Electrical Circuits I

- This lecture discusses the mathematical modeling of simple electrical linear circuits.
- When modeling a circuit, one ends up with a set of implicitly formulated algebraic and differential equations (DAEs), which in the process of horizontal and vertical sorting are converted to a set of explicitly formulated algebraic and differential equations.
- By eliminating the algebraic variables, it is possible to convert these DAEs to a state-space representation.

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich
Mathematical Modeling of Physical Systems

Table of Contents

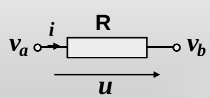
- Components and their models
- The circuit topology and its equations
- An example
- Horizontal sorting
- Vertical sorting
- State-space representation
- Transformation to state-space form

September 20, 2012
© Prof. Dr. François E. Cellier
Start of Presentation

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich
Mathematical Modeling of Physical Systems

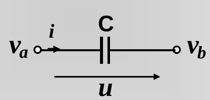
Linear Circuit Components

- Resistors



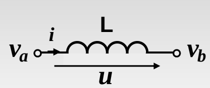
$u = v_a - v_b$
$u = R \cdot i$

- Capacitors



$u = v_a - v_b$
$i = C \cdot \frac{du}{dt}$

- Inductors



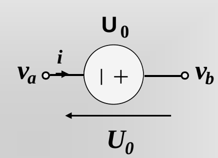
$u = v_a - v_b$
$u = L \cdot \frac{di}{dt}$

September 20, 2012
© Prof. Dr. François E. Cellier
Start of Presentation

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich
Mathematical Modeling of Physical Systems

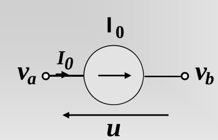
Linear Circuit Components II

- Voltage sources



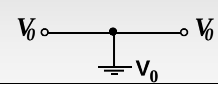
$U_0 = v_b - v_a$
$U_0 = f(t)$

- Current sources



$u = v_b - v_a$
$I_0 = f(t)$

- Ground

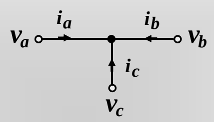


$V_0 = 0$

September 20, 2012
© Prof. Dr. François E. Cellier
Start of Presentation

Circuit Topology

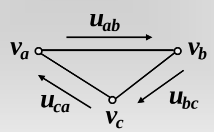
- Nodes



$v_a = v_b = v_c$

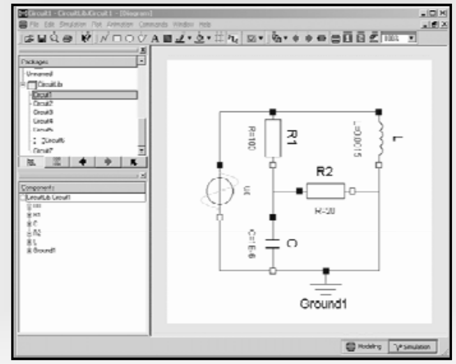
$i_a + i_b + i_c = 0$

- Meshes



$u_{ab} + u_{bc} + u_{ca} = 0$

An Example I



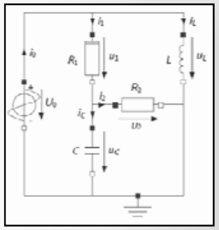
Rules for Systems of Equations I

- The component and topology equations contain a certain degree of redundancy.
- For example, it is possible to eliminate all potential variables (v_i) without problems.
- The current node equation for the ground node is redundant and is not used.
- The mesh equations are only used if the potential variables are being eliminated. If this is not the case, they are redundant.

Rules for Systems of Equations II

- If the potential variables are eliminated, every circuit component defines two variables: the current (i) through the element and die Voltage (u) across the element.
- Consequently, we need two equations to compute values for these two variables.
- One of the equations is the constituent equation of the element itself, the other comes from the topology.

An Example II



The circuit contains 5 components

⇒ We require 10 equations in 10 unknowns

Component equations:

$U_0 = f(t)$ $\quad$ $i_C = C \cdot du_C/dt$

$u_1 = R_1 \cdot i_1$ $\quad$ $u_L = L \cdot di_L/dt$

$u_2 = R_2 \cdot i_2$

Node equations:

$i_0 = i_1 + i_L$ $\quad$ $i_1 = i_2 + i_C$

Mesh equations:

$U_0 = u_1 + u_C$ $\quad$ $u_L = u_1 + u_2$

$u_C = u_2$

Rules for Horizontal Sorting I

- The time *t* may be assumed as known.
- The state variables (variables that appear in differentiated form) may be assumed as known.

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

$\Rightarrow$

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

Rules for Horizontal Sorting II

- Equations that contain only one unknown must be solved for it.
- The solved variables are now known.

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

$\Rightarrow$

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

Rules for Horizontal Sorting III

- Variables that show up in only one equation must be solved for using that equation.

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

$\Rightarrow$

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

Rules for Horizontal Sorting IV

- All rules may be used recursively.

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

$\Rightarrow$

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

$\Rightarrow$

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

$\Downarrow$

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

The algorithm is applied, until every equation defines exactly one variable that is being solved for.

Rules for Horizontal Sorting V

- The horizontal sorting can now be performed using symbolic formula manipulation techniques.

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$u_1 = R_1 \cdot i_1$	$i_1 = i_2 + i_C$
$u_2 = R_2 \cdot i_2$	$U_0 = u_1 + u_C$
$i_C = C \cdot du_C/dt$	$u_C = u_2$
$u_L = L \cdot di_L/dt$	$u_L = u_1 + u_2$

$\Rightarrow$

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$i_1 = u_1 / R_1$	$i_C = i_1 - i_2$
$i_2 = u_2 / R_2$	$u_1 = U_0 - u_C$
$du_C/dt = i_C / C$	$u_2 = u_C$
$di_L/dt = u_L / L$	$u_L = u_1 + u_2$

Rules for Vertical Sorting

- By now, the equations have become assignment statements. They can be sorted vertically, such that no variable is being used before it has been defined.

$U_0 = f(t)$	$i_0 = i_1 + i_L$
$i_1 = u_1 / R_1$	$i_C = i_1 - i_2$
$i_2 = u_2 / R_2$	$u_1 = U_0 - u_C$
$du_C/dt = i_C / C$	$u_2 = u_C$
$di_L/dt = u_L / L$	$u_L = u_1 + u_2$

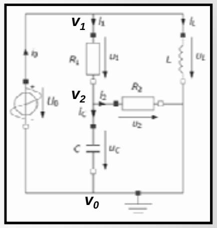
$\Rightarrow$

$U_0 = f(t)$	$i_2 = u_2 / R_2$
$u_1 = U_0 - u_C$	$i_C = i_1 - i_2$
$i_1 = u_1 / R_1$	$u_L = u_1 + u_2$
$i_0 = i_1 + i_L$	$du_C/dt = i_C / C$
$u_2 = u_C$	$di_L/dt = u_L / L$

Rules for Systems of Equations III

- Alternatively, it is possible to work with both potentials and voltages.
- In that case, additional equations for the node potentials must be found. These are the potential equations of the components and the potential equations of the topology. Those had been ignored before.
- The mesh equations are in this case redundant and can be ignored.

An Example III



The circuit contains 5 components and 3 nodes.

⇒ We require 13 equations in 13 unknowns.

Component equations:

$U_0 = f(t)$	$U_0 = v_1 - v_0$
$u_1 = R_1 \cdot i_1$	$u_1 = v_1 - v_2$
$u_2 = R_2 \cdot i_2$	$u_2 = v_2 - v_0$
$i_C = C \cdot du_C/dt$	$u_C = v_2 - v_0$
$u_L = L \cdot di_L/dt$	$u_L = v_1 - v_0$
$v_0 = 0$	

Node equations:

$i_0 = i_1 + i_L$ $\qquad$ $i_1 = i_2 + i_C$

Sorting

- The sorting algorithms are applied just like before.
- The sorting algorithm has already been reduced to a purely mathematical (informational) structure without any remaining knowledge of electrical circuit theory.
- Therefore, the overall modeling task can be reduced to two sub-problems:

  1. Mapping of the physical topology to a system of implicitly formulated DAEs.
  2. Conversion of the DAE system into an executable program structure.

State-space Representation

- ***Linear systems:***

$$\frac{d\mathbf{x}}{dt} = \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} \quad ; \quad \mathbf{x}(t_0) = \mathbf{x_0}$$
$$\mathbf{y} = \mathbf{C} \cdot \mathbf{x} + \mathbf{D} \cdot \mathbf{u}$$

- ***Non-linear systems:***

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\mathbf{u},t) \quad ; \quad \mathbf{x}(t_0) = \mathbf{x_0}$$
$$\mathbf{y} = \mathbf{g}(\mathbf{x},\mathbf{u},t)$$

$\mathbf{x} \in \Re^n$ $\qquad$ $\mathbf{A} \in \Re^{n \times n}$
$\mathbf{u} \in \Re^m$ $\qquad$ $\mathbf{B} \in \Re^{n \times m}$
$\mathbf{y} \in \Re^p$ $\qquad$ $\mathbf{C} \in \Re^{p \times n}$
$\mathbf{D} \in \Re^{p \times m}$

x = State vector
u = Input vector
y = Output vector

n = Number of state variables
m = Number of inputs
p = Number of outputs

Conversion to State-space Form I

$U_0 = f(t)$	$i_2 = u_2 / R_2$
$u_1 = U_0 - u_C$	$i_C = i_1 - i_2$
$i_1 = u_1 / R_1$	$u_L = u_1 + u_2$
$i_0 = i_1 + i_L$	$du_C/dt = i_C / C$
$u_2 = u_C$	$di_L/dt = u_L / L$

$\Rightarrow$

$$
\begin{aligned}
du_C/dt &= i_C / C \\
&= (i_1 - i_2) / C \\
&= i_1/C - i_2/C \\
&= u_1 / (R_1 \cdot C) - u_2 / (R_2 \cdot C) \\
&= (U_0 - u_C) / (R_1 \cdot C) - u_C / (R_2 \cdot C)
\end{aligned}
$$

$$
\begin{aligned}
di_L/dt &= u_L / L \\
&= (u_1 + u_2) / L \\
&= u_1 / L + u_2 / L \\
&= (U_0 - u_C) / L + u_C / L \\
&= U_0 / L
\end{aligned}
$$

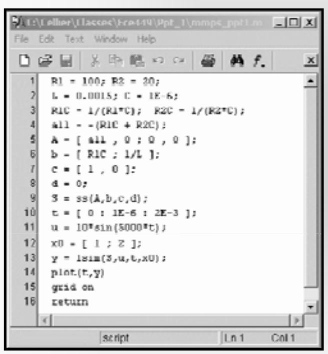
For each equation defining a state derivative, we substitute the variables on the right-hand side by the equations defining them, until the state derivatives depend only on state variables and inputs.

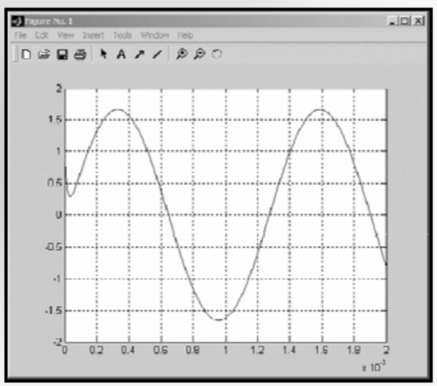
Conversion to State-space Form II

We let:

$$
\begin{aligned}
x_1 &= u_C \\
x_2 &= i_L \\
u &= U_0 \\
y &= u_C
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
\dot{x}_1 &= -\left[\frac{1}{R_1 \cdot C} + \frac{1}{R_2 \cdot C} \right] \cdot x_1 + \frac{1}{R_1 \cdot C} \cdot u \\
\dot{x}_2 &= \frac{1}{L} \cdot u \\
y &= x_1
\end{aligned}
$$

An Example IV





References

- Cellier, F.E. (1991), *Continuous System Modeling*, Springer-Verlag, New York, Chapter 3.
- Cellier, F.E. (2001), Matlab code of circuit example.