{-\frac{CRx^2}{4t}} u(t)}{4\sqrt{\pi} t^{\frac{5}{2}}} + \frac{\sqrt{C} e^{-\frac{CRx^2}{4t}} \delta(t)}{\sqrt{\pi} \sqrt{R} \sqrt{t}} - \frac{\sqrt{C} e^{-\frac{CRx^2}{4t}} u(t)}{2\sqrt{\pi} \sqrt{R} t^{\frac{3}{2}}} \text{ for } t \geq 0$$

Igualando a expressão acima (desconsiderando o termo impulsivo) a zero, e resolvendo para t , encontramos o temp t_{max} onde a corrente atinge seu pico, isto é,

$$t_{max} = \frac{1}{2} RCx^2.$$

A dependência do valor de pico da corrente com o quadrado da distância era conhecido como a Lei dos Quadrados de Thompson. Segundo esta lei, dobrando a distância quadruplicaria o atraso na corrente de pico, o que por sua vez também levava a redução da máxima taxa de transmissão.



Fizemos acima o gráfico da corrente no receptor com uma entrada degrau no transmissor com o pacote Lcapy, que é baseado no pacote Sympy de cálculo simbólico para o ecossistema Python. Se queremos analisar a resposta para outras entradas, devemos agora usar os resultados obtidos acima mas fazer os cálculos e gráficos com os pacotes numérico e gráfico Numpy e Matplotlib. Usando Numpy em python, implementamos a resposta ao impulso obtida previamente e plotamos o gráfico no código abaixo. Inicialmente definimos os parâmetros da linha:

```
[12]: import numpy as np
import matplotlib.pyplot as plt

tmin=1e-15
tmax=10
ndiv=10000

R=1.62
C=2.7e-7
x=3704
```

t é o arranjo que representa o tempo discretizado, e e_i é a entrada que inicialmente é definida como um degrau unitário. Definimos então h_1 e h_2 como as duas parcelas não impulsivas da resposta à entrada degrau calculadas acima.

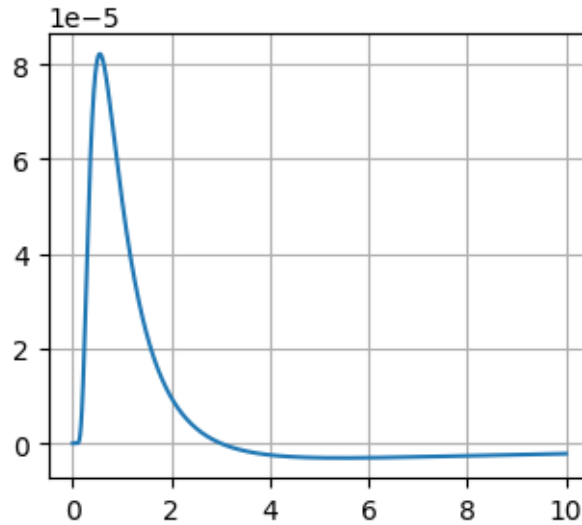
```
[13]: t=np.linspace(tmin,tmax,ndiv)
deltat=(tmax-tmin)/ndiv
```

```
[14]: e_i=np.ones(len(t))
```

```
[15]: exp=(x**2)*R*C/4
f1=((x**2)*(C**1.5)/4)*np.sqrt(R/np.pi)
f2=np.sqrt(C/(R*np.pi))/2
h1=np.exp(-exp/t)/(t**2.5)
h2=np.exp(-exp/t)/(t**1.5)
```

Finalmente o gráfico da resposta h_1-h_2 é feito com o *script* de código abaixo:

```
[16]: fig = plt.figure(figsize=(8,3))
ax1=plt.subplot(1,2,1)
ax1.plot(t,f1*h1[:ndiv]-f2*h2[:ndiv])
ax1.grid()
```

Agora, para simular a transmissão real de uma mensagem baseada na codificação de Morse modificada como discutimos acima, utilizaremos pulsos unitários positivos e negativos representando traços e pontos. Cada pulso tem 1 s de duração, e os pulsos são separados por um nulo de 1 s. Separando os caracteres transmitidos, é enviado um nulo de 2 s de duração de modo que caracteres diferentes possam ser separados na recepção. No código abaixo a entrada é definida como a sequência de dois caracteres.

```
[17]: tmax=20
t=np.linspace(tmin,tmax,ndiv)
deltat=(tmax-tmin)/ndiv

e_i=np.zeros(len(t))
for k in range(len(t)):
    if t[k]<1:
        e_i[k]=1
    elif (t[k]>2 and t[k]<3):
        e_i[k]=-1
    elif (t[k]>4 and t[k]<5):
        e_i[k]=-1
    elif t[k]>6 and t[k]<7:
        e_i[k]=1
    elif t[k]>8 and t[k]<9:
        e_i[k]=-1
    elif t[k]>11 and t[k]<12:
        e_i[k]=1
    elif t[k]>13 and t[k]<14:
        e_i[k]=1
    elif t[k]>15 and t[k]<16:
        e_i[k]=-1
```



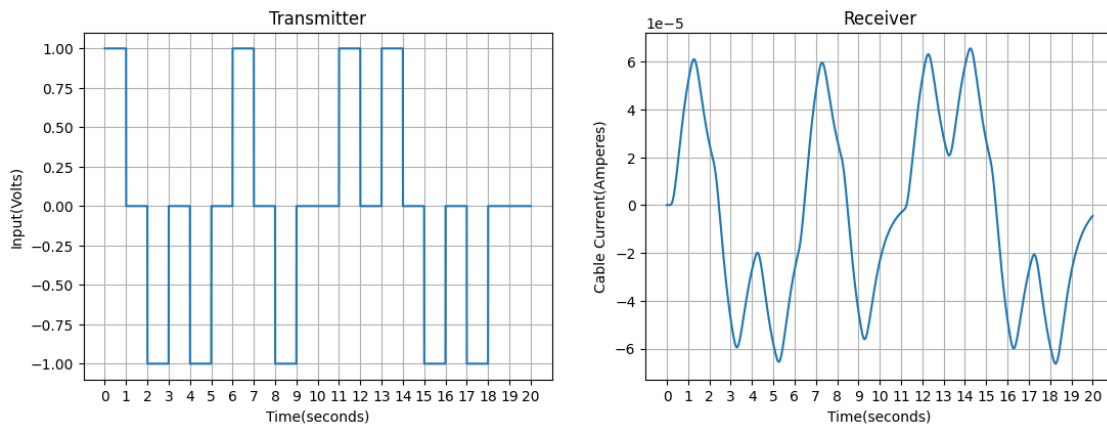
```
elif t[k]> 17 and t[k]<18:
    e_i[k]=-1
```

```
h1=np.exp(-exp/t)/(t**2.5)
h2=np.exp(-exp/t)/(t**1.5)
```

A corrente no receptor é obtida como a convolução numérica entre a entrada implementada acima e a resposta ao impulso, e depois, os pulsos de entrada e a corrente de saída são plotados.

```
[18]: i=(f1*np.convolve(e_i,h1)-f2*np.convolve(e_i,h2))*deltat
```

```
[19]: fig = plt.figure(figsize=(13,4.4))
ax1=plt.subplot(1,2,1)
ax1.plot(t,e_i[:ndiv])
ax1.set_xticks(np.arange(0,tmax+1))
ax1.set_title('Transmitter')
ax1.set_xlabel('Time(seconds)')
ax1.set_ylabel('Input(Volts)')
ax1.grid()
ax2=plt.subplot(1,2,2)
ax2.plot(t,i[:ndiv])
ax2.set_xticks(np.arange(0,tmax+1))
ax2.set_title('Receiver')
ax2.set_xlabel('Time(seconds)')
ax2.set_ylabel('Cable Current(Amperes)')
ax2.grid()
```



O gráfico acima à direita mostra que os caracteres transmitidos podem ser corretamente identificados na recepção.