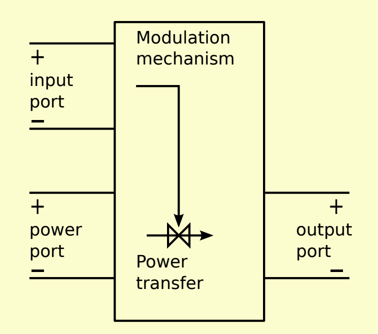
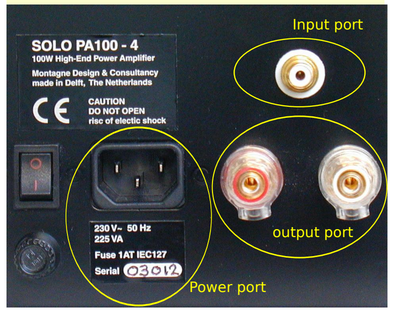
Amplifiers, concept, types and ideal behavioral models





Definition

Amplifier object

Three electrical ports
- input port: connection to signal source
- output port: connection to load
- power port: connection to power supply

Amplification function
- provide load with accurate copy of source signal

Characteristic property
- Available power gain exceeds unity

Functional model

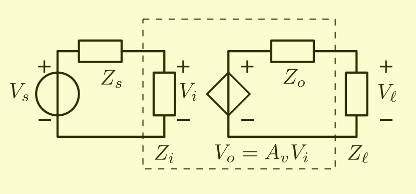
- Two-port input and output port only
- Active (delivers power)
- Linear, instantaneous, and time-invariant:

$$y(t) = A\,x(t)$$
$$A = \text{constant}$$

Example voltage amplifier

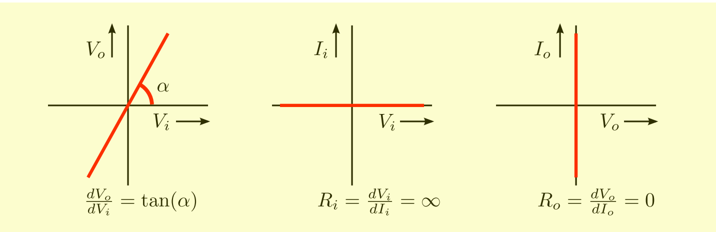
Source
- Information accurately related to open-circuit voltage
- Source impedance inaccurately known

Load
- Information accurately related to driving voltage
- Load impedance inaccurately known

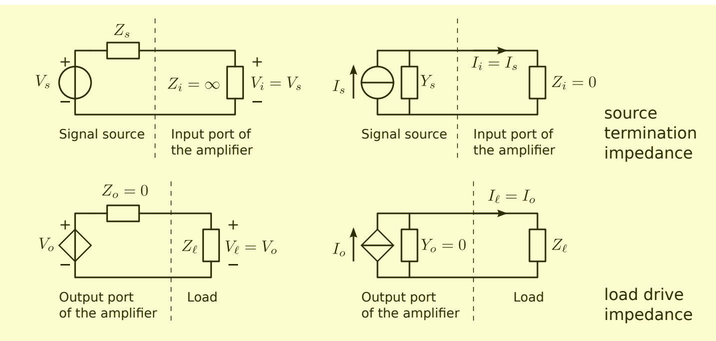


$$\frac{V_\ell}{V_s} = \frac{Z_i}{Z_i+Z_s} A_v \frac{Z_\ell}{Z_\ell+Z_o}$$

Ideal characteristics



Source termination impedance and load drive impedance



Amplifier types

Follow from best source termination and load drive conditions for accurate signal transfer

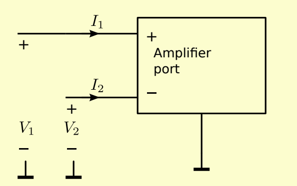
Type	Zi	Zo	A	B	C	D
voltage amplifier	infinite	0	A	0	0	0
transadmittance	infinite	infinite	0	B	0	0
transimpedance	0	0	0	0	C	0
current amplifier	0	infinite	0	0	0	D
voltage to voltage / current	infinite	Ro	A	B	0	0
current to voltage / current	0	Ro	0	0	C	D
voltage / current to voltage	Ri	0	A	0	C	0
voltage / current to current	Ri	infinite	0	B	0	D
voltage / current to voltage /current	Ri	Ro	A	B	C	D

Port isolation properties

input-output	input-power	output-power	configuration
non-isolated	non-isolated	non-isolated	common-ground
non-isolated	non-isolated	isolated	x
non-isolated	isolated	non-isolated	differential receiver
non-isolated	isolated	isolated	floating supply
isolated	non-isolated	non-isolated	x
isolated	non-isolated	isolated	differential driver
isolated	isolted	non-isolated	x
isolated	isolated	isolated	differential receiver / driver

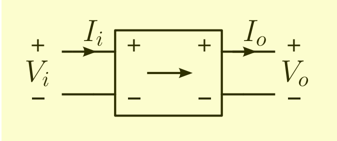
Floating port modeling and characterization

$$V_{cm} = \frac{V_1+V_2}{2}$$
$$V_{dm} = V_1 - V_2 \qquad Z_{cm} = \frac{V_{cm}}{I_{cm}}$$
$$I_{cm} = I_1 + I_2 \qquad Z_{dm} = \frac{V_{dm}}{I_{dm}}$$
$$I_{dm} = \frac{I_1-I_2}{2}$$



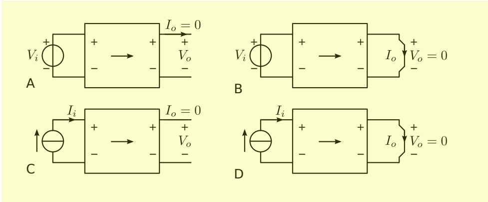
Modeling of ideal behavior (natural two-port)

Transmision-1 matrix representation



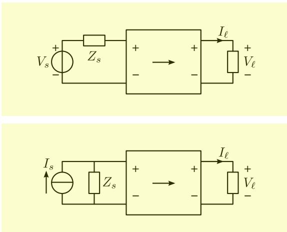
Anti-causal notation:

$$\begin{pmatrix} V_i \\ I_i \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} V_o \\ I_o \end{pmatrix}$$



$$A = \frac{1}{\mu} = \left.\frac{V_i}{V_o}\right|_{I_o=0},$$
$$B = \frac{1}{\gamma} = \left.\frac{V_i}{I_o}\right|_{V_o=0},$$
$$C = \frac{1}{\zeta} = \left.\frac{I_i}{V_o}\right|_{I_o=0},$$
$$D = \frac{1}{\alpha} = \left.\frac{I_i}{I_o}\right|_{V_o=0}.$$

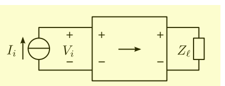
Source-to-load transfer

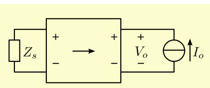


$$A_v = \frac{V_\ell}{V_s} = \frac{1}{A+B\frac{1}{Z_\ell}+CZ_s+D\frac{Z_s}{Z_\ell}}$$
$$A_y = \frac{I_\ell}{V_s} = \frac{1}{AZ_\ell+B+CZ_\ell Z_s+DZ_s}$$
$$A_z = \frac{V_\ell}{I_s} = \frac{1}{A\frac{1}{Z_s}+B\frac{1}{Z_s Z_\ell}+C+D\frac{1}{Z_\ell}}$$
$$A_i = \frac{I_\ell}{I_s} = \frac{1}{A\frac{Z_\ell}{Z_s}+B\frac{1}{Z_s}+CZ_\ell+D}$$

Port impedances

$$Z_i = \frac{V_i}{I_i} = \frac{AZ_\ell+B}{CZ_\ell+D}$$





$$Z_o = \frac{V_i}{I_i} = \frac{DZ_s+B}{CZ_s+A}$$

For design purposes: source-to-load transfers and port impedances can be expressed in terms of the T1 matrix parameters (A,B,C,D) and the source and the load impedance.

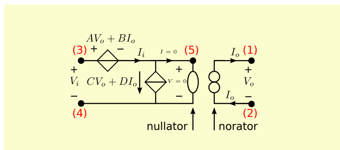
Unilateral amplifier types

Zero reverse transfer

$$AC = BD$$

Nullor and six non-unilateral types not listed in the table

Network model



MNA matrix stamp

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ C & -C & 0 & 0 & 0 & D & 1 \\ -C & C & 0 & 0 & 0 & -D & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ -A & A & 1 & 0 & -1 & -B & 0 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ I_o \\ I_i \end{pmatrix}$$

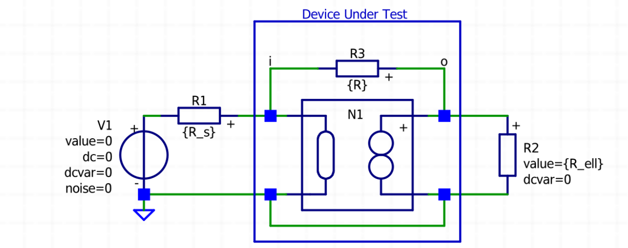
SLiCAP test bench for determination of the T1 matrix parameters

—	Variable	—	SLiCAP or Python keyword
—	Comment	—	String variable

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Python test bench for determination of the T1 parameters
- Use source identifier 'V1'
- Use load resistor value 'R_ell'
- Use the same file name for the .asc, .cir, .PNG, and .SVG
  files associated with a circuit.
"""
from SLiCAP import *

prj = initProject('ABCD-test')

# Create a list with circuit file names if you want to run multiple circuits.
fileNames = ['ABCD-test']

for fileName in fileNames:

    #makeNetlist(fileName + '.asc', fileName)

    # Define an instruction
    i1 = instruction()
    i1.setCircuit(fileName + '.cir')

    i1.setSimType('symbolic')
    i1.setGainType('gain')
    i1.setDataType('laplace')

    # Use the same identifier for the source in all files
    i1.setSource('V1')

    # Define the detector for determination of the input voltage
    i1.setDetector('V_i')
    result = i1.execute()
    # V_i is the transfer from V1 to the voltage V_i at node i
    V_i = result.laplace

    # Define the detector for determination of the input current
    i1.setDetector('I_V1')
    result = i1.execute()
    # I_i is the transfer from V1 to the current through V1
    I_i = -result.laplace

    # Define the detector for determination of the output voltage
    i1.setDetector('V_o')
    result = i1.execute()
    # V_o is the transfer from V1 to the output voltage
    V_o = result.laplace

    # Define the detector for determination of the output current
    i1.setDetector('I_R2')
    result = i1.execute()
    # I_o is the transfer from V1 to the current through R2
    I_o = result.laplace

    # Calculate the T1 parameters
    # Use the same name for the load resistance in all files
    R_ell = sp.Symbol('R_ell')
    A     = sp.simplify(sp.limit(V_i/V_o, R_ell, 'oo'))
    B     = sp.simplify(sp.limit(V_i/I_o, R_ell, 0))
    C     = sp.simplify(sp.limit(I_i/V_o, R_ell, 'oo'))
    D     = sp.simplify(sp.limit(I_i/I_o, R_ell, 0))

    htmlPage('Determination of T1 matrix parameters')
    head2html('Test circuit')
    img2html(fileName + '.svg', 400)
    head2html('T1 matrix of the device under test')
    text2html('The T1 matrix of the device under test is found as:')
    T1 = sp.Matrix([[A,B], [C,D]])
    eqn2html('T_1', T1)
    Vili = sp.Matrix([[sp.Symbol('V_i')], [sp.Symbol('I_i')]])
    Volo = sp.Matrix([[sp.Symbol('V_o')], [sp.Symbol('I_o')]])
    # Display the matrix equation without execution of the multiplication
    text2html('The matrix equation for the two-port (DUT) is found as:')
    text2html('$$' + sp.latex(Vili) + "=" + sp.latex(T1) + '\\cdot' +
              sp.latex(Volo) + '$$')
    # Display the matrix equation with execution of the multiplication
    text2html('This can be written as:')
    eqn2html(Vili, T1*Volo)
```

Determination of T1 matrix parameters

Test circuit



T1 matrix of the device under test

The T1 matrix of the device under test is found as:

$$T_1 = \begin{bmatrix} 0 & 0 \\ -\frac{1}{R} & 0 \end{bmatrix} \tag{1}$$

The matrix equation for the two-port (DUT) is found as:

$$\begin{bmatrix} V_i \\ I_i \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -\frac{1}{R} & 0 \end{bmatrix} \cdot \begin{bmatrix} V_o \\ I_o \end{bmatrix}$$

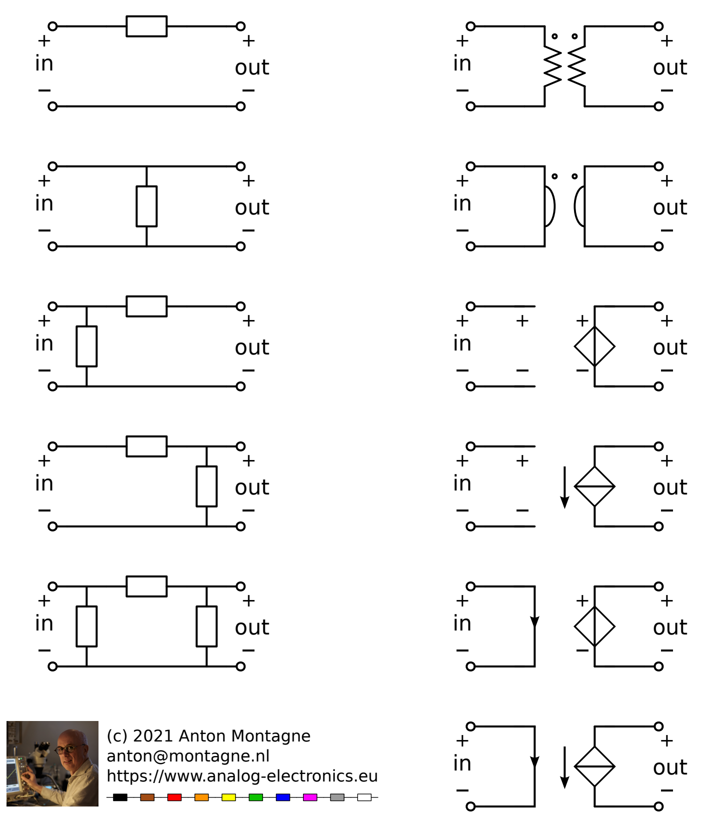
This can be written as:

$$\begin{bmatrix} V_i \\ I_i \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{V_o}{R} \end{bmatrix} \tag{2}$$

Go to ABCD-test_index

Try yourself and verify the answer with SLiCAP

Assign symbolic values to the elements and determine the T1 transmission parameters by hand. Verify the obtained results with SLiCAP



(c) 2021 Anton Montagne
anton@montagne.nl
https://www.analog-electronics.eu