Answers to selected questions in Assignment 1

Question 2(b):

Discuss the various ways of improving power factor in electrical power systems.

Solution:

Improving the power factor enhances the efficiency of the power system and reduces overall energy consumption. There are several methods for improving the power factor of inductive loads, including:

- Capacitor Compensation: Using capacitors to offset the inductive effects of the load.
- Synchronous Condensers: Employing synchronous motors that can adjust their power factor by varying the excitation.
- Phase Advancers: Used primarily for induction motors to improve the power factor.

In our solution, we will discuss only the capacitor method for power factor improvement. The power factor signifies the phase difference between voltage and current in an AC circuit. For an inductive LR load, the current lags the voltage, resulting in a low power factor. By introducing capacitors into the circuit, we can improve the power factor in two ways:

- Connecting a capacitor in parallel with the load introduces a leading current that can compensate for the lagging current caused by the inductive load. This reduces the phase difference between voltage and current, thereby improving the power factor.
- Connecting a capacitor in series with the load results in the capacitive reactance $X_C = -1/(C\omega)$ compensating for the inductive reactance $X_L = L\omega$ of the load. This also reduces the overall reactance, improving the power factor.

For an impedance load $Z = R + iX$ (R and X stand for the real and imaginary parts respectively), connected to a voltage source $V(t) = V_0 \exp(i\omega t)$, where V_0 is a real amplitude, the current flowing through the load is given by $I(t) = V(t)/Z$:

$$I(t) = \frac{V_0 \exp(i\omega t)}{R+iX} = \frac{V_0 \exp(i\omega t)(R-iX)}{R^2+X^2} = \frac{V_0 \exp(i\omega t)(R-iX)\sqrt{R^2+X^2}}{(R^2+X^2)\sqrt{R^2+X^2}} =$$
$$= \frac{V_0 \exp(i\omega t)}{\sqrt{R^2+X^2}}\left(\frac{R}{\sqrt{R^2+X^2}} - i\frac{X}{\sqrt{R^2+X^2}}\right) = \frac{V_0 \exp(i\omega t)\times exp(i\varphi)}{\sqrt{R^2+X^2}} = \frac{V_0 \exp(i(\omega t+\varphi))}{\sqrt{R^2+X^2}} \qquad (1)$$

where

$$\boxed{\varphi = \tan^{-1}\left(-\frac{X}{R}\right) \qquad (2)}$$

is the current phase with respect to the voltage as the reference signal (zero phase).

The instantaneous power delivered to the load is calculated as follows:

$$P(t) = \mathrm{Re}[V(t)] \times \mathrm{Re}[I(t)] = \frac{V_0^2}{\sqrt{R^2+X^2}} \cos(\omega t)\cos(\omega t + \varphi) \quad (3)$$

where Re[…] signifies the real part. The power averaged over the period is:

$$\begin{aligned} \bar{P} &= \frac{1}{T}\int_0^T P(t)dt = \frac{V_0^2}{\sqrt{R^2+X^2}} \times \frac{1}{T}\int_0^T \cos\left(\frac{2\pi}{T}t\right)\cos\left(\frac{2\pi}{T}t + \varphi\right)dt = \\ &= \frac{V_0^2}{\sqrt{R^2+X^2}} \times \frac{1}{T}\int_0^T \cos\left(\frac{2\pi}{T}t\right)\left(\cos\left(\frac{2\pi}{T}t\right)\cos(\varphi) - \sin\left(\frac{2\pi}{T}t\right)\sin(\varphi)\right)dt = \\ &= \frac{V_0^2\cos(\varphi)}{\sqrt{R^2+X^2}} \times \frac{1}{T}\int_0^T \cos^2\left(\frac{2\pi}{T}t\right)dt - \frac{V_0^2\sin(\varphi)}{\sqrt{R^2+X^2}}\frac{1}{T}\int_0^T \cos\left(\frac{2\pi}{T}t\right)\sin\left(\frac{2\pi}{T}t\right)dt = \\ &= \frac{V_0^2\cos(\varphi)}{2\sqrt{R^2+X^2}} \end{aligned} \quad (4)$$

where

$$\frac{1}{T}\int_0^T \cos^2\left(\frac{2\pi}{T}t\right)dt = \frac{1}{2}$$

$$\frac{1}{T}\int_0^T \cos\left(\frac{2\pi}{T}t\right)\sin\left(\frac{2\pi}{T}t\right)dt \equiv 0$$

Finally, we derive the expression for the average power:

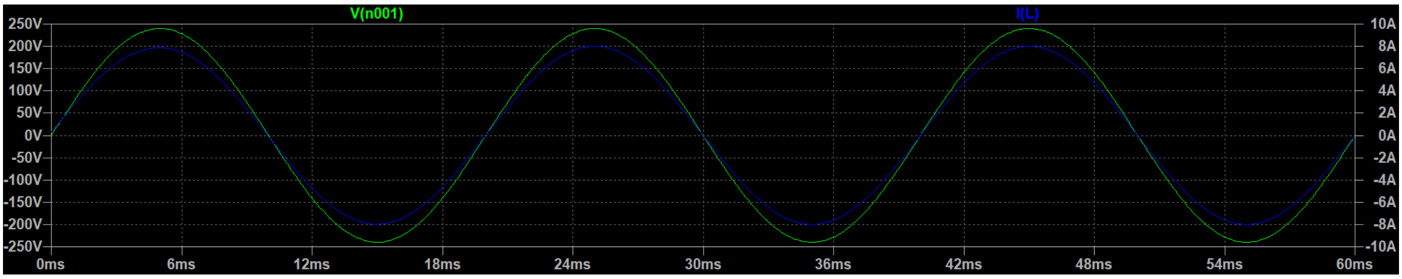
$$\bar{P} = \frac{V_0^2\cos(\varphi)}{2\sqrt{R^2+X^2}} = V_{rms}I_{rms}\cos(\varphi) \quad (5)$$

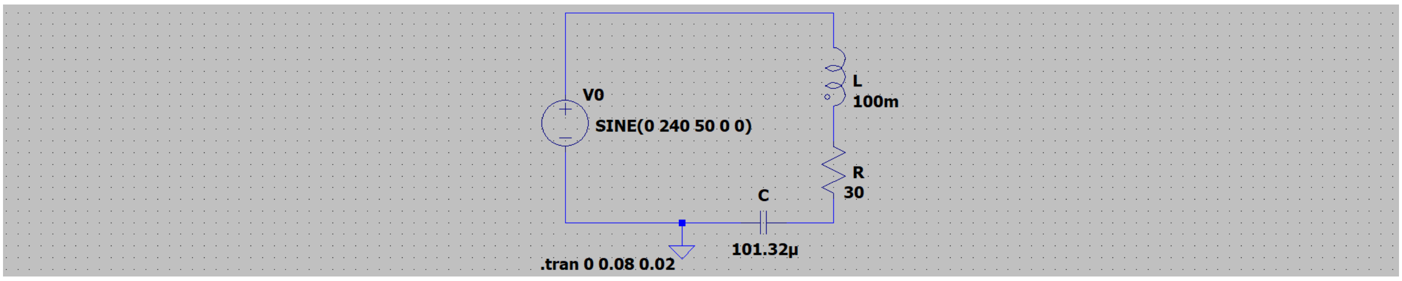
where $V_{rms} = \frac{V_0}{\sqrt{2}}$ and $I_{rms} = \frac{V_0}{\sqrt{2}\times\sqrt{R^2+X^2}} = \frac{V_{rms}}{\sqrt{R^2+X^2}}$. RMS amplitudes can be understood as constant voltage and current values that offer equivalent performance to an AC voltage source, given that the voltage and current are in phase. **The coefficient $\cos(\varphi)$ is known as the power factor. We aim to increase it.**

When the compensation capacitor is connected in series with the RL load, the total impedance and the current phase become $Z = R + iX = R + i\left(L\omega - \frac{1}{C\omega}\right)$ and $\varphi = \tan^{-1}\left(-\frac{L\omega - \frac{1}{C\omega}}{R}\right) = \tan^{-1}\left(-\frac{LC\omega^2 - 1}{C\omega R}\right)$ respectively. The phase becomes zero, resulting in $\cos(\varphi) = 1$, for the specific value of the compensation capacitor:

$$C = \frac{1}{\omega^2 L} = \frac{1}{(2\pi f)^2 L} \quad (6)$$

The LTspice circuit simulation for $f = 50$ Hz ($\omega = 2\pi f$), $L = 100$ mH, $R = 30\ \Omega$, and $C = 101.32$ μF (calculated using Eq. (6)) is displayed below. Perfect phase alignment between the voltage and current sinusoids is observed. During simulation, click on a wire to display the voltage relative to the ground (reference point), and click on any element to view the current passing through it.





When the compensation capacitor is connected in parallel with the RL load, the total impedance becomes:

$$Z = \frac{\left(-\frac{i}{C\omega}\right)(R+i\omega L)}{R+i\omega L-\frac{i}{C\omega}} = \frac{\omega L - iR}{CR\omega + i(LC\omega^2-1)} = \frac{(\omega L - iR)(CR\omega - i(LC\omega^2-1))}{(CR\omega)^2+(LC\omega^2-1)^2} = \\ = \frac{R}{(CR\omega)^2+(LC\omega^2-1)^2} - i\,\frac{CR^2\omega - \omega L(1-LC\omega^2)}{(CR\omega)^2+(LC\omega^2-1)^2} \quad (7)$$

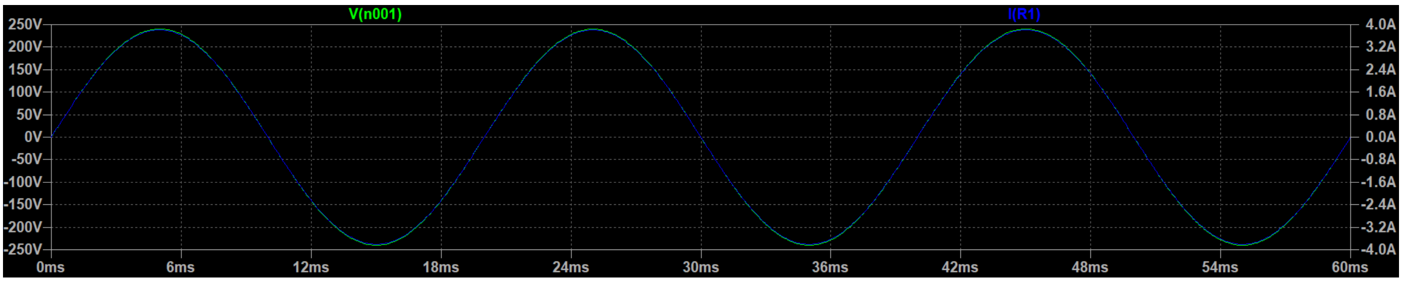
Here, $\frac{R}{(CR\omega)^2+(LC\omega^2-1)^2}$ is the real part of the impedance and $X = -\frac{CR^2\omega - \omega L(1-LC\omega^2)}{(CR\omega)^2+(LC\omega^2-1)^2}$ is the imaginary part. Hence, based on Eq. (2), we derive the phase as follows:

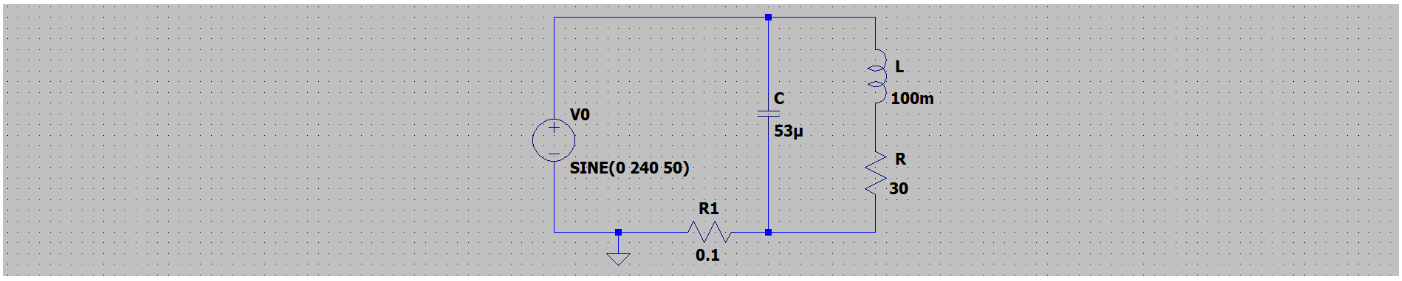
$$\varphi = \tan^{-1}\left(\frac{CR^2\omega - \omega L(1-LC\omega^2)}{R}\right) \quad (8)$$

The phase becomes zero if $CR^2\omega - \omega L(1-LC\omega^2) = 0$. From here, we derive the equation for the compensation capacitance:

$$C = \frac{L}{R^2+L^2\omega^2} \quad (9)$$

The LTspice circuit simulation for $f = 50$ Hz, $L = 100$ mH, $R = 30\ \Omega$, and $C = 53$ μF (calculated using Eq. (9)) is displayed below. Perfect phase alignment between the voltage and current sinusoids is observed. During simulation, click on a wire to display the voltage relative to the ground (reference point). To measure the current outside the LCR network, we introduced a small “sensor resistor” $R_1 = 0.1\ \Omega$. This addition minimally impacts the current conditions but facilitates the probe's ability to measure the current by clicking on it. This method of current measurement is commonly employed in practical circuits. The sensor resistor typically possesses a very low value and high tolerance. By measuring the voltage across the resistor and dividing it by the resistance, one can calculate the current.





[Python code for the LR circuit]:

```
#
# Calculating all parameters of the LR circuit, including impedance, total current, power and power factor,
# compensation capacitor for its series or parallel connection.
#
# Dr. Dmitriy Makhnovskiy, City College Plymouth, England, 09.04.2024
#

import math

# Given values
R = 80  # Resistance (in ohms)
L = 8e-2  # Inductance (in henries)
f = 100  # Frequency (in Hz)
omega = 200 * math.pi  # Angular frequency (in rad/s)

# Calculate impedance
Z = complex(R, omega * L)

# Calculate impedance magnitude
Z_magnitude = abs(Z)

# Calculate current in the circuit
V = 220  # Voltage (in volts)
I = V / Z_magnitude

# Calculate RMS current
I_rms = I / math.sqrt(2)

# Calculate phase angle (in radians)
phi_rad = math.atan(-omega * L / R)

# Convert phase angle to degrees
phi_deg = math.degrees(phi_rad)

# Calculate power factor
power_factor = math.cos(phi_rad)

# Calculate power consumed by the circuit
P = I_rms ** 2 * R

# Calculate the compensation capacitance for the series connection
C1 = 1 / (omega ** 2 * L)
C1 = C1 / 1.0e-6  # uF

# Calculate the compensation capacitance for the parallel connection
C2 = L / (R ** 2 + L ** 2 * omega ** 2)
C2 = C2 / 1.0e-6  # uF

# Print results
print('')
print("Circuit Impedance (Z): {:.3f} + {:.3f}j ohms".format(Z.real, Z.imag))
print("|Z| (Impedance Magnitude): {:.3f} ohms".format(Z_magnitude))
print("Current in the circuit (I): {:.3f} A".format(I))
print("RMS Current (I_rms): {:.3f} A".format(I_rms))
print("Phase Angle (ϕ): {:.2f} degrees".format(phi_deg))
print("Power Factor (cosϕ): {:.2f}".format(power_factor))
print("Power Consumed by the Circuit (P): {:.2f} W".format(P))
print("Compensation Capacitance for the series connection: {:.3f} uF".format(C1))
print("Compensation Capacitance for the parallel connection: {:.3f} uF".format(C2))
```

Question 2(c):

A single-phase AC generator supplies the following loads:

(i) Lighting load of 30 kW at unity power factor.

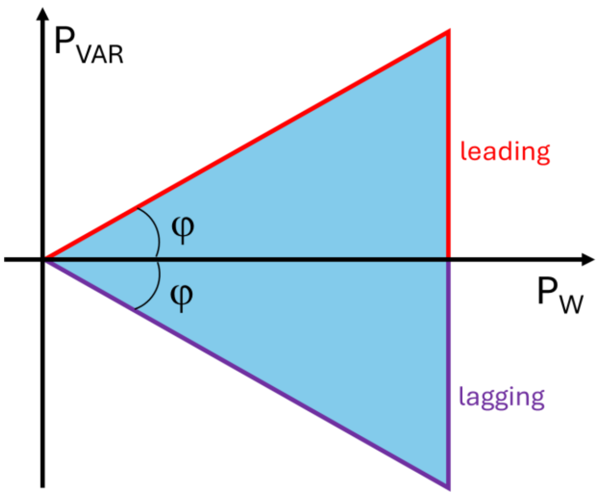
(ii) Induction motor load of 150 kW at $p.f. = 707$ lagging.

(iii) Synchronous motor load of 60 kW at $p.f. = 0.9$ leading.

Calculate the total kW and kVA delivered by the generator and the power factor at which it works.

Solution:

In a circuit containing reactive elements, all quantities become vectors due to the phase differences between them, including power, as illustrated in the power triangle below. We can use either voltage or current as a reference signal and measure the phases of other quantities relative to this reference. Only true power, measured in Watts (W), is actually dissipated and contributes to real work. Reactive and apparent (total) power, although also measured in units of Volt-Amperes (VA) and Volt-Amperes-Reactive (VAR), which coincide with Watts in terms of dimension, do not contribute to real work. Reactive power oscillates between the source and the load, essentially being useless for performing any actual work. Therefore, it is advisable to minimize reactive power.



Here:

$p.f. = \cos(\varphi)$ is the power factor

P_W is the true power (W)

P_{VAR} is the reactive power (VAR)

$P_{VA} = \frac{P_W}{\cos(\varphi)}$ is the apparent power (VA)

$\vec{P}_{VA} = (P_W, P_{VAR})$ is the apparent power **vector**

$|\vec{P}_{VA}| = P_{VA} = \sqrt{P_W^2 + P_{VAR}^2}$ is the apparent power **magnitude**

$P_{VAR} = \pm\sqrt{P_{VA}^2 - P_W^2} = \pm\sqrt{\frac{P_W^2}{p.f.^2} - P_W^2} = \pm P_W \times \frac{\sqrt{1-p.f.^2}}{p.f.}$ (leading "+" or lagging "–")

For the three loads (i, ii, ii) indicated in the problem statement, we obtain:

$$\vec{P}_{VA} = \vec{P}^{\text{i}}_{VA} + \vec{P}^{\text{ii}}_{VA} + \vec{P}^{\text{iii}}_{VA} = \left(P^{\text{i}}_{W}, P^{\text{i}}_{VAR}\right) + \left(P^{\text{ii}}_{W}, P^{\text{ii}}_{VAR}\right) + \left(P^{\text{iii}}_{W}, P^{\text{iii}}_{VAR}\right)$$

$$\vec{P}_{VA} = \left(P^{\text{i}}_{W} + P^{\text{ii}}_{W} + P^{\text{iii}}_{W}, P^{\text{i}}_{VAR} + P^{\text{ii}}_{VAR} + P^{\text{iii}}_{VAR}\right)$$

where

$\vec{P}_{VA}$ is the apparent power **vector** of the whole circuit (VA)

$P_W = P^{\text{i}}_{W} + P^{\text{ii}}_{W} + P^{\text{iii}}_{W}$ is the true power of the whole circuit (W)

$P_{VAR} = P^{\text{i}}_{VAR} + P^{\text{ii}}_{VAR} + P^{\text{iii}}_{VAR}$ is the reactive power of the whole circuit (VAR)

$P_{VA} = \sqrt{\left(P^{\text{i}}_{W} + P^{\text{ii}}_{W} + P^{\text{iii}}_{W}\right)^2 + \left(P^{\text{i}}_{VAR} + P^{\text{ii}}_{VAR} + P^{\text{iii}}_{VAR}\right)^2}$ is the apparent power of the whole circuit

$\varphi_0 = \text{atan2}(P_{VAR}, P_W)$ is the phase of the total current in the circuit calculated using the atan2 function

$p.f. = \cos(\varphi_0)$ is the power factor of the whole circuit

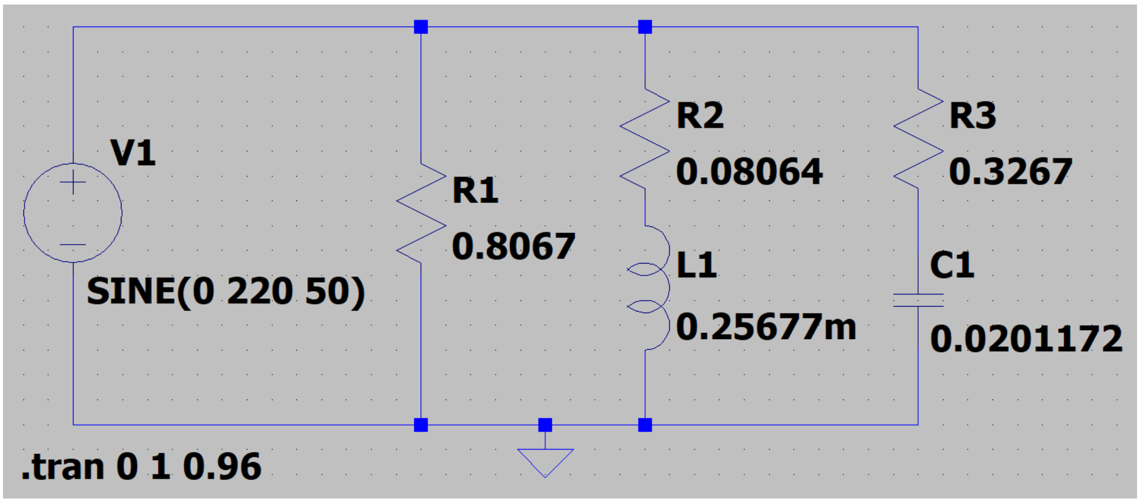
Numerical calculations obtained using the Python code shown at the end of the section:

```
PVA1 = (30.0, 0.0)
PVA2 = (150.0, -150.045)
PVA3 = (60.0, 29.059)

True power of the whole circuit =  240.0  kW
Reactive power of the whole circuit =  -120.986  kVAR
Apparent power of the whole circuit =  268.771  kVA

Power factor of the whole circuit =  0.893  (lagging)
Phase (rad) of the total current in the circuit =  -0.467
Phase (deg) of the total current in the circuit =  -26.753
```

The circuit with different power dissipations and power factors can be simulated in LTspice, but a voltage source must be specified, for example, $V(t) = 220\sin(\omega t)$ with $\omega = 100\pi$. In this case, $V_{rms} = 220/\sqrt{2} \approx 155.563$ V. A purely resistive load is simulated by a resistor R, a lagging load by an LR circuit, and a leading load by a CR circuit, as illustrated in the figure below.



For a purely resistive load, we have:

$$R = \frac{V_{rms}^2}{P_W}$$

For the inductive (LR) and capacitive (CR) loads, we have:

$$I_{rms} = \frac{P_W}{p.f.\times V_{rms}} \text{ is the rms current through the load}$$

$$R = \frac{P_W}{I_{rms}^2} \text{ is the load resistance}$$

$$Z = \sqrt{R^2 + X^2} = \frac{V_{rms}}{I_{rms}} \text{ is the load impedance}$$

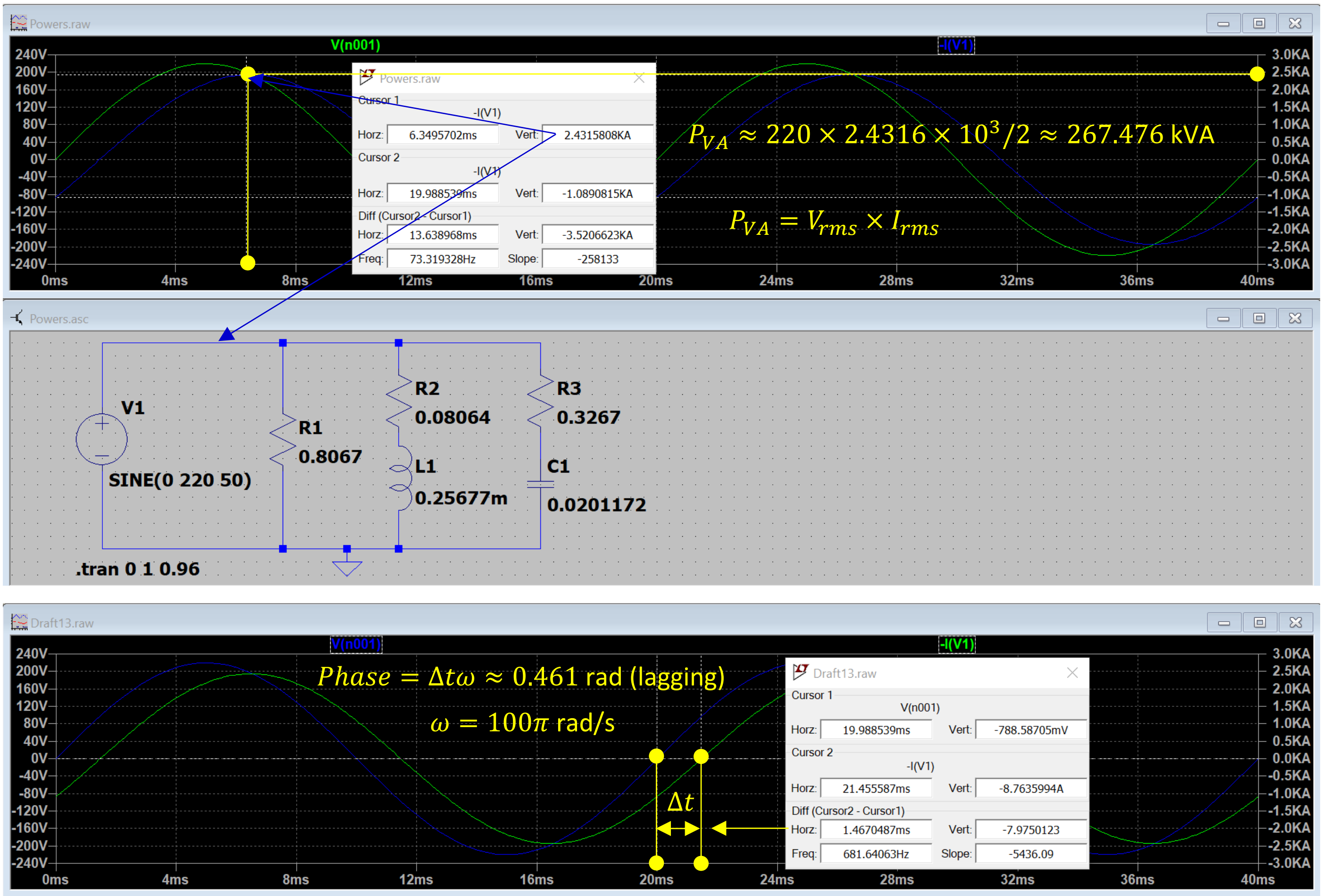
$$X = \pm\sqrt{\frac{V_{rms}^2}{I_{rms}^2} - R^2} \text{ is the load reactance}$$

$$L = \frac{|X|}{\omega} \text{ is the load inductance (if lagging) or } C = \frac{1}{\omega|X|} \text{ is the load capacitance (if leading)}$$

For the three loads indicated in the problem statement, we obtain the values used in the circuit above:

```
Resistance_1 = 0.8067 Ohms
Resistance_2 = 0.0806 Ohms
Resistance_3 = 0.3267 Ohms
Component_1 = no reactive part
Component_2 = 0.00025677 H
Component_3 = 0.0201172 F
```

The apparent power and phase obtained from the LTspice simulation graphs are close to those theoretically calculated above. Here it is necessary to account for the inaccuracies when reading data using cursors.



Python code: in lines (9,10), you can enter any number of P_W and the accompanying power factors.

```
1. # Calculating the vector sum of powers and the power factor
2.
3. import math  # math library
4.
5. sign = lambda num: 1 if num > 0 else -1 if num < 0 else 0  # sign function
6. pi = math.pi  # pi = 3.14159...
7.
8. # Loads' parameters
9. pf_array = [1.0, -0.707, 0.9]  # power factor array for the three loads ("+" for leading, and "-" for lagging)
10. PW_array = [30.0, 150.0, 60.0]  # true power array (kW) for the three loads
11.
12. # Parameters for LTspice simulations
13. V0 = 220.0  # voltage amplitude in V(t) = V0sin(w*t)
14. f = 50  # frequency in Hz
15.
16. # Calculations of the powers, phase, and power factor
17. ind = [i for i in range(len(pf_array))]  # indexes
18. PVA_array = [pw / abs(pf) for pw, pf in zip(PW_array, pf_array)]  # VA or kVA
19. PVAR_array = [sign(pf) * (pav**2 - pw**2)**0.5 for pav, pw, pf in zip(PVA_array, PW_array, pf_array)]  # VAR or kVAR
20. print('\n'.join([f'PVA{i + 1} = ({round(pw, 3)}, {round(pvar, 3)})' for pw, pvar, i in zip(PW_array, PVAR_array, ind)]))
21. PW = sum([pw for pw in PW_array])  # true power of the whole circuit
22. PVAR = sum([pvar for pvar in PVAR_array])  # reactive power of the whole circuit
23. PVA = (PW**2 + PVAR**2)**0.5  # apparent power of the whole circuit
24. phase_rad = math.atan2(PVAR, PW)  # phase (rad) of the total current in the circuit
25. phase_deg = phase_rad * 180.0 / pi  # phase (degree) of the total current in the circuit
26. PF = math.cos(phase_rad)  # power factor of the whole circuit
27. print('')
28. print('True power of the whole circuit = ', round(PW,3), ' kW')
29. print('Reactive power of the whole circuit = ', round(PVAR,3), ' kVAR')
30. print('Apparent power of the whole circuit = ', round(PVA,3), ' kVA')
31.
32. print('')
33. if sign(phase_rad) < -0.0005:
34.     print('Power factor of the whole circuit = ', round(PF,3), ' (lagging)')
35. elif sign(phase_rad) > 0.0005:
36.     print('Power factor of the whole circuit = ', round(PF, 3), ' (leading)')
37. else:
38.     print('Power factor of the whole circuit = ', 1, ' (perfect matching)')
39.
40. print('Phase (rad) of the total current in the circuit = ', round(phase_rad,3))
41. print('Phase (deg) of the total current in the circuit = ', round(phase_deg,3))
42.
43. # Calculation of the equivalent circuit parameters
44. print('')
45. print('Equivalent circuit parameters:')
46. PW_array = [x * 1.0e+3 for x in PW_array]  # transfer kW to W for calculations
47. Vrms = V0 / (2.0**0.5)  # voltage rms amplitude
48. w = 2.0 * pi * f
49. Irms_array = [pw / (abs(pf) * Vrms) for pw, pf in zip(PW_array, pf_array)]  # currents in the loads A
50. print('')
51. print('\n'.join([f'Current_{i + 1} = {round(Irms * 2.0**0.5, 3)} A' for Irms, i in zip(Irms_array, ind)]))
52.
53. print('')
54. R_array = [pw / (Irms**2) for pw, Irms in zip(PW_array, Irms_array)]  # resistances
55. print('\n'.join([f'Resistance_{i + 1} = {R} Ohms' for R, i in zip(R_array, ind)]))
56. X_array = [(Vrms**2 / (Irms**2) - R**2)**0.5 for Irms, R in zip(Irms_array, R_array)]  # reactances
57.
58. for i in range(len(pf_array)):
59.     if pf_array[i] < -0.0:
60.         L = X_array[i] / w
61.         print(f'Component_{i + 1} = {L} H')
62.     elif 0.0 < pf_array[i] < 1.0:
63.         C = 1.0 / (X_array[i] * w)
64.         print(f'Component_{i + 1} = {C} F')
65.     else:
66.         print(f'Component_{i + 1} = no reactive part')
```

Question 3:

A 2-wire DC distributor AB of length 500 m is fed at both ends with equal voltage of 240 V. Loads of 60 A, 70 A, 50 A, and 40 A are tapped at distance of 100 m, 250 m, 350 m, and 400 m from the end A respectively. If the cross-section area of each conductor is $A = 1$ cm^2 and the resistivity of the material of the conductor is $\rho = 1.8$ μΩcm:

a. Sketch a clearly labelled diagram of the transmission line.

b. Determine the total current supplied by each feeder.

c. Determine the point of minimum potential.

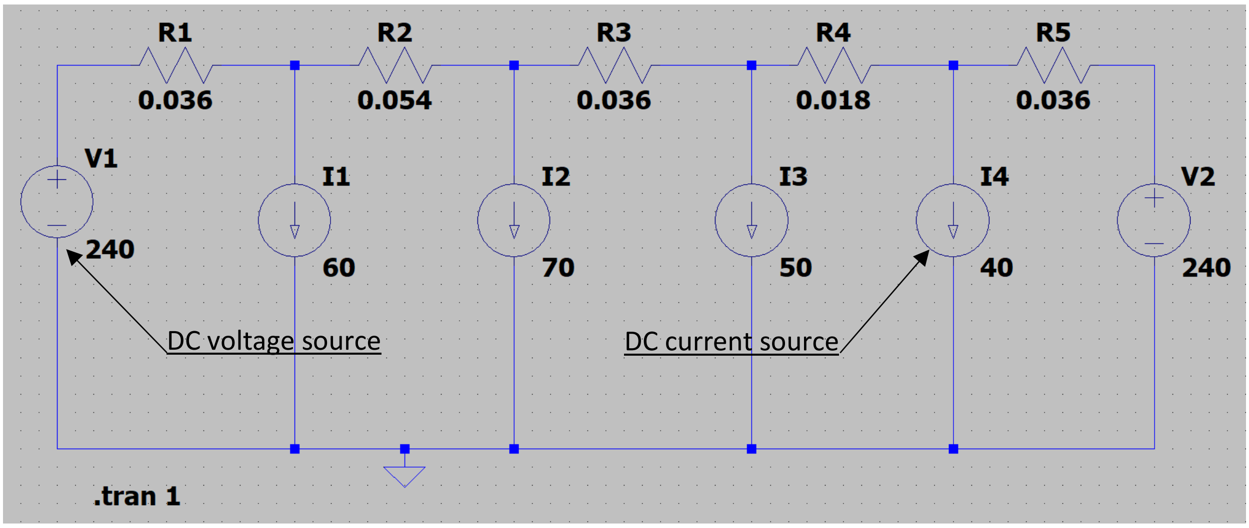
d. Calculate the voltages at the various load points.

e. Calculate the power dissipated in each section of the transmission line.

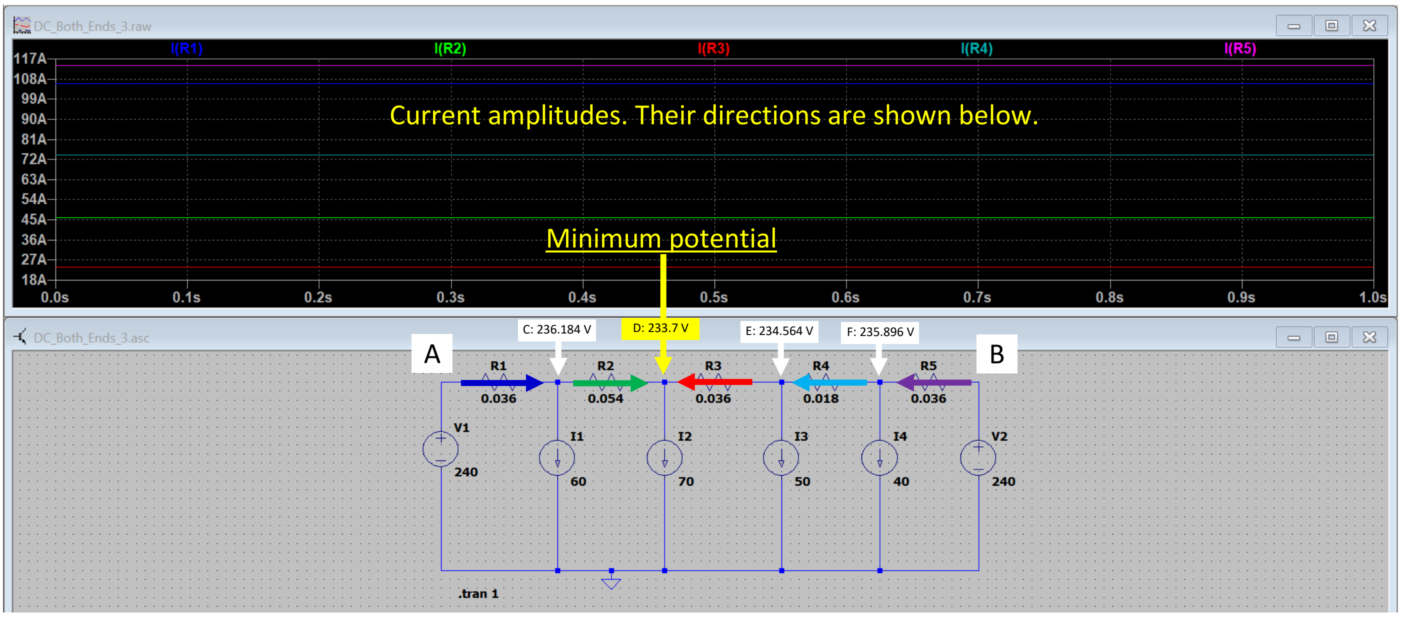
f. Determine the overall efficiency of the network.

Solution:

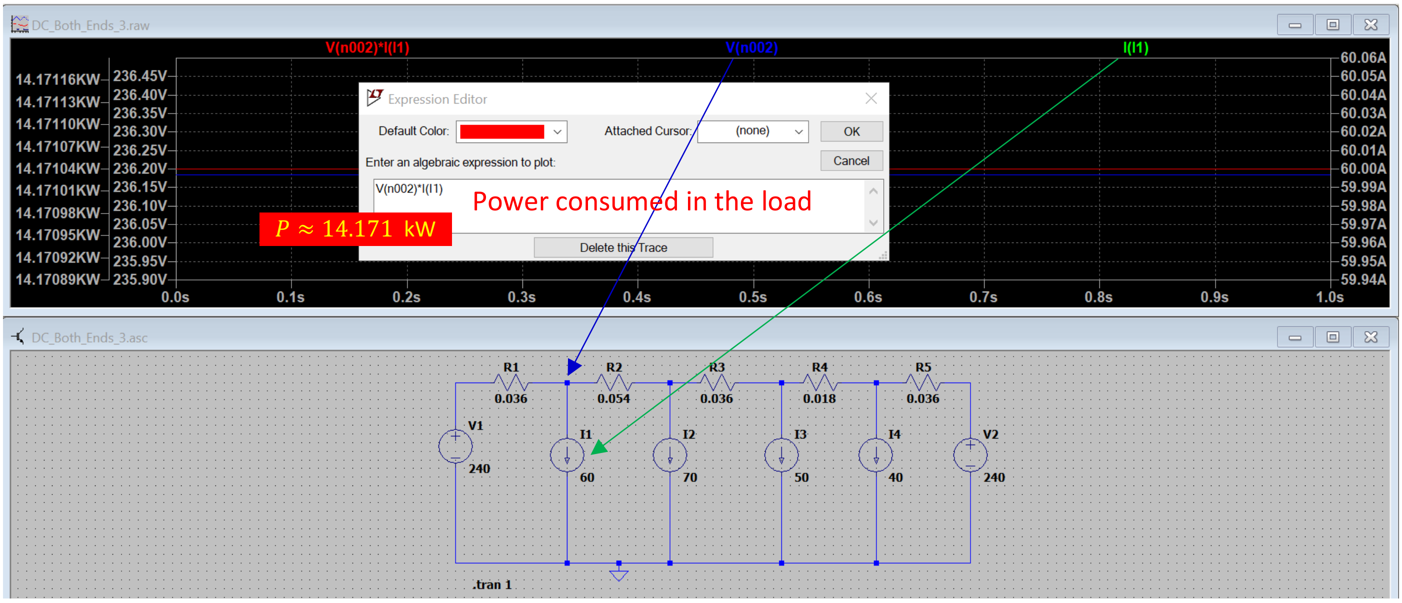
A DC distribution system can be easily modelled in LTspice using current (I) and voltage (V) sources as shown in the figure below.



The linear resistance R of a two-wire line is determined by the equation $R = \frac{\rho L}{A} \times 2 = 3.6 \times 10^{-4} \times L$, where $A = 10^{-4}$ m^2 is the wire cross-section, $\rho = 1.8 \times 10^{-8}$ Ωm is the resistivity, and L is the length measured in meters. For the line sections of 100 m, 150 m, 100 m, 50 m, and 100 m, we obtain respectively: $R_1 = 0.036$ Ω, $R_2 = 0.054$ Ω, $R_3 = 0.036$ Ω, $R_4 = 0.018$ Ω, $R_5 = 0.036$ Ω. The current amplitudes in the sections and their directions modelled in LTspice are shown in the figure below: $I(R_1) = 106$ A, $I(R_2) = 46$ A, $I(R_3) = 24$ A, $I(R_4) = 74$ A, and $I(R_5) = 114$ A. You can also use positive and negative values to indicate the current directions, for example assuming a clockwise direction to be positive. To see the direction of current through a component in LTspice, place the cursor over this component (without clicking on it), after which the simulator will indicate the direction of the current with a red arrow. However, if the current value turns out to be negative, then this direction should be reversed. This is how the current directions shown in the figure below were obtained.

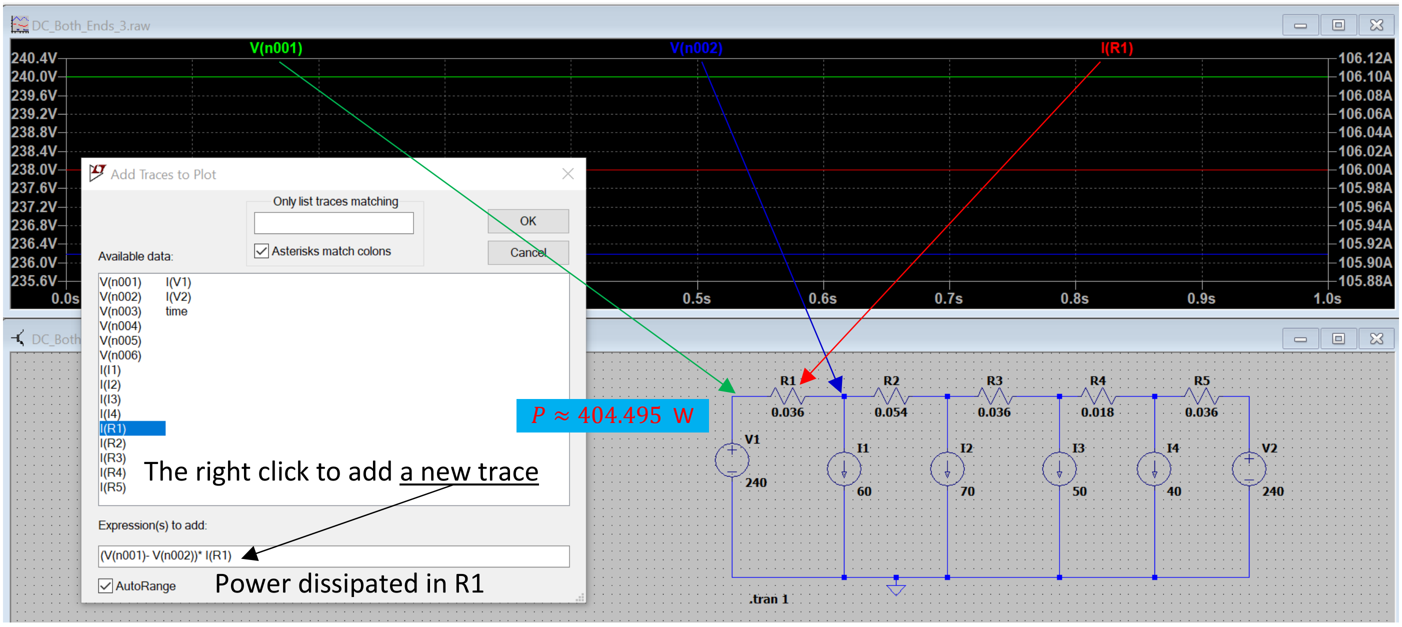


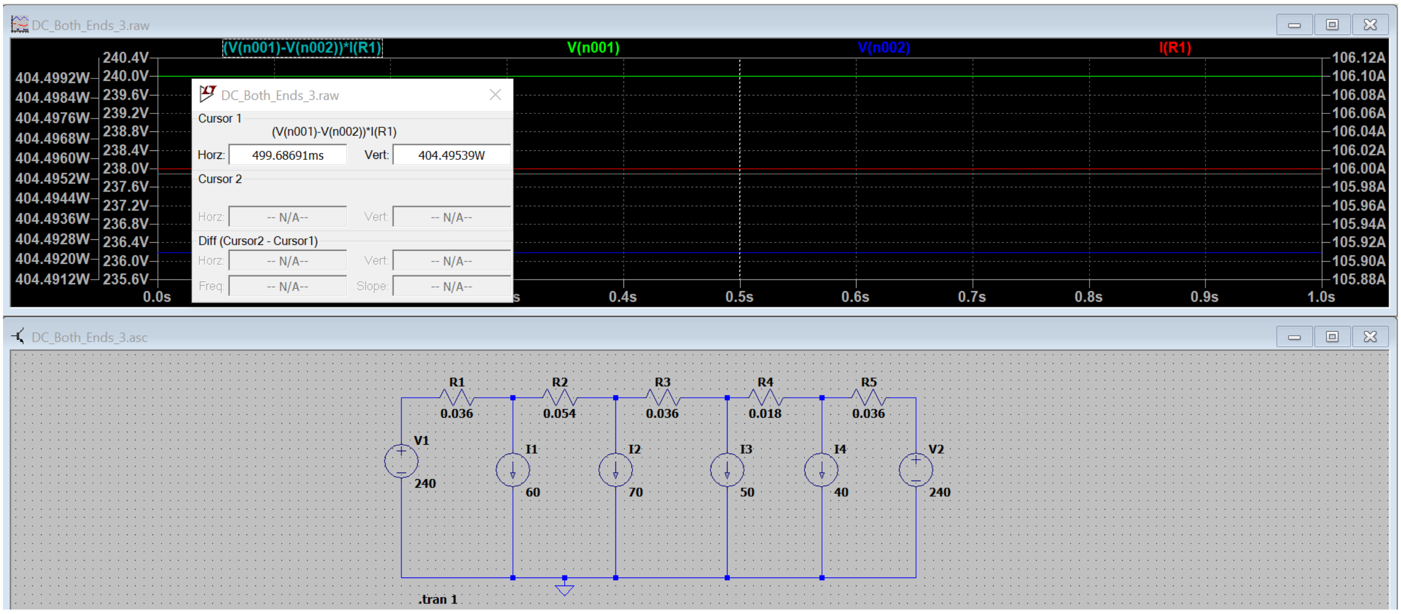
To show the potential at a point relative to the ground in LTspice, click on this point with the probe. We obtain the following potentials at load points: $V_A = V_B = 240$ V, $V_C = 236.184$ V, $V_D = 233.7$ V, $V_E = 234.564$ V, and $V_F = 235.896$ V. The minimum potential is achieved at D. Using the right click over the black simulation screen, we can add traces made up of basic parameters using simple mathematical operations. For example, we can multiply the current **I(I1)** in the first load by the voltage **V(n002)** in the corresponding node, and thereby obtain the power **P = V(n002)×I(I1)** consumed in this load, as shown in the figure below.



Power consumed in loads: $P_1 = 14.171$ kW, $P_2 = 16.359$ kW, $P_3 = 11.728$ kW, and $P_4 = 9.436$ kW. The total power consumed in the loads is $P_c = \sum_{i=1}^{4} P_i = 51.694$ kW. The total power supplied by two voltage sources $P_s = 240 \times (I(R_1) + I(R_5)) = 240 \times (106 + 114) = 52.8$ kW. The line efficiency $\eta = \frac{P_c}{P_s} \times 100\% \approx 98\%$.

Below we show how to determine the power dissipated in a section of line. It is calculated as the potential difference across the section multiplied by the current through it. The corresponding trace can be added using the right click on the black screen.





Power dissipated in the line sections: $P_{R1} = 404.495$ W, $P_{R2} = 114.264$ W, $P_{R3} = 20.736$ W, $P_{R4} = 98.568$ W, and $P_{R5} = 467.856$ W. The total power dissipated in the line sections is $P_d = \sum_{i=1}^{5} P_{Ri} = 1{,}105.919$ W. The total power supplied by two voltage sources $P_s = 52.8$ kW. The line efficiency $\eta = \frac{P_s - P_d}{P_s} \times 100\% \approx 98\%$.

In modern engineering, designing distribution systems solely with a calculator is unimaginable. Therefore, we aim to demonstrate how to prepare and run simulations using tools like LTspice and program algorithms using Python or other programming languages. Let us try automating the development process by using the fundamental design equations for a line fed from both ends:

1. $V_A - V_B = I(R_1) \times R_1 + \sum_{i=2}^{N+1} \left(I(R_1) - \sum_{k=1}^{i-1} I_k \right) \times R_i$ is the equation to find $I(R_1)$
2. $I(R_i) = I(R_1) - \sum_{k=1}^{i-1} I_k$ are the currents through the line sections for $2 \leq i \leq N + 1$
3. $V_1 = V_A - I(R_1) \times R_1$ is the voltage at the first load node
4. $V_j = V_A - I(R_1) \times R_1 - \sum_{i=2}^{j} \left(I(R_1) - \sum_{k=1}^{i-1} I_k \right) \times R_i$ are the voltages at the load nodes for $2 \leq j \leq N - 1$

where N is the number of loads. Note that in the first equation we have double summation over two indices (i, k). These equations can be programmed in Python for any number of loads (array):

```python
#
# DC transmission lines fed from both ends
#
# Dr. Dmitriy Makhnovskiy, City College Plymouth, England
# 02.06.2024
#

# Given values:
R = [0.036, 0.054, 0.036, 0.018, 0.036]  # Resistances
VA = 240  # Voltage at A
VB = 240  # Voltage at B
loads = [60, 70, 50, 40]  # Loads in Amperes at specified distances

def calculate_currents_and_voltages(R, VA, VB, loads):
    N = len(R) - 1
    I = [0] * (N + 1)
    V = [0] * N

    # Calculate I(R1)
    numerator = VA - VB
    sum_term = 0
    for i in range(1, N + 1):
        sum_term += R[i] * (1 - sum([1 for k in range(i)]))
    I[0] = numerator / (R[0] + sum_term)
    I[0] = 106  # As calculated analytically

    # Calculate the currents I(Ri) for i >= 2
    for i in range(1, N + 1):
        I[i] = I[0] - sum(loads[:i])

    # Calculate the voltages at each node
    V[0] = VA - I[0] * R[0]
    for j in range(1, N):
        V[j] = V[0] - sum([(I[0] - sum(loads[:i])) * R[i] for i in range(1, j + 1)])

    return I, V

# Calculations
currents, voltages = calculate_currents_and_voltages(R, VA, VB, loads)
print("Currents: ", currents)
print("Voltages: ", voltages)

```

Numerical solutions:

```
Currents:  [106, 46, -24, -74, -114]
Voltages:  [236.184, 233.7, 234.564, 235.896]
```

References for learning Python:

- Download Python (free): https://www.python.org/downloads/
- Python Tutorial: https://www.w3schools.com/python/
- Python IDE (Community, free): https://www.jetbrains.com/pycharm/download/?section=windows