MDF (continuare): Analiza circuitelor liniare în regim tranzitoriu

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" Bucureşti, Facultatea de Inginerie Electrică

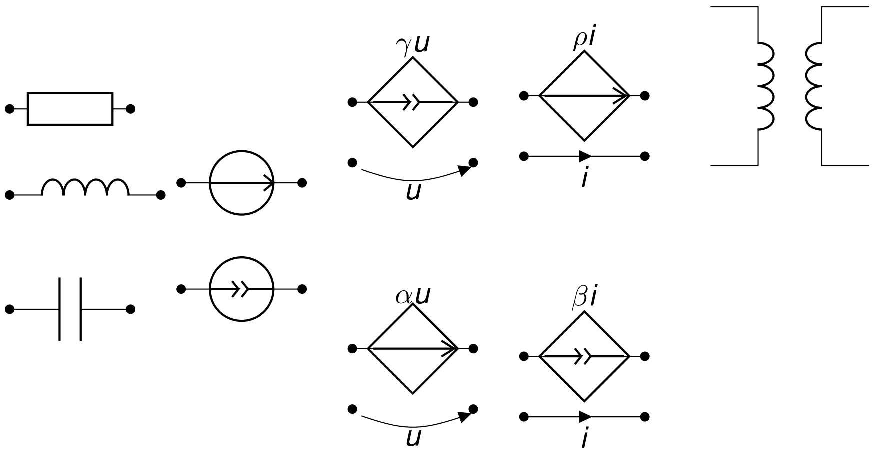
Suport didactic pentru disciplina *Metode Numerice*, 2017-2018

Notes

Cuprins

Notes

Tipuri de elemente ideale



Liniare!

Notes

Analiza circuitelor electrice liniare în regim tranzitoriu

Date:

- *Topologia circuitului* (graful circuitului) - poate fi descris:
  - geometric;
  - numeric (matrice topologice/ *netlist*);
- Pentru fiecare latură k:
  - tipul laturii (R,L,C,M,SUCU,SICI,SICU,SUCI, SIT,SIC);
  - caracteristica constitutivă
    - R_k, C_k, L_k, L_{kj};
    - parametrul de transfer α, β, γ, ρ;
    - semnalul de comandă (curent/tensiune, latură/noduri);
    - dep. de timp a parametrului: ($e_k(t)$, $j_k(t)$, $t_{\min} < t < t_{\max}$)
- Condiţiile iniţiale:
  - curenţii prin bobine $i_{Lk}(t_{\min})$
  - tensiunile la bornele condensatoarelor $u_{Ck}(t_{\min})$

Se cer: $i_k(t)$, $u_k(t)$, $k = 1, 2, \ldots, L$.

Notes

Analiza circuitelor electrice liniare în regim tranzitoriu

Problema fundamentală este bine formulată dacă are soluţie şi aceasta este unică.

- O *condiţie necesară* de formulare corectă: circuitul să aibă un arbore normal care să conţină toate SIT şi nicio SIC (SIT nu formează bucle, SIC nu formează secţiuni).

Vom reveni asupra acestui aspect.

Notes

Ca la c.c.

1. Kirchhoff I
2. Kirchhoff II
3. Ecuaţii constitutive pentru elementele rezistive:
   - laturi de tip SRC, SRT;
   - laturi de tip SIC, SIT;
   - laturi de tip SUCU, SICI, SUCI, SICU - comandate liniar.

relaţii algebrice

DAR

Notes

Diferit de c.c.

Ecuaţii constitutive pentru elementele reactive:

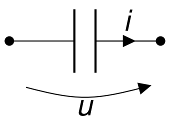
- bobine;
- condensatoare;
- bobine cuplate.

relaţii diferenţiale

Sistemul de rezolvat va fi un sistem diferenţial-algebric DAE

Notes

Condensatorul ideal liniar



Regula receptoare:

$$i = C\frac{\mathrm{d}u}{\mathrm{d}t} \quad \Leftrightarrow \quad u(t) = u(0) + \frac{1}{C}\int_0^t i(t')\,\mathrm{d}t'. \tag{1}$$

Obs: relaţia constitutivă este liniară doar dacă $u(0) = 0$.

Puterea convenţional primită:

$$p = ui = uC\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{Cu^2}{2}\right) = \frac{\mathrm{d}W}{\mathrm{d}t}, \tag{2}$$

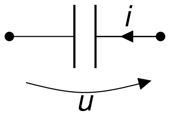
unde

$$W = Cu^2 > 0, \quad \text{dacă} \quad C > 0. \tag{3}$$

Tensiunea condensatorului este *variabilă de stare* (determină energia şi este o funcţie continuă de timp).

Notes

Condensatorul ideal liniar



Regula generatoare:

$$i = -C\frac{\mathrm{d}u}{\mathrm{d}t} \quad \Leftrightarrow \quad u(t) = u(0) - \frac{1}{C}\int_0^t i(t')\,\mathrm{d}t'. \tag{4}$$

Obs: relaţia constitutivă este liniară doar dacă $u(0) = 0$.
Puterea: convenţional cedată $p = ui \Rightarrow$ convenţional primită:

$$p = -ui = uC\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{Cu^2}{2}\right) = \frac{\mathrm{d}W}{\mathrm{d}t}, \tag{5}$$

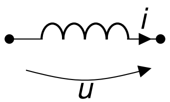
unde

$$W = Cu^2 > 0, \quad \text{dacă} \quad C > 0. \tag{6}$$

Tensiunea condensatorului este *variabilă de stare* (determină energia şi este o funcţie continuă de timp).

Notes

Bobina ideală liniară



Regula receptoare:

$$u = L\frac{\mathrm{d}i}{\mathrm{d}t} \quad \Leftrightarrow \quad i(t) = i(0) + \frac{1}{L}\int_0^t u(t')\,\mathrm{d}t'. \tag{7}$$

Obs: relaţia constitutivă este liniară doar dacă $i(0) = 0$.
Puterea convenţional primită:

$$p = ui = iL\frac{\mathrm{d}i}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{Li^2}{2}\right) = \frac{\mathrm{d}W}{\mathrm{d}t}, \tag{8}$$

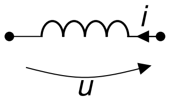
unde

$$W = Li^2 > 0, \quad \text{dacă} \quad L > 0. \tag{9}$$

Curentul prin bobină este *variabilă de stare* (determină energia şi este o funcţie continuă de timp).

Notes

Bobina ideală liniară



Regula generatoare:

$$u = -L\frac{\mathrm{d}i}{\mathrm{d}t} \quad \Leftrightarrow \quad i(t) = i(0) - \frac{1}{L}\int_0^t u(t')\,\mathrm{d}t'. \tag{10}$$

Obs: relaţia constitutivă este liniară doar dacă $i(0) = 0$.
Puterea: convenţional cedată $p = ui \Rightarrow$ convenţional primită:

$$p = -ui = iL\frac{\mathrm{d}i}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{Li^2}{2}\right) = \frac{\mathrm{d}W}{\mathrm{d}t}, \tag{11}$$

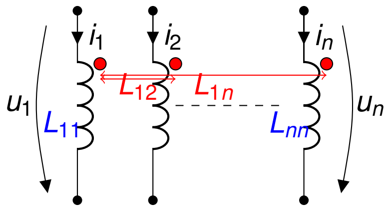
unde

$$W = Li^2 > 0, \quad \text{dacă} \quad L > 0. \tag{12}$$

Curentul prin bobină este *variabilă de stare* (determină energia şi este o funcţie continuă de timp).

Notes

Perechea de bobine cuplate



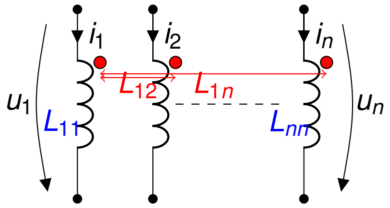
$$\mathbf{u} = \mathbf{L}\frac{\mathrm{d}\mathbf{i}}{\mathrm{d}t}, \tag{13}$$

$$\mathbf{L} = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1n} \\ L_{21} & L_{22} & \cdots & L_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ L_{n1} & L_{n2} & \cdots & L_{nn} \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad \mathbf{i} = \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix} \in \mathbb{R}^n, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \in \mathbb{R}^n, \tag{14}$$

L - matricea inductanţelor, simetrică: $L_{kj} = L_{jk}$ $k = j$: *inductanţe proprii*; $k \neq j$: *inductanţe mutuale*.

Notes

Perechea de bobine cuplate



$$\mathbf{u} = \mathbf{L}\frac{\mathrm{d}\mathbf{i}}{\mathrm{d}t}, \qquad (13)$$

Regula standard:

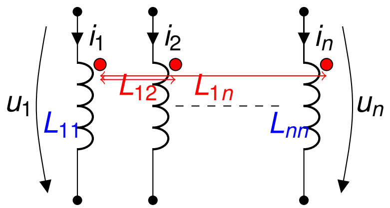
- pentru fiecare bobină: regula de la receptoare
- toţi curenţii intră în bobine prin bornele polarizate.

Schimbarea bornei polarizate (care are caracter convenţional) determină schimbarea semnului inductanţei mutuale.

Notes

Introducere
MDF: Circuite discretizate
Tipuri de elemente ideale de circuit
Formularea problemei
Ecuaţii

Perechea de bobine cuplate



$$\mathbf{u} = \mathbf{L}\frac{\mathrm{d}\mathbf{i}}{\mathrm{d}t}, \qquad (13)$$

Puterea convenţional primită:

$$p = \mathbf{i}^T\mathbf{u} = \mathbf{i}^T\mathbf{L}\frac{\mathrm{d}\mathbf{i}}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}\mathbf{i}^T\mathbf{L}\mathbf{i}\right) = \frac{\mathrm{d}W}{\mathrm{d}t}, \qquad (14)$$

unde

$$W = \frac{1}{2}\mathbf{i}^T\mathbf{L}\mathbf{i} > 0, \qquad (15)$$

dacă $\mathbf{L}$ e pozitiv definită $\Leftrightarrow$ $L_{kk} > 0$ şi $|L_{kj}| < \sqrt{L_{kk}L_{jj}}$.

Notes

Metoda diferenţelor finite

Prin rezolvarea numerică se vor obţine valori aproximative ale mărimilor într-o mulţime discretă de valori ale timpului notate

$$t_0 = t_{\min},\ t_1,\ t_2, \ldots,\ t_n = t_{\max}.$$

Valorile mărimilor în aceste momente de timp vor fi notate

$$u_k^{(j)} \approx u_k(t_j), \qquad i_k^{(j)} \approx i_k(t_j)$$

- $k = 1, \ldots, L$ este un indice de latură,
- $j = 1, \ldots, n$ reprezintă momentul de timp t_j.

Notes

Metoda diferenţelor finite

Ideea:

Discretizarea ecuaţilor cu derivate:

- se va scrie ecuaţia la momentul de timp t_j;
- pentru aproximarea numerică a derivatei se va folosi o formulă de diferenţe finite regresive de ordinul 1 (Euler implicit):

$$\frac{df}{dt}(t_j) \approx \frac{f^{(j)} - f^{(j-1)}}{t_j - t_{j-1}}$$

unde $f^{(j)} \approx f(t_j)$. Pentru simplificare, pp.:

$$t_{\min} = 0 \qquad t_j - t_{j-1} = h$$

$$\Rightarrow t_1 = h, t_2 = 2h, \ldots, t_j = jh, \ldots, t_n = nh = t_{\max}.$$

Notes

Circuitul discretizat asociat bobinei

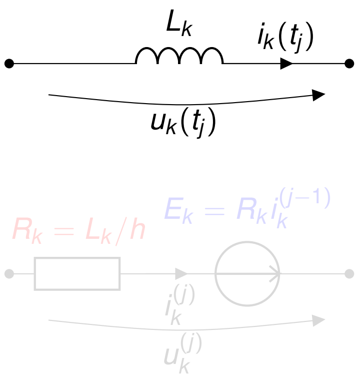
$$u_k(t_j) = L_k \frac{\mathrm{d}i_k}{\mathrm{d}t}(t_j)$$

discretizată:

$$u_k^{(j)} = L_k \frac{i_k^{(j)} - i_k^{(j-1)}}{h}$$

$$u_k^{(j)} = \frac{L_k}{h} i_k^{(j)} - \frac{L_k}{h} i_k^{(j-1)}$$

$$u_k^{(j)} = R_k i_k^{(j)} - E_k$$



Notes

Circuitul discretizat asociat bobinei

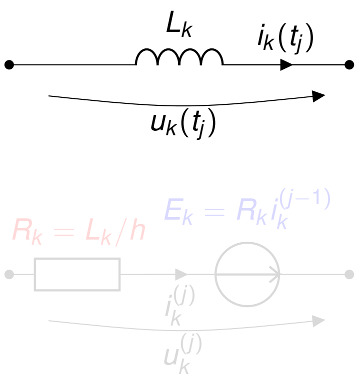
$$u_k(t_j) = L_k \frac{\mathrm{d}i_k}{\mathrm{d}t}(t_j)$$

discretizată:

$$u_k^{(j)} = L_k \frac{i_k^{(j)} - i_k^{(j-1)}}{h}$$

$$u_k^{(j)} = \frac{L_k}{h} i_k^{(j)} - \frac{L_k}{h} i_k^{(j-1)}$$

$$u_k^{(j)} = R_k i_k^{(j)} - E_k$$



Notes

Circuitul discretizat asociat bobinei

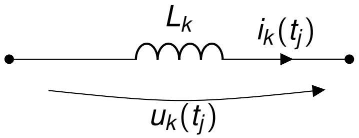
$$u_k(t_j) = L_k \frac{\mathrm{d}i_k}{\mathrm{d}t}(t_j)$$

discretizată:

$$u_k^{(j)} = L_k \frac{i_k^{(j)} - i_k^{(j-1)}}{h}$$

$$u_k^{(j)} = \frac{L_k}{h} i_k^{(j)} - \frac{L_k}{h} i_k^{(j-1)}$$

$$u_k^{(j)} = R_k i_k^{(j)} - E_k$$



Notes

Introducere
MDF: Circuite discretizate
Schema de discretizare în timp
Circuite companion
Algoritmul metodei

Circuitul discretizat asociat bobinei

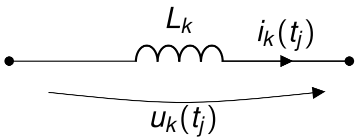
$$u_k(t_j) = L_k \frac{\mathrm{d}i_k}{\mathrm{d}t}(t_j)$$

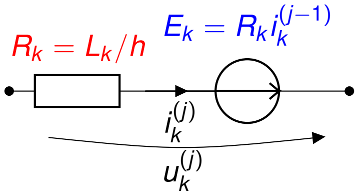
discretizată:

$$u_k^{(j)} = L_k \frac{i_k^{(j)} - i_k^{(j-1)}}{h}$$

$$u_k^{(j)} = \frac{L_k}{h} i_k^{(j)} - \frac{L_k}{h} i_k^{(j-1)}$$

$$u_k^{(j)} = R_k i_k^{(j)} - E_k$$





15/32
Gabriela Ciuprina
MDF - Circuite liniare în regim trazitoriu

Notes

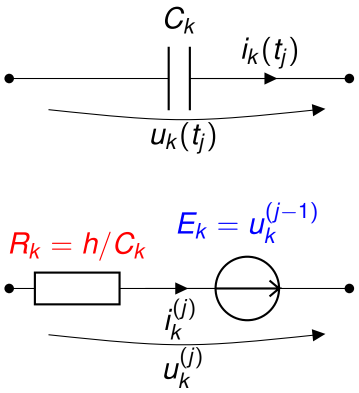
Circuitul discretizat asociat condensatorului

$$i_k(t_j) = C_k \frac{\mathrm{d}u_k}{\mathrm{d}t}(t_j)$$

discretizată:

$$i_k^{(j)} = C_k \frac{u_k^{(j)} - u_k^{(j-1)}}{h}$$

$$i_k^{(j)} = \frac{C_k}{h} u_k^{(j)} - \frac{C_k}{h} u_k^{(j-1)}$$



Notes

Introducere
MDF: Circuite discretizate
Schema de discretizare în timp
Circuite companion
Algoritmul metodei

Circuitul discretizat asociat condensatorului

$$i_k(t_j) = C_k \frac{\mathrm{d}u_k}{\mathrm{d}t}(t_j)$$

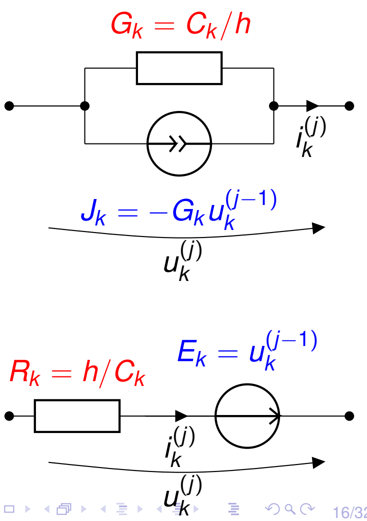
discretizată:

$$i_k^{(j)} = \frac{C_k}{h} u_k^{(j)} - \frac{C_k}{h} u_k^{(j-1)}$$

$$i_k^{(j)} = G_k u_k^{(j)} + J_k$$

$$u_k^{(j)} = \frac{1}{G_k} i_k^{(j)} - \frac{J_k}{G_k}$$

$$u_k^{(j)} = R_k i_k^{(j)} - E_k$$



Notes

Ideea algoritmului

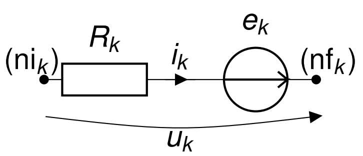
Se rezolvă o succesiune de circuite rezistive liniare.

$t = t_{\min}$
repetă
  $t = t + h$
  înlocuieşte elementele reactive cu schemele lor discrete
  rezolvă circuitul rezistiv liniar (sursele au valorile la t)
  calculează mărimile de stare
cât timp $t \le t_{\max}$

Notes

Cel mai simplu algoritm - pe ce ne bazăm

Primul algoritm scris pentru circuite rezistive liniare (crl) - laturi SRT



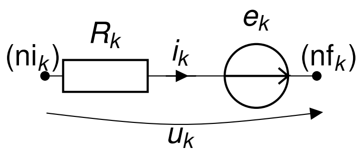
; declaratii date - varianta A

întreg N	; număr de noduri
întreg L	; număr de laturi
tablou întreg ni[L]	; noduri iniţiale ale laturilor
tablou întreg nf[L]	; noduri finale ale laturilor
tablou real R[L]	; rezistenţe
tablou real e[L]	; tensiuni electromotoare

Notes

Cel mai simplu algoritm - pe ce ne bazăm

Primul algoritm scris pentru circuite rezistive liniare (crl) - laturi SRT



; declaraţii date - varianta B
<u>înregistrare</u> circuit
    <u>întreg</u> *N* ; număr de noduri
    <u>întreg</u> *L* ; număr de laturi
    <u>tablou</u> <u>întreg</u> ni[*L*] ; noduri iniţiale ale laturilor
    <u>tablou</u> <u>întreg</u> nf[*L*] ; noduri finale ale laturilor
    <u>tablou</u> <u>real</u> *R*[*L*] ; rezistenţe
    <u>tablou</u> <u>real</u> *e*[*L*] ; tensiuni electromotoare
•

Notes

Cel mai simplu algoritm - pe ce ne bazăm

Să pp că avem la dispoziţie o procedură:

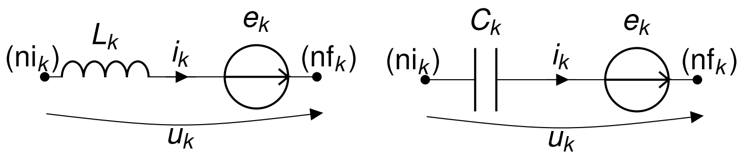
<u>procedură</u> nodal_crl(circuit,*v*)
; rezolvă un circuit rezistiv liniar cu metoda nodală
; date de intrare: structura circuit
; ieşire: valorile potenţialelor *v* în noduri, ultimul nod este de referinţă
. . .
<u>retur</u>

Obs: procedura cuprinde atât asamblarea sistemului de ecuaţii cât şi rezolvarea lui.

Notes

Cel mai simplu algoritm - ce e nou

- Admitem acum în plus, laturi L şi C;
- Putem presupune că pot avea în serie o SIT.



Obs:

- Pp. pentru început că valorile surselor sunt ct. în timp. Stare staţionară (dată de condiţiile iniţiale) $\rightarrow$ altă stare staţionară (impusă de topologie).
- Dacă $e_k(t)$ - modificarea (conceptuală) este minoră.

Notes

Cel mai simplu algoritm - ce e nou

Structura de date ce descrie circuitul în regim tranzitoriu trebuie extinsă:

```
; declaraţii date - varianta B
înregistrare circuit
        întreg N                      ; număr de noduri
        întreg L                      ; număr de laturi
        tablou întreg ni[L]           ; noduri iniţiale ale laturilor
        tablou întreg nf[L]           ; noduri finale ale laturilor
        tablou real e[L]              ; tensiuni electromotoare
        tablou caracter tip[L]        ; tipul laturii R/L/C
        tablou real p[L]              ; parametrul rezistenţă/inductivitate/capacitate
        tablou real IC[L]             ; condiţia iniţială
```

•

OBS: IC are sens doar pentru laturi de tip L/C.

Notes

Cel mai simplu algoritm - etapa de preprocesare

```
funcţie citire_date ()
; declaraţii
...
citeşte circuit.N, circuit.L
pentru k = 1,circuit.L
        citeşte circuit.ni_k, circuit.nf_k
        citeşte circuit.e_k, circuit.tip_k, circuit.p_k
        dacă circuit.tip_k = "L" sau circuit.tip_k = "C"
                citeşte circuit.IC_k
        •
citeşte tmin, tmax          ; intervalul de timp de simulare
citeşte h ; pasul de timp
•
întoarce circuit
```

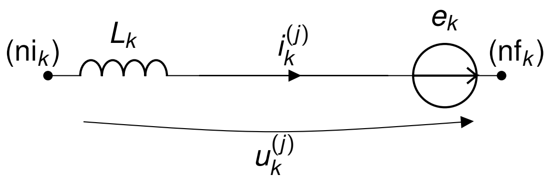

Notes

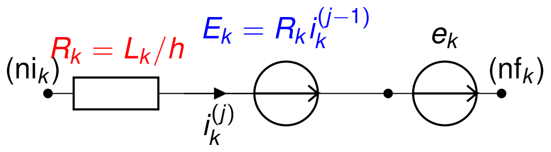
Cel mai simplu algoritm - etapa de rezolvare

```
procedură rezolvă_cl_tranz (circuit,tmin,tmax,h)
circuit_d.N = circuit.N
circuit_d.L = circuit.L
circuit_d.ni = circuit.ni
circuit_d.nf = circuit.nf
IC = circuit.IC
t = tmin
repetă
        t = t + h
        circuit_d.IC = IC
        pentru k = 1,L
                dacă circuit.tip(k) = "C"
                        circuit_d.R(k) = circuit.p(k)/h
                        circuit_d.e(k) = circuit.e(k) + circuit_d.R(k)*IC(k)
                altfel dacă circuit.tip(k) = "C"
                        circuit_d.R(k) = h/circuit.p(k)
                        circuit_d.e(k) = circuit.e(k) - IC(k)
                altfel ; latura este de tip "R"
                        circuit_d.R(k) = circuit.p(k)
                        circuit_d.e(k) = circuit.e(k)
                •
        •
        nodal_crl(circuit_d,v)
        ?
```


Notes

Cel mai simplu algoritm - etapa de rezolvare

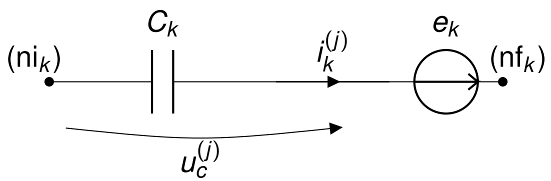


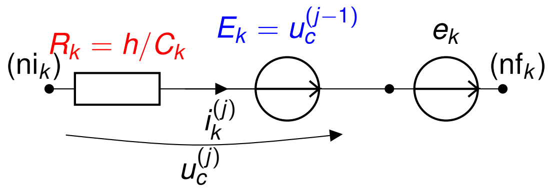


$$i_k^{(j)} = \frac{V(ni(k)) - V(nf(k)) + e_k + R_k i_k^{(j-1)}}{R_k} = i_k^{(j-1)} + \frac{V(ni(k)) - V(nf(k)) + e_k}{R_k}$$

Notes

Cel mai simplu algoritm - etapa de rezolvare





$$u_c^{(j)} = V(ni(k)) - V(nf(k)) + e_k$$

Notes

Cel mai simplu algoritm - etapa de rezolvare

```
nodal_crl(circuit_d,v)
pentru k = 1,L
        dacă circuit.tip(k) = "L"
                IC(k) = IC(k) + (v(ni(k))-v(nf(k))+ circuit.e(k))/circuit_d.R(k)
                scrie latura k, crt. prin bobina IC(k)
        altfel dacă circuit.tip(k) = "C"
                IC(k) = v(ni(k))-v(nf(k)) + circuit.e(k)
                scrie latura k, tens. pe condensator IC(k)
        •
•
cât timp t ≤ tmax
```


Notes

Cel mai simplu algoritm

Îmbunătăţiri sunt posibile, bazate pe următoarele observaţii:

- Dacă pasul de timp este constant, circuitul discretizat are aceleaşi conductanţe la fiecare iteraţie.
- Dacă regimul tranzitoriu aproape s-a stins, este o risipă de efort să folosim paşi mici de timp.
- Calculul altor mărimi din circuit s-ar putea face pe baza teoremei substituţiei.

Notes

Cel mai simplu algoritm

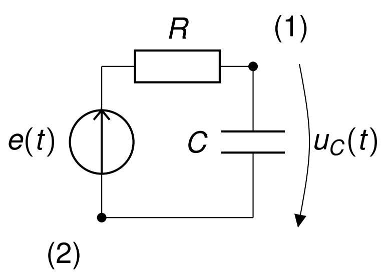
Îmbunătăţiri sunt posibile, bazate pe următoarele observaţii:

- Dacă pasul de timp este constant, circuitul discretizat are aceleaşi conductanţe la fiecare iteraţie.
- Dacă regimul tranzitoriu aproape s-a stins, este o risipă de efort să folosim paşi mici de timp.
- Calculul altor mărimi din circuit s-ar putea face pe baza teoremei substituţiei.

Idei de implementare?

Notes

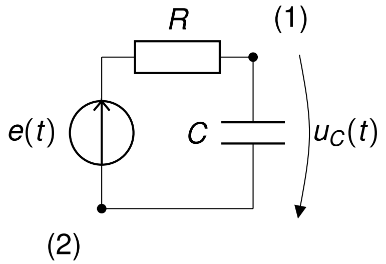
Exemplul 1

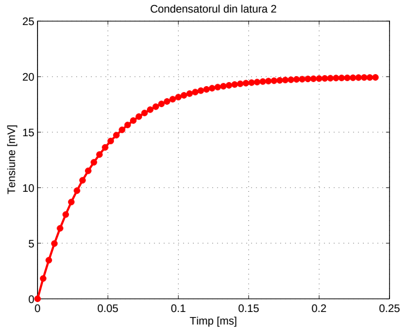



```
circuit.N = 2;
circuit.L = 2;
circuit.ni = [2; 1];
circuit.nf = [1; 2];
circuit.tip = ['R';'C'];
circuit.p = [10; 4e-6];
circuit.e = [20e-3; 0];
circuit.IC = [0; 0];
% info despre simularea dorita
simulare.tmin = 0;
simulare.tmax = 6*10*4e-6;
simulare.h = 4e-6;
```


Notes

Exemplul 1



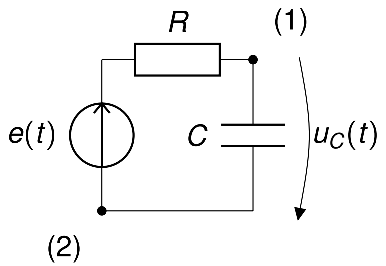


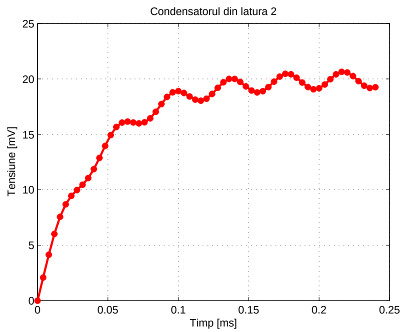
$$e(t) = 20\,\mathrm{step}(t)\ [\mathrm{mV}]$$

Notes

Introducere
MDF: Circuite discretizate
Schema de discretizare în timp
Circuite companion
Algoritmul metodei

Exemplul 1

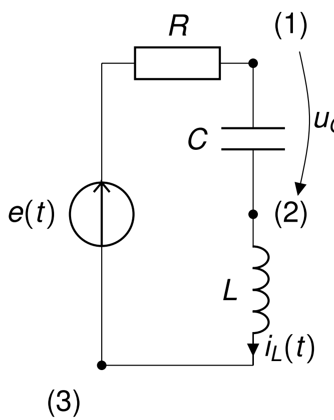




$$e(t) = 20 + 5\sin(157080t)\ [\mathrm{mV}]$$

Notes

Exemplul 2

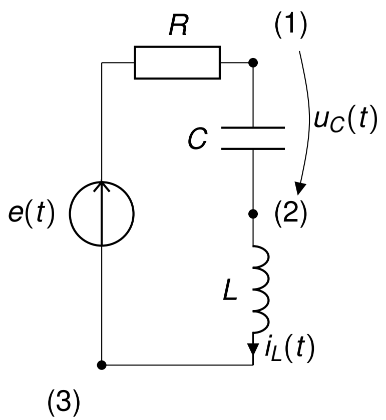


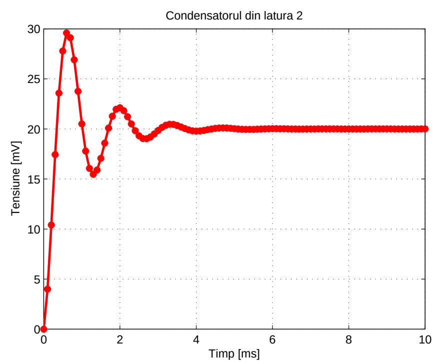

```
circuit.N = 3;
circuit.L = 3;
circuit.ni = [3; 1; 2];
circuit.nf = [1; 2; 3];
circuit.tip = ['R';'C';'L'];
% regim oscilant amortizat R/(2*L) < 1/sqrt(LC)
circuit.p = [0.001; 20e-6; 2e-3];
% regim critic R/(2*L) = 1/sqrt(LC)
% circuit.p(1) = 2*circuit.p(3)/sqrt(circuit.p(2)*circuit.p(3));
% regim aperiodic R/(2*L) > 1/sqrt(LC)
% circuit.p(1) = 8*circuit.p(3)/sqrt(circuit.p(2)*circuit.p(3));
circuit.e = [20e-3; 0; 0];
circuit.IC = [0; 0; 0];
% info despre simularea dorita
simulare.tmin = 0;
simulare.tmax = 10e-3;
simulare.h = 1e-4;
```


Notes

Introducere
MDF: Circuite discretizate
Schema de discretizare în timp
Circuite companion
Algoritmul metodei

Exemplul 2

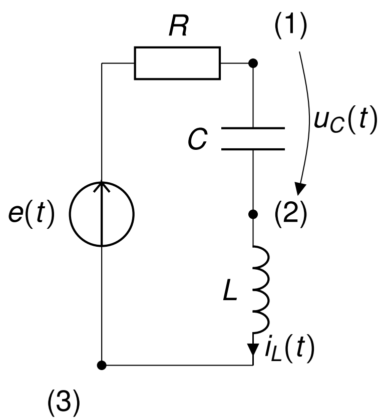


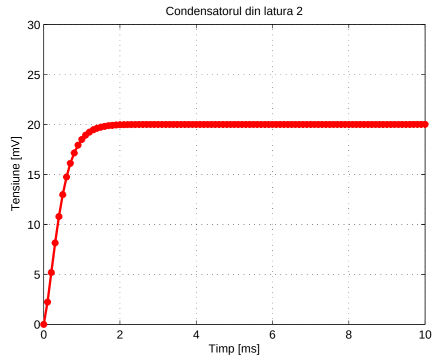


$e(t) = 20\,\mathrm{step}(t)$ [mV]
Regim oscilant amortizat.
$(R/(2 * L) < 1/\sqrt{LC})$

Notes

Exemplul 2

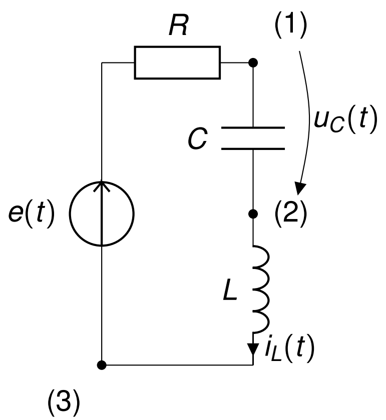


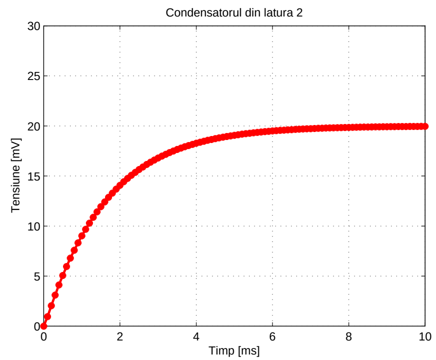


$e(t) = 20\,\mathrm{step}(t)$ [mV]
Regim critic. $(R/(2*L) = 1/\sqrt{LC})$

Notes

Exemplul 2

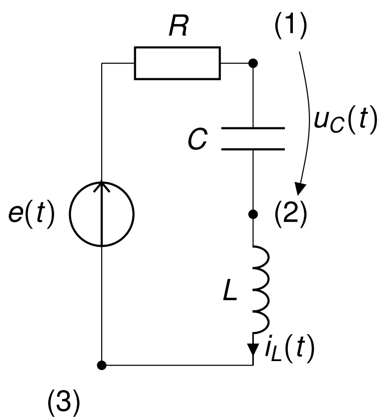


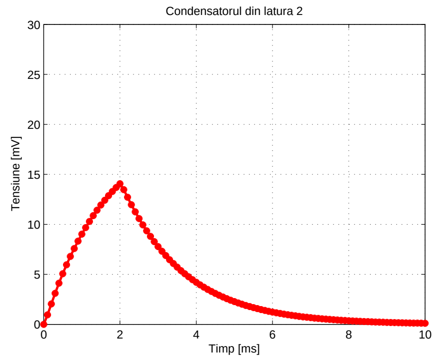


$e(t) = 20\,\mathrm{step}(t)$ [mV]
Regim aperiodic.
$(R/(2*L) > 1/\sqrt{LC})$

Notes

Exemplul 2

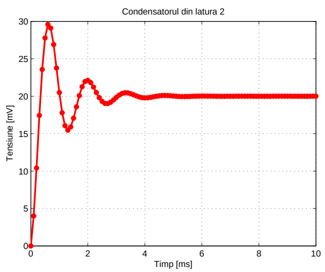


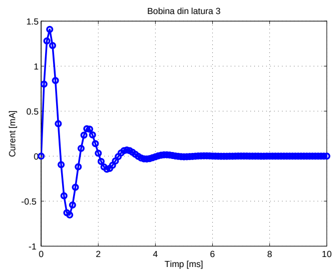


$e(t) = 20\,\text{step}(t)$ [mV], $t < 2$ s şi
$e(t) = 0, t \geq 2$ s.
Regim aperiodic.
$(R/(2 * L) > 1/\sqrt{LC})$

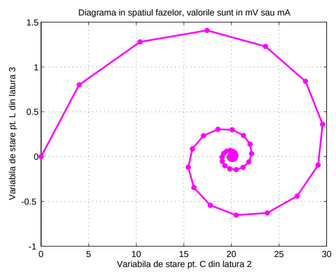
Notes

Exemplul 2 - variabile şi diagrama de stare



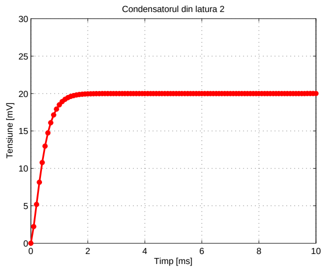


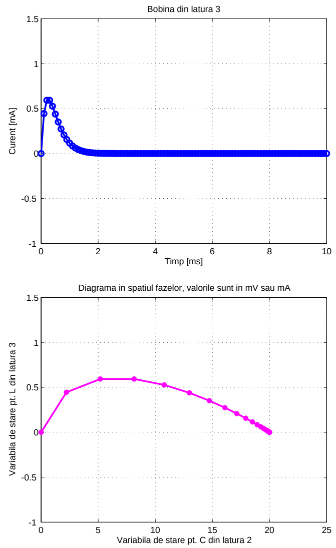
Regim oscilant amortizat.



Notes

Exemplul 2 - variabile şi diagrama de stare

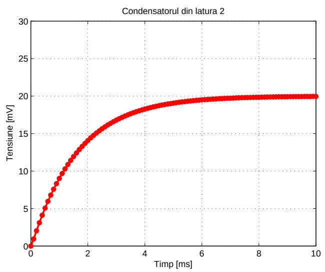


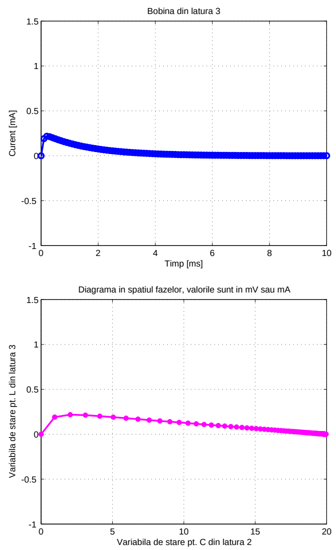


Regim critic.

Notes

Exemplul 2 - variabile şi diagrama de stare

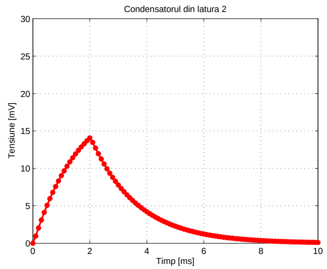


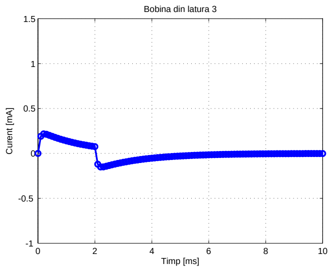


Regim aperiodic.

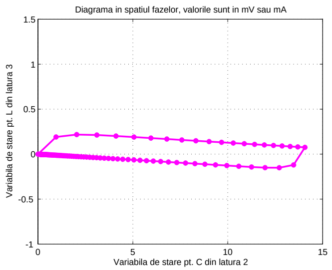
Notes

Exemplul 2 - variabile şi diagrama de stare





Regim aperiodic (parametri), dar $e(t)$.



Notes

Alte scheme de discretizare ?

Până acum - diferenţe finite regresive de ordinul 1.
Ce ar însemna diferenţe centrate de ordinul 2?

$$i_k(t_j) = C_k \frac{\mathrm{d}u_k}{\mathrm{d}t}(t_j)$$

Discretizată

$$i_k^{(j)} = C_k \frac{u_k^{(j+1)} - u_k^{(j-1)}}{2h} \quad \Rightarrow \quad u_k^{(j+1)} = u_k^{(j-1)} + \frac{2h}{C_k} i_k^{(j)}$$

:(

Nu mai corespunde rezolvării unui circuit - nu apare $i_k^{(j+1)}$.

Notes

Alte scheme de discretizare ?

Până acum - diferenţe finite regresive de ordinul 1.
Ce ar însemna diferenţe centrate de ordinul 2?

$$i_k(t_j) = C_k \frac{\mathrm{d}u_k}{\mathrm{d}t}(t_j)$$

Discretizată

$$i_k^{(j)} = C_k \frac{u_k^{(j+1)} - u_k^{(j-1)}}{2h} \quad \Rightarrow \quad u_k^{(j+1)} = u_k^{(j-1)} + \frac{2h}{C_k} i_k^{(j)}$$

:|
Pentru alte scheme de discretizare

Notes

Introducere
MDF: Circuite discretizate
Schema de discretizare în timp
Circuite companion
Algoritmul metodei

Alte scheme de discretizare ?

Până acum - diferenţe finite regresive de ordinul 1.
Ce ar însemna diferenţe centrate de ordinul 2?

$$i_k(t_j) = C_k \frac{\mathrm{d}u_k}{\mathrm{d}t}(t_j)$$

Discretizată

$$i_k^{(j)} = C_k \frac{u_k^{(j+1)} - u_k^{(j-1)}}{2h} \quad \Rightarrow \quad u_k^{(j+1)} = u_k^{(j-1)} + \frac{2h}{C_k} i_k^{(j)}$$

:)
Pentru alte scheme de discretizare
asamblăm sistemul de stare.

Notes

Lectură obligatorie

[Ioan98] D. Ioan et al., *Metode numerice în ingineria electrică*, Ed. Matrix Rom, Bucuresti, 1998. (Capitolul 20)

Cartea se găseşte la biblioteca UPB, puteţi verifica accesând catalogul http://www.library.pub.ro/.

Notes

Notes