Control Systems Assignment-1
Problem 17

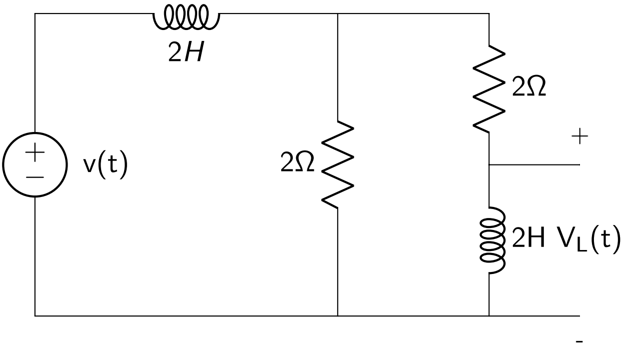
Manjunath Kakkireni
EE19BTECH11015

September 10, 2020

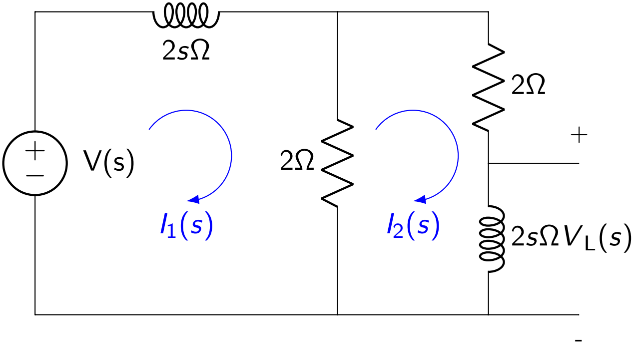
Overview

Problem a

Find transfer function G(s)=V_L(s)/V(s) for the network given below.



Solution: For the given network, equivalent circuit in S domain can be obtained by Laplace Transform.



Resultant Circuit

Apply Kirchhoff's voltage law to each closed loop.
Mesh-1:

$$V(s) - 2sI_1(s) - 2(I_1(s) - I_2(s)) = 0$$
$$(2s+2)I_1 - 2I_2(s) = V(s) \tag{1}$$

Mesh-2:

$$-2(I_2(s) - I_1(s)) - 2I_2(s) - 2sI_2(s) = 0$$
$$-2I_1(s) + (2s+4)I_2(s) = 0$$
$$-I_1(s) + (s+2)I_2 = 0 \tag{2}$$

Solve 1 and 2 by using Cramer's rule, y $=D_y/D$

$$I_2(s) = \frac{\begin{vmatrix} 2(s+1) & V(s) \\ -1 & 0 \end{vmatrix}}{\begin{vmatrix} 2(s+1) & -2 \\ -1 & s+2 \end{vmatrix}}$$

$$I_2(s) = \frac{V(s)}{2s^2 + 6s + 2}$$

From the circuit,

$$V_L(s) = I_2(s) * 2s$$

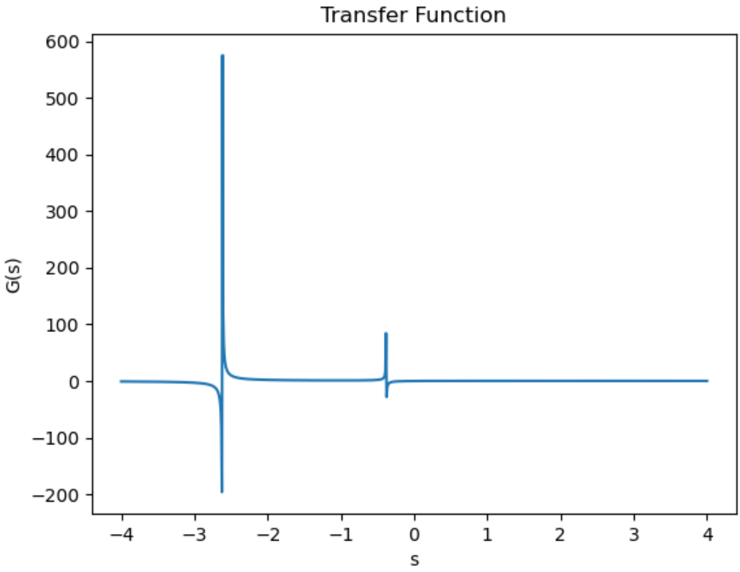
$$V_L(s) = \frac{V(s)}{2s^2 + 6s + 2} * 2s$$

$$V_L(s) = \frac{V(s) * s}{s^2 + 3s + 1}$$

$$\frac{V_L(s)}{V(s)} = \frac{s}{s^2 + 3s + 1}$$

Therefore, transfer function G(s) of the given network is equal to

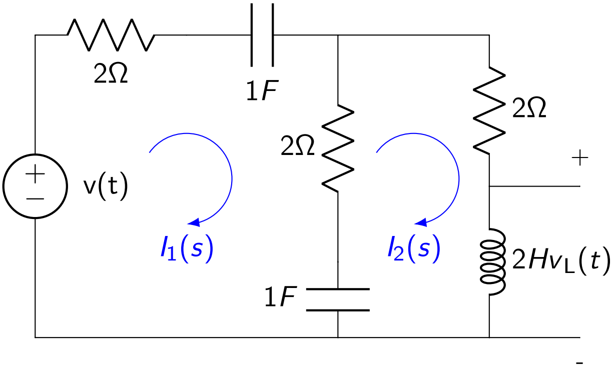
$$\frac{s}{s^2 + 3s + 1}$$


Transfer Function
600
500
400
300
200
100
0
−100
−200
G(s)
−4
−3
−2
−1
0
1
2
3
4
s

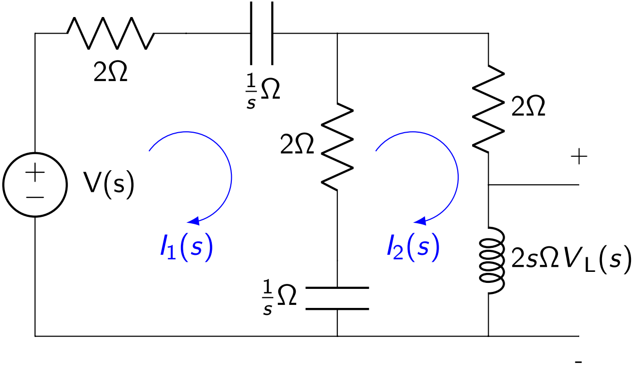
```python
import sympy
import numpy as np
import matplotlib.pyplot as plt
s,I_1,I_2,V,V_L =sympy.symbols("s I_1 I_2 V V_L") #declare variables
#Mesh Analysis
equation_1=sympy.Eq((2*s+2)*I_1 - 2*I_2,V)#Apply KVL around first mesh
equation_2=sympy.Eq(-2*I_1 + 2*(s+2)*I_2,0)#Apply KVL around second mesh
#solve equations 1 & 2 to find I_1 and I_2
solution_12=sympy.solve((equation_1, equation_2),(I_1,I_2))
equation_3=sympy.Eq(I_2,solution_12[I_2])#Assign the found out value to I_2
equation_4=sympy.Eq(V_L,2*s*I_2)#voltage across inductor(V_L)=2s*I_2
#solve equations 3 & 4 to find V and V_L
solution_34=sympy.solve((equation_3,equation_4),V,V_L)
G=solution_34[V_L]/solution_34[V] #divide V_L by V to get transfer function (G(s))
print("Transfer Function(G(s))="+str(G))

#plotting transfer function
x=np.linspace(-4,4, 1001)#1001 evenly spaced points on x axis
f = sympy.lambdify(s,G)#function to map from x to y
y=f(x) #array of y values of corresponding x values
plt.plot(x,y)
plt.title("Transfer Function")
plt.xlabel("s")
plt.ylabel("G(s)")
plt.show()
```

Find transfer function G(s)=$V_L(s)/V(s)$ for the network given below.



Solution: For the given network, equivalent circuit in S domain can be obtained by Laplace Transform.



Resultant Circuit

Apply Kirchhoff's voltage law to each closed loop.

Mesh-1:

$$V(s) - 2sI_1(s) - \frac{I_1(s)}{S} - 2(I_1(s) - I_2(s)) - \frac{I_1(s) - I_2(s)}{s} = 0$$

$$(4 + \frac{2}{s})I_1(s) - (2 + \frac{1}{s})I_2(s) = V(s) \tag{3}$$

Mesh-2:

$$-\frac{1}{s}(I_2(s) - I_1(s)) - 2(I_2(s) - I_1(s)) - 2I_2(s) - 2sI_2(s) = 0$$

$$-(2 + \frac{1}{s})I_1(s) + (2s + \frac{1}{s} + 4)I_2(s) = 0 \tag{4}$$

Solve 3 and 4 by using Cramer's rule, y $=D_y/D$

$$I_2(s) = \frac{\begin{vmatrix} 4+2/s & V(s) \\ -(2+1/s) & 0 \end{vmatrix}}{\begin{vmatrix} 4+2/s & -(2+1/s) \\ -(2+1/s) & 2s+1/s+4 \end{vmatrix}}$$

$$I_2(s) = \frac{V(s)*s}{4s^2+6s+1}$$

From the circuit,

$$V_L(s) = I_2(s)*2s$$

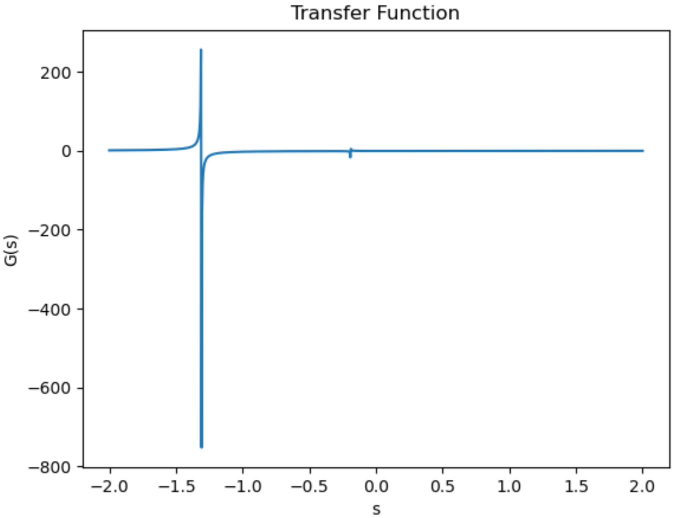
$$V_L(s) = \frac{V(s)*s}{4s^2+6s+1}*2s$$

$$V_L(s) = \frac{V(s)*2s^2}{4s^2+6s+1}$$

$$\frac{V_L(s)}{V(s)} = \frac{2s^2}{4s^2+6s+1}$$

Therefore, transfer function G(s) of the given network is equal to

$$\frac{2s^2}{4s^2 + 6s + 1}$$



```python
#import necessary libraries
import sympy
import numpy as np
import matplotlib.pyplot as plt
s,I_1,I_2,V,V_L=sympy.symbols("s I_1 I_2 V V_L") #declare variables

#Mesh Analysis
equation_1=sympy.Eq(((4+ 2/s)*I_1 - (2+1/s)*I_2),V)#Apply KVL around first mesh
equation_2=sympy.Eq(-(2+1/s)*I_1 + (2*s + 1/s + 4)*I_2,0)#Apply KVL around second mesh
#solve equations 1 & 2 to find I_1 and I_2
solution_12=sympy.solve((equation_1,equation_2),(I_1,I_2))
equation_3=sympy.Eq(I_2,solution_12[I_2])#Assign the found out value to I_2
equation_4=sympy.Eq(V_L,2*s*I_2)#voltage across inductor is equal to 2s*I_2
#solve equations 3 and 4 to find V_L and V
solution_34=sympy.solve((equation_4,equation_3),(V_L,V))
G=solution_34[V_L]/solution_34[V] #dividing V_L by V
G=sympy.simplify(G)
print("Transfer Function(G(s))="+str(G))

#plotting transfer function
x=np.linspace(-2,2, 1001)#1001 evenly spaced points on x axis
f = sympy.lambdify(s,G)#function to map from x to y
y=f(x) #array of y values of corresponding x values
plt.plot(x,y)
plt.title("Transfer Function")
plt.xlabel("s")
plt.ylabel("G(s)")
plt.show()
```