DC distribution lines

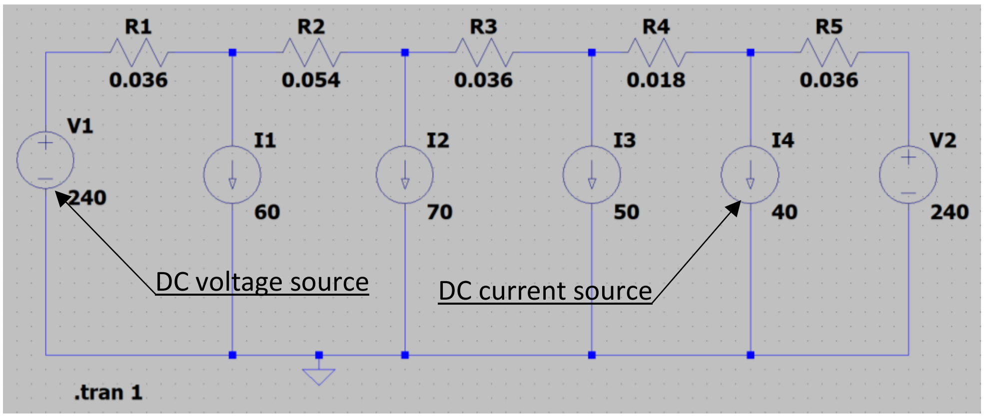
Task:

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:

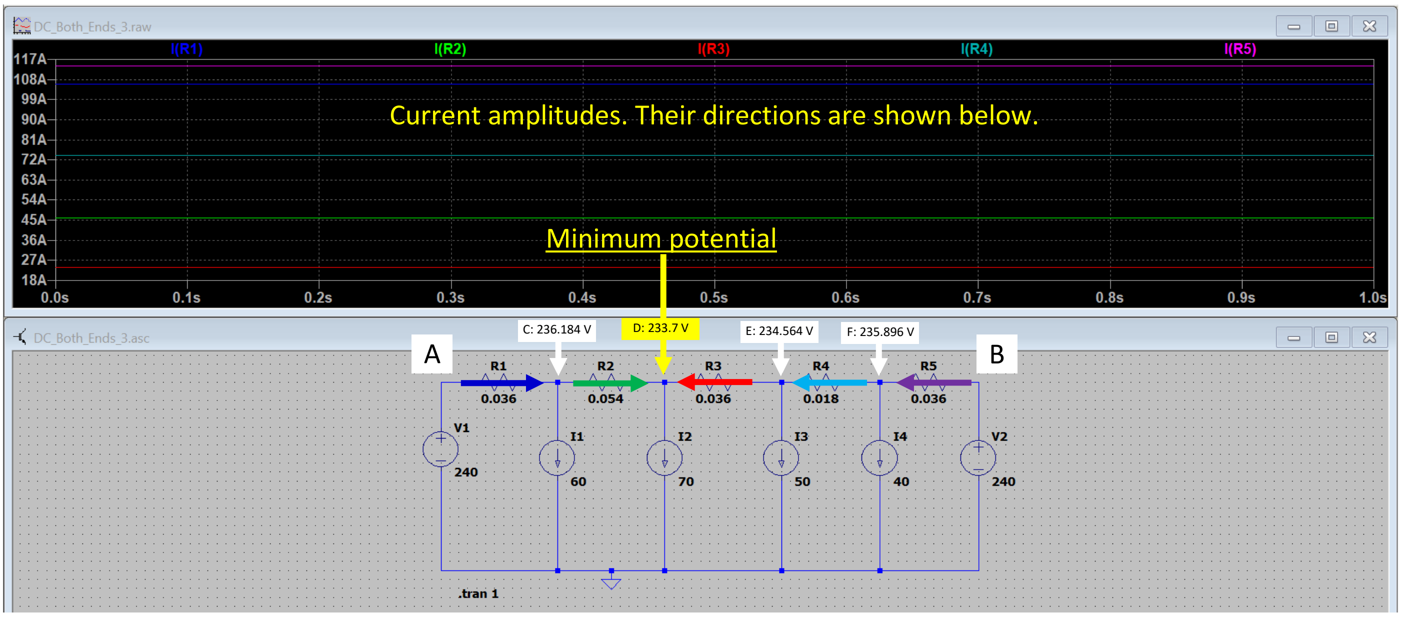
a. Sketch a clearly labelled diagram of the transmission line.
b. Determine the total current supplied by each feeder.
c. Determine the point of minimum potential.
d. Calculate the voltages at the various load points.
e. Calculate the power dissipated in each section of the transmission line.
f. Determine the overall efficiency of the network.

Solution:

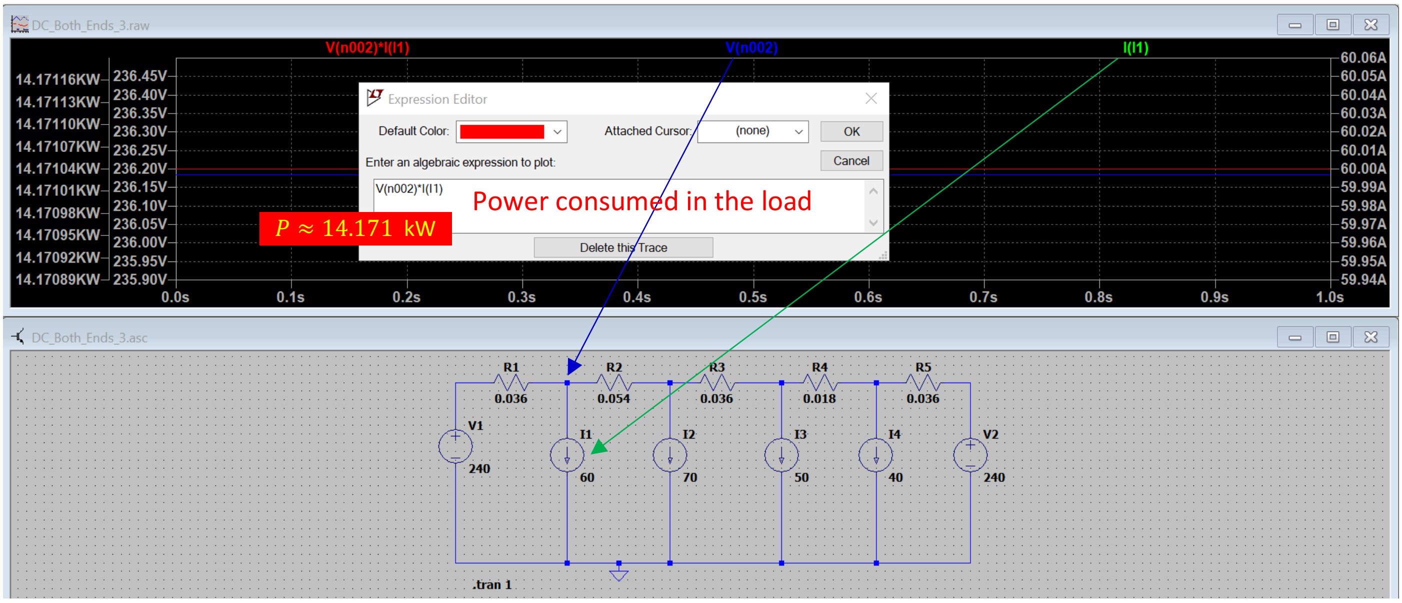
A DC distribution system can be easily modelled in LTspice using current (I) and voltage (V) sources as shown in the figure below.



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.

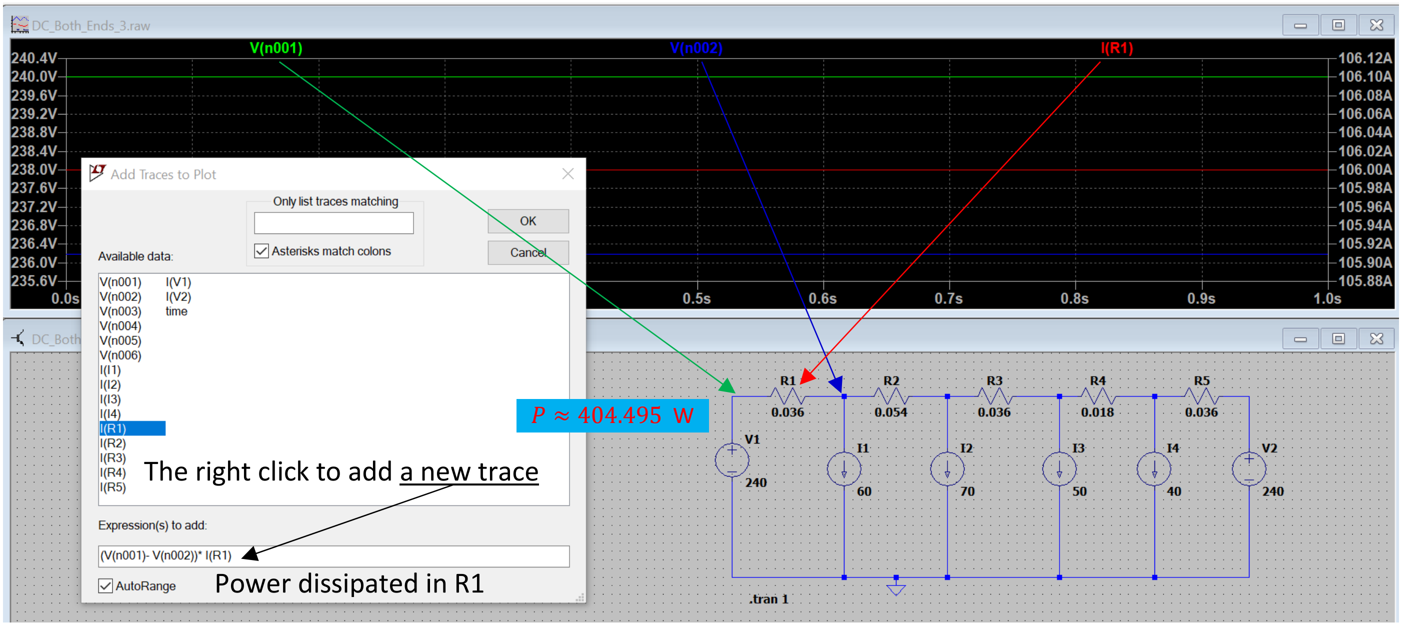


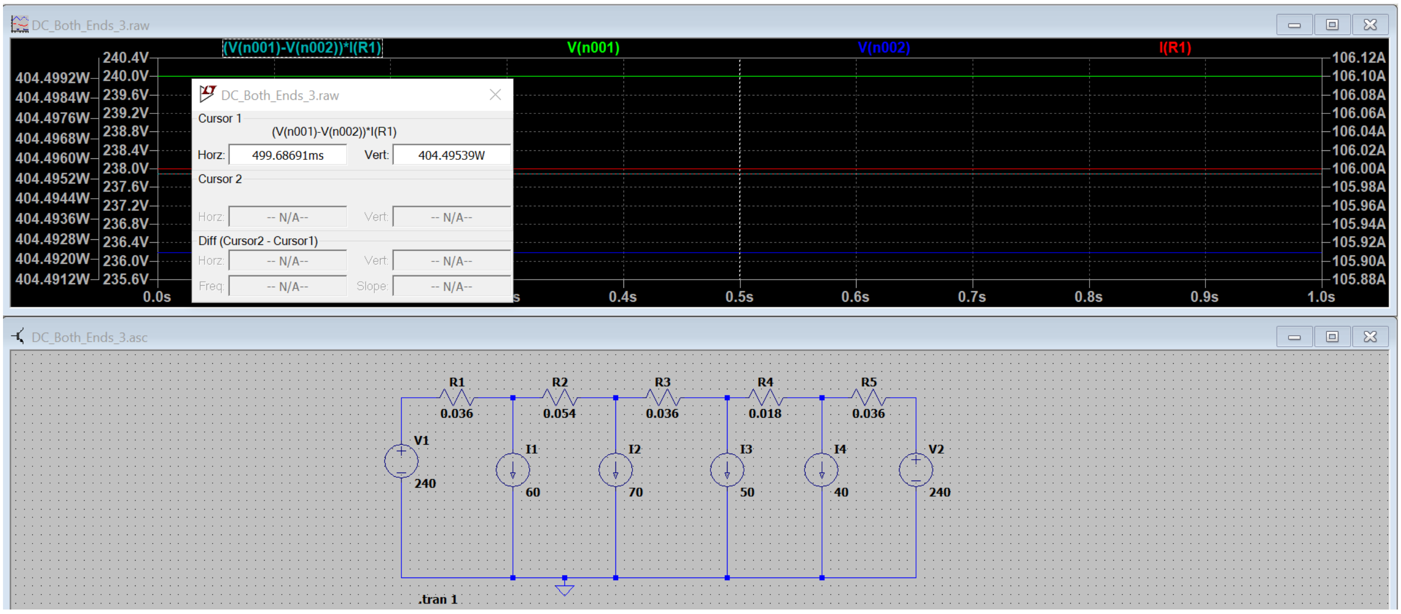
To show the potential at a point relative to the ground in LTspice, click on this point with the probe. We obtain the following potentials at load points: $V_A = V_B = 240$ V, $V_C = 236.184$ V, $V_D = 233.7$ V, $V_E = 234.564$ V, and $V_F = 235.896$ V. The minimum potential is achieved at D. Using the 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 $\eta = \frac{P_s - P_d}{P_s} \times 100\% \approx 98\%$ (same as calculated above).

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):

```
#
# 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