

Alexander & Sadiku Example Problem 10.6

Michael Gustafson

> restart

Handy functions for dealing with phasors

```
> j := I:  
> polard := (mag, angd) → polar(mag, angd*Pi / 180) :  
> argumentd := (num) → argument(num) * 180 / Pi :  
> listphasors := proc(plist) local k  
for k from 1 to nops(plist[ ]) do  
printf("%s = %f < %f deg\n", lhs(plist[ ][k]), evalc(abs(rhs(plist[ ][k]))),  
evalc(argumentd(rhs(plist[ ][k]))))  
end do end proc:
```

Circuit equations

```
> KCLn2 :=  $\frac{(Vn2 - Va)}{j\omega L} - Ib + \frac{Vo}{R1} = 0 :$   
> KCLn3 :=  $-\frac{Vo}{R1} + j\omega C(Vn2 - Vo) + \frac{(Vn2 - Vo - Vc)}{R2} = 0 :$ 
```

Solve circuit equations

```
> MySoln := solve({KCLn2, KCLn3}, [Vn2, Vo]) :  
> collect(MySoln, jomega)
```

$$\begin{aligned} Vn2 &= (C Ib LR1 R2 jomega^2 + (CR1 R2 Va + Ib LR1 + Ib LR2 + L Vc) jomega \\ &\quad + R1 Va + R2 Va) / (CLR2 jomega^2 + (CR1 R2 + L) jomega + R1 + R2), \quad (1) \\ &= \frac{R1 (C Ib LR2 jomega^2 + (CR2 Va + Ib L) jomega + Va - Vc)}{CLR2 jomega^2 + (CR1 R2 + L) jomega + R1 + R2} \end{aligned}$$

Define lists for elements, then for each frequency independently

```
> ElVals := R1 = 1, R2 = 4, L = 2, C = 0.1 :  
> Valsa := ElVals, jomega = j·2, Va = polard(10, 0), Ib = 0, Vc = 0 :  
> Valsb := ElVals, jomega = j·5, Va = 0, Ib = polard(2, -90), Vc = 0 :  
> Valsc := ElVals, jomega = j·0, Va = 0, Ib = 0, Vc = polard(5, 0) :
```

Find solutions for each frequency

```
> MySolna := subs(Valsa, MySoln) :  
> MySolnb := subs(Valsb, MySoln) :  
> MySolnc := subs(Valsc, MySoln) :
```

Find phasors for each frequency

```
> listphasors(MySolna)  
Vn2 = 9.877484 < -60.353678 deg  
Vo = 2.498097 < -30.784147 deg  
> listphasors(MySolnb)  
Vn2 = 5.606810 < -119.538782 deg  
Vo = 2.328101 < -77.905243 deg  
> listphasors(MySolnc)
```

Vn2 = 0.000000 < 0.000000 deg
Vo = 1.000000 < 180.000000 deg

=>

Conclusion:

$$vo(t) = 2.498 \cos(2 t - 30.78 \text{ deg}) + 2.328 \cos(5 t - 77.91 \text{ deg}) - 1$$