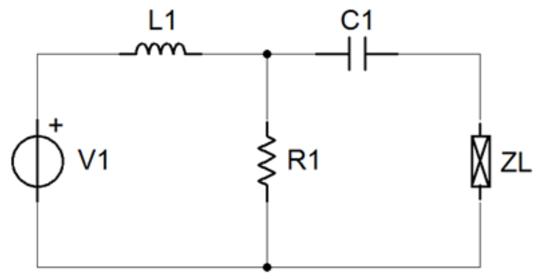
PROBLEM #1



In the above circuit, the voltage source V_1 is $75V_{eff}$ at a frequency of 400Hz. The other circuit components are R1=820Ω, L1=33mH, and C_1=470nF.

a) What value of ZL will result in maximum average power delivered to it?

b) What will the complex power delivered to the load in part (a) be?

c) What percentage of the real power delivered by the source will be absorbed by the load?

d) If the load is to be constructed using just two components from the list of components in Appendix H, draw a circuit for ZL that will absorb the maximum power.

e) What fraction of the maximum average power that could potentially be delivered to the load will be delivered to the circuit in part (d)?

A PYTHON SCRIPT WAS WRITTEN TO PERFORM THE NUMERICAL COMPUTATIONS TO SAVE EFFORT AND REDUCE ERRORS. BOTH THE HOMEWORK SPECIFIC SCRIPT, HW02.PY, AND THE SCRIPT THAT DEFINES GENERIC FUNCTIONS, Z.PY, ARE ATTACHED.

THE OUTPUT OF THE SCRIPT FOLLOWS THE WORK AND IS CROSS-REFERENCED TO THE WORK FOR EASY REFERENCE.

PART (a)

FOR MAX. POWER TRANSFER, WANT

$$\bar{Z}_{LOAD} = \bar{Z}_{TH}^*$$

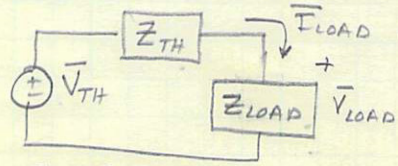
$$\bar{Z}_{TH} = \bar{Z}_{C1} + (\bar{Z}_{R1} \| \bar{Z}_{L1}) \quad \text{(BY INSPECTION)}$$

① $$= 8.30\Omega - j765\Omega$$

$$\bar{Z}_{LOAD} = 8.30\Omega + j765\Omega$$ ← (#1a)

PROBLEM #1 (CONT'D)

PART (b)



$Z_{TH} = R_{TH} + j X_{TH}$

$$P_{LOAD} = Re\{\bar{V}_{LOAD} \cdot \bar{I}^*_{LOAD}\}$$

$$\bar{V}_{LOAD} = \frac{\bar{V}_{TH} \cdot \bar{Z}_{LOAD}}{(\bar{Z}_{TH} + \bar{Z}_{LOAD})}$$

$$\bar{I}^*_{LOAD} = \frac{\bar{V}^*_{TH}}{(Z_{TH} + \bar{Z}_{LOAD})^*}$$

$$\bar{S}_{LOAD} = \bar{V}_{LOAD} \cdot \bar{I}_{LOAD}$$

$$= \frac{(\bar{V}_{TH} \cdot \bar{V}^*_{TH})\ \bar{Z}_{LOAD}}{(Z_{TH} + Z_{LOAD})(Z_{TH} + Z_{LOAD})^*} = \frac{|\bar{V}_{TH}|^2}{|\bar{Z}_{TH} + \bar{Z}_{LOAD}|^2} \cdot \bar{Z}_{LOAD}$$

IF $\bar{Z}_{LOAD} = \bar{Z}^*_{TH}$,

$$\bar{Z}_{TH} + \bar{Z}_{LOAD} = R_{TH} + j X_{TH} + R_{TH} - j X_{TH} = 2R_{TH}$$

$$\bar{S}_{LOAD} = \frac{V_{TH}^2}{(2R_{TH})^2} \cdot (R_{TH} - j X_{TH})$$

(2) $\bar{S}_{LOAD} = (167.6 + j1543)\,VA$ ← (#1b)

PART (c)

$$P_{LOAD} = Re\{\bar{S}_{LOAD}\} = 167.6\,W$$

$$P_{SOURCE} = Re\{\bar{S}_{SOURCE}\}$$

$$\bar{S}_{SOURCE} = \bar{V}_{SOURCE} \cdot \bar{I}^*_{SOURCE}$$

$$\bar{I}_{SOURCE} = \frac{\bar{V}_{SOURCE}}{\bar{Z}_{TOT}}$$

$$\bar{Z}_{TOT} = Z_{L1} + (\bar{Z}_{R1} \| (\bar{Z}_{C1} + \bar{Z}_L))$$ (BY INSPECTION)

(3) $\bar{Z}_{TOT} = (16.12 + j3.26)\,\Omega$

(4) $\bar{S}_{SOURCE} = (335 + j67.8)\,VA$

$$P_{SOURCE} = Re\{\bar{S}_{SOURCE}\} = 335\,W$$

$$\text{POWER TRANSFER} = \frac{P_{LOAD}}{P_{SOURCE}} = \frac{167.6\,W}{335\,W}$$

POWER TRANSFER = 50.0% ← (#1c)

NOTE: YOU CANNOT USE V_{TH} FOR THIS, BUT MUST USE THE ORIGINAL CIRCUIT.

PROBLEM #1 (CONT'D)

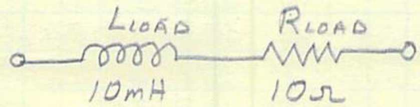
PART (d)

THE OPTIMAL LOAD IS $\bar{Z}_{LOAD} = 8.30\Omega + j765\Omega = R_{TH} + jX_{TH}$

(5) $R = 8.30\Omega$ $L = \frac{1}{2\pi f X_{TH}} = 304\text{mH}$ (6)

THE SMALLEST AVAILABLE RESISTOR IS 10Ω WHILE THE LARGEST AVAILABLE INDUCTOR IS 10mH.

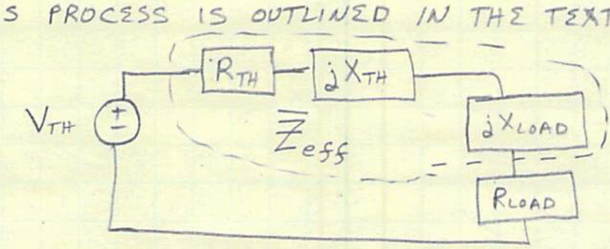
A NAIVE ATTEMPT WOULD USE THE COMPONENTS AS BEING THE CLOSEST MATCH, YIELDING THE FOLLOWING LOAD:



$\bar{Z}_{LOAD} = (10 + j25.1)\Omega$

HOWEVER, THE PROPER METHOD IS TO CHOOSE X_{LOAD} SO AS TO OFFSET AS MUCH OF X_{TH} AS POSSIBLE AND THE SET R_{LOAD} EQUAL TO THE MAGNITUDE OF THE REST OF THE EFFECTIVE COMPONENT'S IMPEDANCE.

THIS PROCESS IS OUTLINED IN THE TEXT (SEC 10.6 - p388)



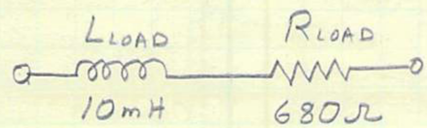
$R_{LOAD} = |\bar{Z}_{eff}| = |Z_{TH} + jX_{LOAD}|$

$= \sqrt{R_{TH}^2 + (X_{TH} + X_{LOAD})^2}$

(7) $R_{LOAD} = 739\Omega$ (CHOICE OF 680Ω or 1000Ω)

✓ 680Ω IS $\frac{(680\Omega - 739\Omega)}{739\Omega} = \frac{-59}{739} = 8.0\%$ TOO SMALL

1000Ω IS $\frac{(1000\Omega - 739\Omega)}{739\Omega} = \frac{261}{739} = 35.3\%$ TO BIG



$\bar{Z}_{LOAD} = (680 + j25.1)\Omega$ ← (#1d)

PROBLEM #1 (CONT'D)

PART (e)

THE MAXIMUM POTENTIAL POWER THAT COULD BE DELIVERED TO THE LOAD IS WHEN THE OPTIMAL LOAD, $\bar{Z}_{TH}^*$, IS USED. THUS, P_{MAX} IS THE RESULT GIVEN IN PARTS (b) & (c).

$$P_{MAX} = 167.6\text{ W}$$

THE THEVENIN EQUIVALENT CIRCUIT IN PART (b) CAN BE USED WITH THE NON-IDEAL LOAD AND THE P_{AVG} THAT RESULTS CAN BE DIVIDED BY P_{MAX} ABOVE.

$$P_{AVG} = Re\{\bar{S}_{LOAD}\}$$

$$= Re\{\bar{V}_{LOAD} \cdot \bar{I}_{LOAD}^*\}$$

$$P_{AVG} = Re\{(\bar{I}_{LOAD} \cdot \bar{Z}_{LOAD})\bar{I}_{LOAD}^*\}$$

$$\left(w/\ \bar{I}_{LOAD} = \frac{\bar{V}_{TH}}{\bar{Z}_{TH} + \bar{Z}_{LOAD}}\right)$$

FOR THE NAIVE ATTEMPT ($R_{LOAD} = 10\Omega$), THE RESULT IS

(8)

$$\bar{S}_{LOAD} = (101.8 + j256)\text{ mVA}$$

$$P_{AVG} = 0.0607\%\text{ OF } P_{MAX}$$

FOR THE BEST AVAILABLE LOAD ($R_{LOAD} = 680\Omega$):

(9)

$$\bar{S}_{LOAD} = (3.711 + j0.137)\text{ VA}$$

$$P_{AVG} = 2.21\%\text{ OF } P_{MAX}$$ ← (#1e)

FOR COMPLETENESS, THE RESULTS WERE ALSO RUN FOR 739Ω AND 1000Ω:

R_{LOAD}	P_{AVG}	% P_{MAX}
739Ω	3.724 W	2.22%
1000Ω	3.562 W	2.12%

THUS, THE 680Ω IS CLEARLY THE BETTER CHOICE AND IS VERY CLOSE THE THE BEST PERFORMANCE WITH THE 10mH CONSTRAINT. ALSO, NOTE THAT IT ABSORBS MORE THAN 36x THE POWER AS WOULD THE 10Ω RESISTOR.

Output from HW02.py script:

```
    Thevenin Equivalent Circuit
    Vth = 74.24-7.509j V = (74.62 @ -5.775 deg V)
(1) Zth = 8.304-764.5j ohms

    Optimal load for max power transfer
    Z load   = 8.304+764.5j ohms
(5) R opt    = 8.3 ohms
(6) L opt    = 0.304 H

    Power transfer with optimal load
(3) Z total  = 16.12+3.26j ohms (seen by source)
    I source = 4.47-0.9043j A = (4.561 @ -11.44 deg A)
(4) S source = 335.3+67.82j VA = (342.1 @ 11.44 deg VA)
    I load   = 4.47-0.4521j A = (4.493 @ -5.775 deg A)
    V load   = 382.8+3414j V = (3435 @ 83.6 deg V)
(2) S load   = 167.6+1.543e+04j VA = (1.543e+04 @ 89.38 deg VA)
    P load   =  50% of P source

    Constrained load -- match R as close as possible
    Z load   =   10+25.13j ohms
    I load   = 0.01263+0.1001j A = (0.1009 @ 82.81 deg A)
    V load   = -2.389+1.319j V = (2.729 @ 151.1 deg V)
    S load   = 0.1018+0.2559j VA = (0.2754 @ 68.3 deg VA)
(8) P load   = 0.0607% of potential load power

    Constrained load -- match -Xth as close as possible
(7) R ideal  = 739 ohm
    Z load   =  680+25.13j ohms
    I load   = 0.05552+0.04873j A = (0.07387 @ 41.27 deg A)
    V load   = 36.53+34.53j V = (50.27 @ 43.39 deg V)
    S load   = 3.711+0.1371j VA = (3.713 @ 2.117 deg VA)
(9) P load   = 2.21% of potential load power
```

```
#================================================================
# USERID:......... wbahn
# PROGRAMMER:..... Bahn, William L.
# COURSE:......... EENG-382
# TERM:........... SP14
# PROJECT:........ N/A
# FILENAME:....... HW02.py
# PYTHON VERSION:. 3.3.3
#================================================================

from z import *

Vs = 75     # 75 Vrms
R1 = 820    # 820 ohms
L1 = 0.033  # 33mH
C1 = 470e-9 # 470nF
freq = 400  # 400 Hz

# Compute phasors and impedances
VS  = ZR(Vs)
ZR1 = ZR(R1)
ZL1 = ZL(L1, freq)
ZC1 = ZC(C1, freq)

# Compute Thevenin equivalent circuit ①
print()
print("Thevenin Equivalent Circuit")
Vth = VS*ZR1/(ZL1+ZR1)
Zth = s(ZC1, p(ZR1,ZL1))
print("Vth = %s V = (%s V)" % (showR(Vth), showP(Vth)))
print("Zth = %s ohms" % showR(Zth))

# Optimal Load ⑤⑥
print()
print("Optimal load for max power transfer")
Zload = Zstar(Zth)
print("Z load   = %s ohms" % showR(Zload))
print("R opt    = %3.3g ohms" % R(Zload))
if (X(Zload)<0):
    print("C opt    = %3.3g F" % C(X(Zload), freq))
else:
    print("L opt    = %3.3g H" % L(X(Zload), freq))

# Compute total impedance seen by source ③
print()
print("Power transfer with optimal load")
Zeq = s(ZL1, p(ZR1, s(ZC1, Zload)))
print("Z total  = %s ohms (seen by source)" % showR(Zeq))

# Compute source parameters ④
IS = VS/Zeq
print("I source = %s A = (%s A)" % (showR(IS), showP(IS)))

SS = VS*Zstar(IS)
```

```
print("S source = %s VA = (%s VA)" % (showR(SS), showP(SS)))

# Compute load parameters (2)
IL = Vth/(Zth+Zload)
print("I load   = %s A = (%s A)" % (showR(IL), showP(IL)))

VL = IL*Zload
print("V load   = %s V = (%s V)" % (showR(VL), showP(VL)))

SL = VL*Zstar(IL)
print("S load   = %s VA = (%s VA)" % (showR(SL), showP(SL)))

# Compute fraction of source power delivered to load
Psource = P(SS)
Pmax = P(SL)
print("P load   = %3.3g%% of P source" % (100*(Pmax/Psource)))

# Constrained component values attempt #1 - match R
print()
print("Constrained load -- match R as close as possible")
R_s = 10    # 10 ohm (smallest available R)
L_s = 0.01  # 10 mH  (largest available L)

Zload = ZR(R_s)+ZL(L_s, freq)
print("Z load   = %s ohms" % showR(Zload))

# Compute load parameters (8)
IL = Vth/(Zth+Zload)
print("I load   = %s A = (%s A)" % (showR(IL), showP(IL)))

VL = IL*Zload
print("V load   = %s V = (%s V)" % (showR(VL), showP(VL)))

SL = VL*Zstar(IL)
print("S load   = %s VA = (%s VA)" % (showR(SL), showP(SL)))

# Compute fraction of source power delivered to load
Pload = P(SL)
print("P load   = %3.3g%% of potential load power" % (100*(Pload/Pmax)))

# Constrained component values attempt #1 - match -X
print()
print("Constrained load -- match -Xth as close as possible")
L_s = 0.01  # 10 mH  (largest available L)

# Calculate ideal R load (7)
R_s = mag(Zth+ZL(L_s, freq))
print("R ideal  = %3.3g ohm" % R_s)

R_s = 680   # 680 ohm (nearest value to 739 ohm ideal)

Zload = ZR(R_s)+ZL(L_s, freq)
print("Z load   = %s ohms" % showR(Zload))
```

```
# Compute load parameters (9)
IL = Vth/(Zth+Zload)
print("I load   = %s A = (%s A)" % (showR(IL), showP(IL)))

VL = IL*Zload
print("V load   = %s V = (%s V)" % (showR(VL), showP(VL)))

SL = VL*Zstar(IL)
print("S load   = %s VA = (%s VA)" % (showR(SL), showP(SL)))

# Compute fraction of source power delivered to load
Pload = P(SL)
print("P load   = %3.3g%% of potential load power" % (100*(Pload/Pmax)))
```

```
#==============================================================
# USERID:......... wbahn
# PROGRAMMER:..... Bahn, William L.
# COURSE:......... EENG-382
# TERM:........... SP14
# PROJECT:........ N/A
# FILENAME:....... z.py
# VERSION:........ 1.1
# PYTHON VERSION:. 3.3.3
#==============================================================

# This file contains simple functions that are useful for
# "scratchpad" Phasor circuit analysis computations.

import cmath

pi = cmath.pi

# Convert R, L, C into complex impedances
def ZR(R):      return complex(R,0)
def ZC(C, fHz): return complex(0,-1/(2*pi*fHz*C))
def ZL(L, fHz): return complex(0, 2*pi*fHz*L)
def ZZ(R, X):   return complex(R,X)

# Convert reactance into L or C
def L(X, fHz):  return X/(2*pi*fHz)
def C(X, fHz):  return -1/(2*pi*fHz*X)

# Combine impedances in series and parallel
def s(Z1,Z2):   return Z1 + Z2
def p(Z1,Z2):   return (Z1*Z2)/(Z1+Z2)

# Complex conjugate
def Zstar(Z):   return complex(Re(Z), -Im(Z))

# Get real/imaginary components (plus aliases)
def Re(z):      return z.real
def Im(z):      return z.imag
def P(S):       return Re(S)
def Q(S):       return Im(S)
def R(Z):       return Re(Z)
def X(Z):       return Im(Z)

# Polar components
def mag(Z):     return abs(Z)
def arg(Z):     return cmath.phase(Z)
def d2r(deg):   return deg*(pi/180)
def r2d(rad):   return rad*(180/pi)

# Generate strings for printing
def showR(Z):   return ("%4.4g%+4.4gj" % (Re(Z), Im(Z)))
def showP(Z):   return ("%4.4g @ %4.4g deg" % (mag(Z), r2d(arg(Z))))
def show(Z):    return ("%s = %s" % (showR(Z), showP(Z)))
```