



Mihai Neghină

Modelarea și controlul sistemelor

Aplicații de laborator în Matlab și Simulink

Editura Universității “Lucian Blaga” din Sibiu

2018

ISBN: 978-606-12-1530-0

Copertă: Anna Enache

Toate drepturile acestei ediții sunt rezervate autorului

Cuvânt înainte

În această lucrare sunt prezentate noțiuni fundamentale de analiză, modelare și control al sistemelor fizice și electrice, însoțite de exemple și exerciții implementate parțial sau integral în *Matlab* și *Simulink*. Indrumarul conține 10 lucrări de laborator, fiecare având o scurtă prezentare teoretică a noțiunilor, exemple pentru consolidarea acestora și aplicații didactice în *Matlab* și *Simulink*.

Lucrarea este adresată în special studenților Facultății de Inginerie din cadrul Universității “*Lucian Blaga*” din Sibiu, ca suport pentru activitățile cursurilor “*Controlul digital al proceselor*” și “*Modelare, identificare și simulare a sistemelor*” însă poate fi utilă oricărei persoane interesate de principiile modelării și controlului sistemelor.

Doresc pe această cale să mulțumesc Domnului Prof. Dr. Ing. MIHU P. Ioan pentru tot sprijinul acordat în activitatea mea didactică, pentru încurajarea de a scrie acest indrumar precum și pentru observațiile privitoare la conținutul acestuia.

Sibiu 2018

Mihai Neghină

Cuprins

Matlab & Simulink Noțiuni introductive	7
Proprietăți ale sistemelor.....	21
Reprezentarea și simularea modelelor sistemelor electrice	31
Reprezentarea și simularea modelelor sistemelor mecanice.....	39
Modelarea sistemelor cu ajutorul funcției de transfer.....	45
Modelarea sistemelor cu ajutorul variabilelor de stare	53
Stabilitatea sistemelor liniare	65
Controlul sistemelor și ajustarea parametrilor controller-ului <i>PID</i>	73
Modelarea sistemelor în timp discret	87
Controller-ul <i>Deadbeat</i>	95

Matlab & Simulink

Noțiuni introductive

Obiective: însușirea unor elemente de bază din *Matlab Script* și *Simulink*.

Noțiuni teoretice

Matlab (Matrix Laboratory) este un mediu de simulare de limbaj înalt, folosit intens în cercetare și în inginerie. Aplicațiile din acest îndrumar au fost realizate în versiunea de *Matlab R2105a*, însă majoritatea funcțiilor și blocurilor *Simulink* sunt disponibile și în versiuni anterioare ale programului *Matlab*.

Introducere în *Matlab*

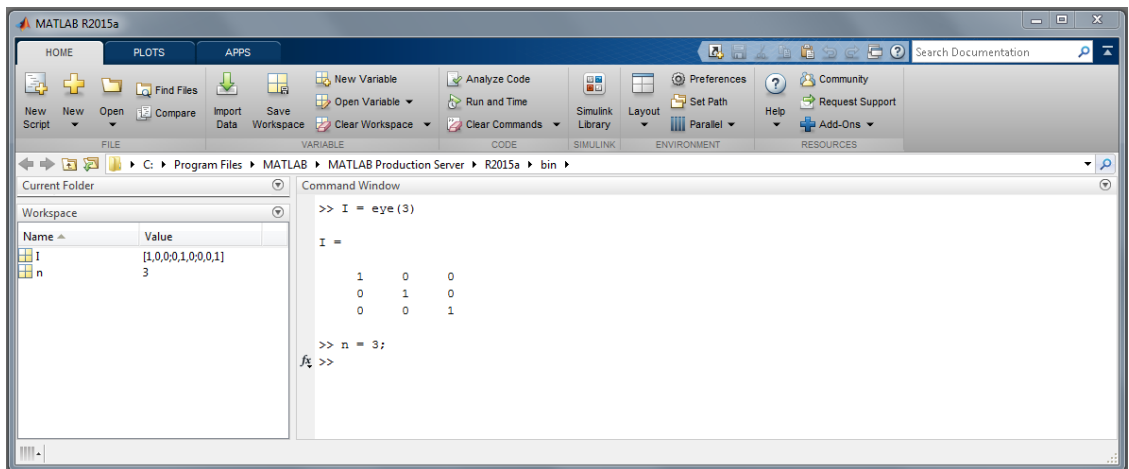


Figura 1. Ferestrele principale din *Matlab*

Cele mai importante ferestre din *Matlab* sunt:

- *Current Directory* – permite setarea directorului curent
- *Command Window* – permite executarea de instrucțiuni izolate și este fereastra în care se afișează rezultatul operațiilor acestor instrucțiuni sau a instrucțiunilor din fișierele *Script*
- *Workspace* – permite vizualizarea tuturor variabilelor din memoria curentă

Pe lângă acestea, există și fereastra *Editor*, în care se pot implementa fișiere *Matlab Script* (pentru a putea rula mai multe comenzi).

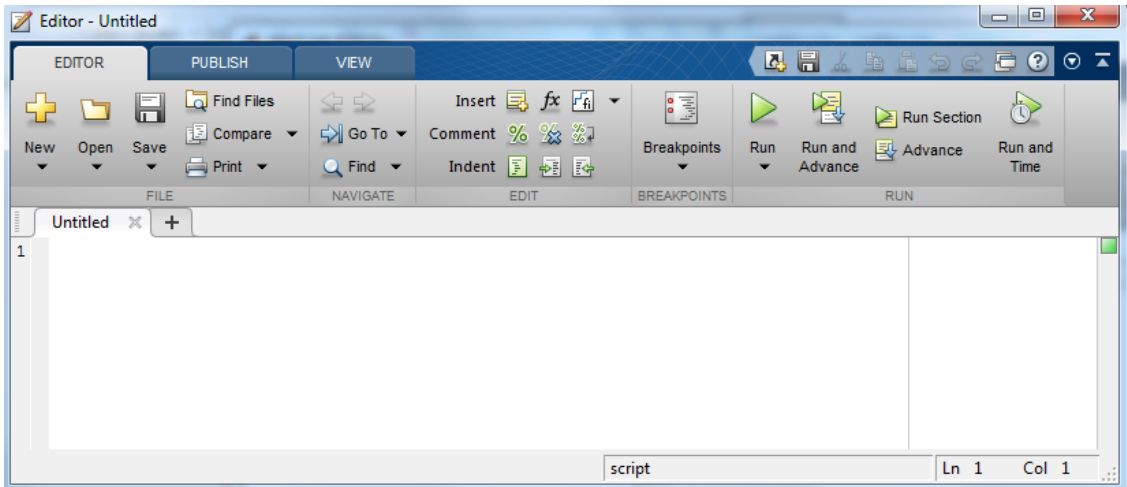


Figura 2. Editorul *Matlab*

Pentru a afla mai multe despre o funcție din *Matlab*, se pot folosi comenzile:

- **lookfor** < cuvânt cheie >

Găsește toate funcțiile care conțin cuvântul cheie în denumire sau descriere

Exemplu: `lookfor average`, caută toate funcțiile care conțin cuvântul cheie *average*

- **help** < nume funcție >

Afișează în *Command Window* descrierea funcției căutate

Exemplu: `help mean`, va afișa descrierea funcției *mean*

Dacă nu sunt declarate de utilizator, *Matlab* consideră variabilele:

- i, j ca fiind $\sqrt{-1}$
- $pi = \pi$ (3.14159 ...)

Comenzile `if`, `switch`, `while` trebuie să fie terminate cu `end`.

Funcțiile `save` și `load` pot fi folosite pentru a salva/încărca variabile în/din fișiere

*.mat

Tabel 1. Particularități ale sintaxei Matlab

Simbol	Explicație	Exemplu
%	introduce un comentariu	<code>% acesta este un comentariu</code>
;	la sfârșitul unei linii de comandă suprimă afișarea rezultatului acelei comenzi	<code>>> a = sqrt(5); >> b = sqrt(5) b = 2.2361</code>
[;]	concatenare pe verticală a două sau mai multe matrice	<code>>> A = [2 , 3; 4 ,0; 1, 2]; >> B = [1 0]; >> C = [A; B] C = 2 3 4 0 1 2 1 0</code>
[,]	concatenare pe orizontală a două sau mai multe matrice	<code>>> A = [2 3; 4 0]; >> B = [1 2 0; 1 3 2]; >> C = [A, B] C = 2 3 1 2 0 4 0 1 3 2</code>
...	permite continuarea comenzii pe linia următoare (exceptând comentariile)	<code>>> a = 1 + 2 + 3 + 4 + 5 ... + 6 +7 + 8 + 9 + 10 a = 55</code>
:	operator pentru generarea unui vector de numere echidistante	<code>>> a = 1 : 5 a = 1 2 3 4 5 >> b = 1 : 3 : 10 b = 1 4 7 10</code>
.	realizează operații element cu element	<code>>> A = [1 2; 0 1]; >> B = [1 0; 5 3]; >> C = A.*B C = 1 0 0 3 >> D = A.^2 D = 1 4 0 1</code>
' '	declară un șir de caractere (string)	<code>>> nume = 'exemplu' nume = exemplu</code>
~	NU logic (NOT)	<code>>> ~0 ans = 1</code>

Alte funcții utile în Matlab:

- **Funcții de afișare**
 - `plot, stem, bar, pie, stairs`
 - `surf, mesh, waterfall`
 - `imshow, image, imagesc`
- **Funcții generale**
 - `dir, mkdir, rmdir, who, whos, which, what, exist`
 - `find, eval`
- **Funcții de interfață cu utilizatorul**
 - `input, disp, keyboard, return, pause`
- **Funcții de generare și operații cu matrice**
 - `ones, zeros, eye`
 - `kron, repmat`
 - `length, size, reshape`
- **Funcții matematice**
 - `min, max, floor, round, ceil`
 - `sum, prod, mean, median, std, var, cov`
 - `abs, mod, rem`
 - `random, rand, pdf, unifrnd, randint, normrnd, randn`
- **Funcții pentru generarea de semnale**
 - `sin, cos, tan, cot`
 - `sinc, square, chirp`
- **Funcții pentru analiza semnalelor și sistemelor**
 - `spectrogram, freqz, impz`
 - `fft, ifft, fftshift`
 - `tf, zpk, ss, sos, tf2ss, ss2tf`
 - `pzmap, bode, nyquist`
 - `sim, lsim`

Introducere în Simulink

Simulink (Simulation + Link) este o extensie a programului *Matlab* care oferă posibilitatea de a modela, simula și analiza sisteme dinamice. Interfața grafică *Simulink* ușurează construcția și simularea (vizualizarea) modelelor sistemelor dinamice. Construcția modelelor este simplificată prin existența unor blocuri (librării) care conțin funcții ce pot fi ușor utilizate în construcția modelului (drag-and-drop) și conectarea acestora cu doar câteva click-uri.

Simulink conține o gamă largă de funcții și blocuri pentru construcția și analiza atât a sistemelor liniare cât și a celor neliniare, în domeniile timp și frecvență (Fourier, Laplace, Z).

Ca parte integrală a *Matlab*, *Simulink* poate comunica ușor cu nucleul *Matlab Script*, permițând utilizatorului să folosească caracteristicile avantajoase ale amândurora:

- funcțiile *Matlab* predefinite sau construite de utilizator pot fi apelate direct din blocuri
- variabilele și semnalele definite în *Workspace* sunt vizibile modelelor *Simulink*
- este posibilă exportarea semnalelor din *Simulink* către *Workspace*
- este posibilă ajustarea și rularea simulărilor prin comenzi din *Matlab Script*
- etc.

Accesarea librărilor de blocuri din *Simulink* se poate face fie din *Command Window* (`simulink` + Enter), fie din butonul special pentru deschiderea acestor librării.

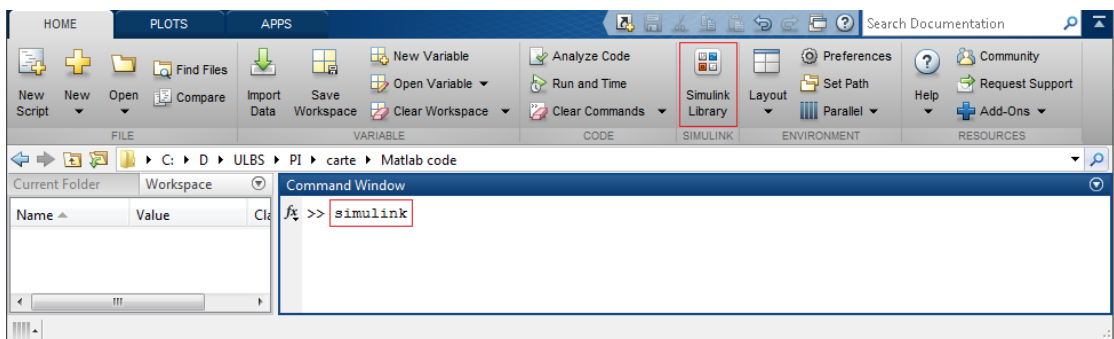


Figura 3. Accesarea *Simulink* din *Command Window*

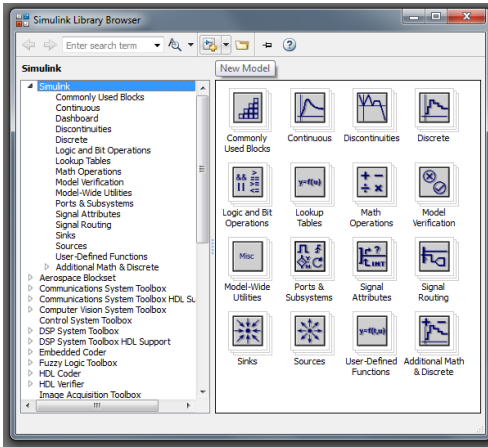


Figura 4. Librăria de blocuri *Simulink*

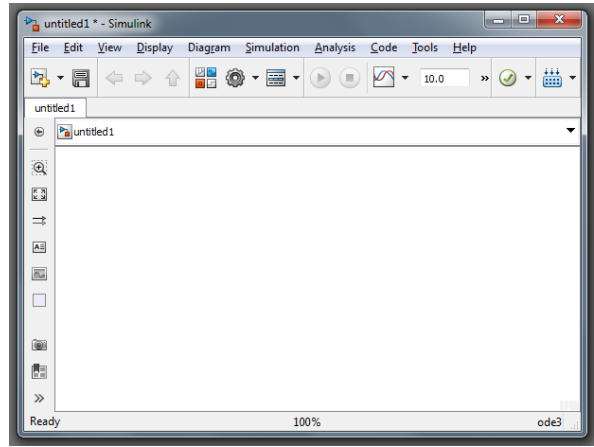
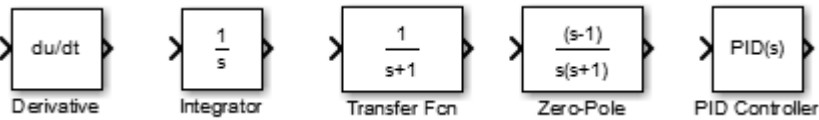


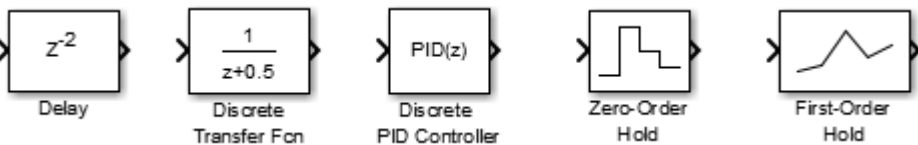
Figura 5. Editorul *Simulink*

Butonul de deschidere a librăriei de blocuri se găsește și în interfața grafică (*.mdl / *.slx) unde este asamblat modelul. Blocurile din librărie sunt grupate în **categoriile funcționale**, între care regăsim:

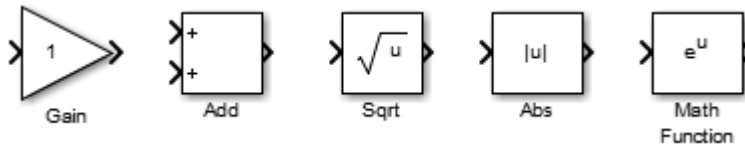
- *Commonly Used Blocks*: așa cum sugerează numele, conține blocuri folosite de curând de utilizator. Dacă un bloc nu mai este folosit un timp, acesta poate dispărea din *Commonly Used Blocks*, de aceea nu este recomandată folosirea acestui grup.
- *Continuous*: conține blocuri utile modelării sistemelor în timp continuu, fie pornind de la ecuațiile diferențiale (blocuri de integrare/derivare), fie după prelucrarea acestor ecuații (modele cu variabile de stare, funcția de transfer Laplace). Cele mai folosite blocuri din acest grup sunt:



- *Discrete*: conține blocuri specifice modelării în timp discret și cu ajutorul transformatei Z. Cele mai folosite blocuri din acest grup sunt:



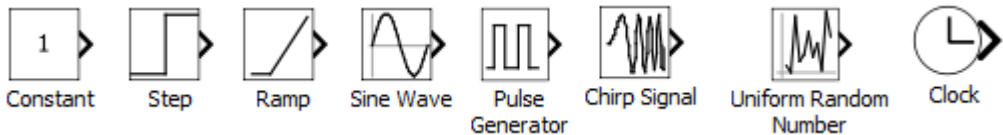
- *Math Operations*: conține operații matematice de bază (adunare, modul, înmulțire cu o constantă - *Gain*). Cele mai folosite blocuri din acest grup sunt:



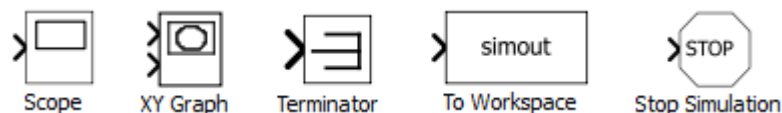
- *Signal Routing*: conține blocuri de agregare sau decuplare a semnalelor (*multiplexor, demultiplexor, selector, switch*). Cele mai folosite blocuri din acest grup sunt:



- *Source*: conține blocuri de generare de semnal (generatoare de semnal sinusoidal, triunghiular sau dreptunghiular, semnale treaptă unitate, rampe, semnale constante sau aleatoare). Cele mai folosite blocuri din acest grup sunt:



- *Sinks*: conține blocuri pentru vizualizarea grafică a semnalelor (*Scope*) și operații asupra semnalelor care nu necesită întoarcerea acestora în model (salvare în *Workspace*, oprirea simulării). Cele mai folosite blocuri din acest grup sunt:



Interfața grafică (*.mdl / *.slx) unde este asamblat modelul mai conține următoarele butoane importante:



New model, accesibil și din *File* → *New*, pentru deschiderea unei interfețe noi



Save, accesibil și din *File* → *Save*, pentru salvarea modelului curent



Run, accesibil și din *Simulation* → *Run*, pentru rularea simulării



Model Configuration Parameters, pentru configurarea parametrilor de simulare, accesibil și din *Simulation* → *Model Configuration Parameters*

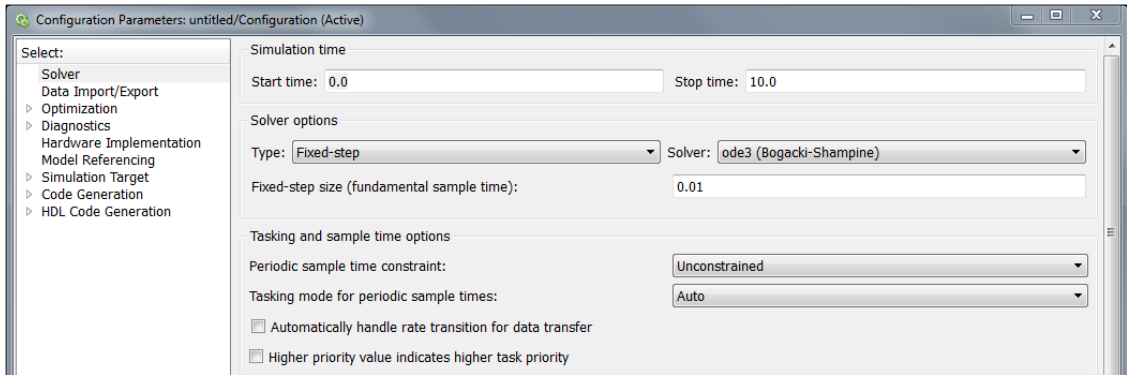


Figura 6. Configurarea parametrilor modelului *Simulink*

Aici se pot ajusta mai mulți parametri de simulare, inclusiv parametri pentru generarea codului C / VHDL corespunzător, însă pentru simulări fără implementare industrial (*deployment*), sunt suficiente setările:

- timpul de început și de terminare al simulării, în cazul în care nu există condiții suplimentare de oprire a simulării când se ajunge într-o anumită stare.
- tipul de rezolvare numerică a ecuațiilor diferențiale (*Solver type: fixed-step / Solver: ode3*) și pasul de simulare (pentru simulare cu pas fix).

Pentru realizarea modelului, trebuie ținut cont că:

- firele (legăturile între blocuri) sunt **semnale**
- blocurile *Simulink* reprezintă **operații** sau **transformări ale semnalelor**
- adăugarea denumirii semnalelor deasupra firelor sau a comentariilor nu afectează simularea

Tabel 2. Scurtături utile în construirea modelelor cu ajutorul *Simulink*

Scurtătură	Descriere
Drag-and-drop	Adăugarea unui bloc din librărie prin tragerea acestuia în model
Dublu-click pe un bloc	Deschiderea ferestrei cu proprietățile blocului
Dublu-click pe etichetă	Redenumirea unui bloc
Dublu-click oriunde pe fundalul modelului	Adăugarea unui comentariu (text), de exemplu adăugarea numelor semnalelor deasupra firelor care le conțin
Ctrl+X, Ctrl+C, Ctrl+V	(Cut, Copy, Paste) Copierea unui bloc sau a unui grup de blocuri în același model sau într-un fișier *.mdl / *.slx nou
Ctrl+R, Ctrl+Shift+R	Rotirea unui bloc în sens orar sau trigonometric
Ctrl+I	(Flip) Oglindirea unui bloc față de axa verticală
Selectare bloc 1 urmat de Ctrl + selectare bloc 2	Generarea unei legături între cele două blocuri, dinspre blocul 1 înspre blocul 2, ca alternativă la trasarea legăturii
Ctrl + Plus, Ctrl + Minus	Zoom-in, Zoom-out model

Desfășurarea lucrării

În cadrul acestei lucrări se va realiza și simula modelul grafic al unui sistem, fără a detalia deocamdată cum a fost obținut modelul sistemului (ecuația diferențială).

Aplicație 1. Fie un pendul simplu caracterizat de ecuația:

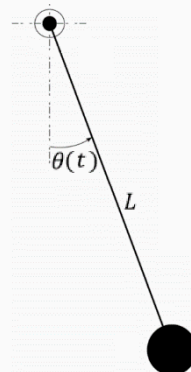
$$\frac{d^2\theta(t)}{dt^2} + \frac{g}{L} \cdot \sin(\theta(t)) = 0 \quad (1)$$

unde:

- $\theta(t)$ este unghiul față de verticală,
- L este lungimea pendulului
- g este accelerația gravitațională;

Condițiile inițiale sunt astfel încât pendulul să pornească de la $\pi/4$.

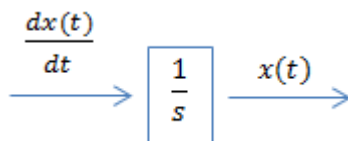
Să se simuleze variația unghiului de pendulare.



Trecerea de la modelul matematic - ecuația diferențială (1) – la modelul grafic se poate face prin analiza diverșilor termeni ai ecuației. Pentru început se poate analiza efectul blocului integrator asupra derivatei unei funcții.

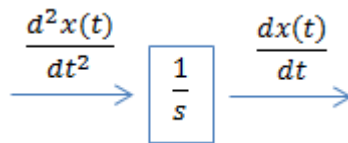
$$\int \frac{dx(t)}{dt} dt = \int dx = x(t) \quad (2)$$

Dacă la intrarea unui integrator se aduce derivata unui semnal $\frac{dx(t)}{dt}$, la ieșirea integratorului va rezulta chiar semnalul $x(t)$:



Dacă la intrarea integratorului se aduce derivata de ordin II a semnalului, $\frac{d^2x(t)}{dt^2}$, la ieșirea acestuia va rezulta derivata de ordin I, $\frac{dx(t)}{dt}$:

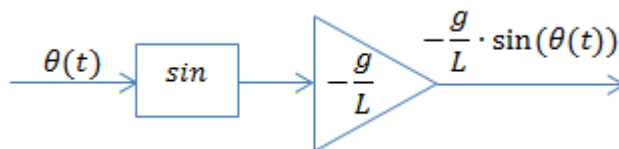
$$\int \frac{d^2x(t)}{dt^2} dt = \frac{dx(t)}{dt} \quad (3)$$



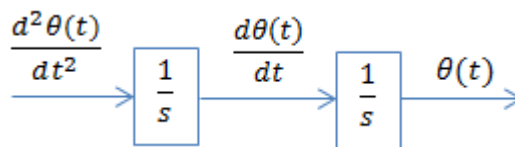
Acum se poate construi schema de simulare a modelului, folosind blocuri funcționale *Simulink*. Pentru aceasta, ecuația diferențială este rescrisă astfel:

$$\frac{d^2\theta(t)}{dt^2} = -\frac{g}{L} \cdot \sin(\theta(t)) \quad (4)$$

Folosind blocul de *multiplicare* și blocul de *sinus*, se poate contrui termenul drept al ecuației (4):



Folosind două blocuri de integrare, se poate contrui termenul stang al ecuației (4):



Pe baza egalității (4) se pot uni cele două grafice unul în continuarea celuilalt. Deoarece atât la intrarea lanțului de blocuri cât și la ieșire este aceleași semnal, și anume $\theta(t)$, se obține schema de simulare din figura 7, în care $\theta(t)$, mărimea urmărită, este conectată la un osciloscop pentru vizualizare

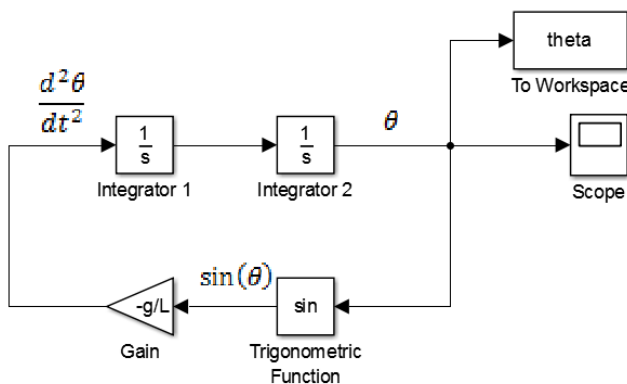


Figura 7. Modelarea grafică și simularea unui pendul în *Simulink*

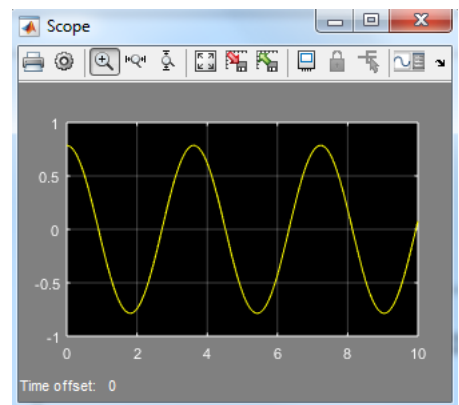
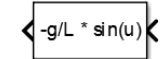



Figura 8. Variația unghiului θ în timp (*Simulink Scope*)

Alternativ, bucla de reacție putea fi construită și prin:

- definirea unei funcții de către utilizator:  User-Defined Function
- fie scriind formula direct în blocul *Fcn*:  MATLAB Function
- fie scriind formula într-un fișier *Matlab Script Function*:

```
function [y] = FctPendul(theta)
global g L;
y = -g/L * sin(theta);
```

În cazul unei funcții implementate într-un fișier script, constantele (g, L) trebuie declarate globale pentru a fi recunoscute de funcție.

Rularea simulării se poate face și direct din *Matlab Command Window* sau dintr-un fișier *Script*, prin comanda:

```
>> sim('ModelSimulink', TimpOprire);
```

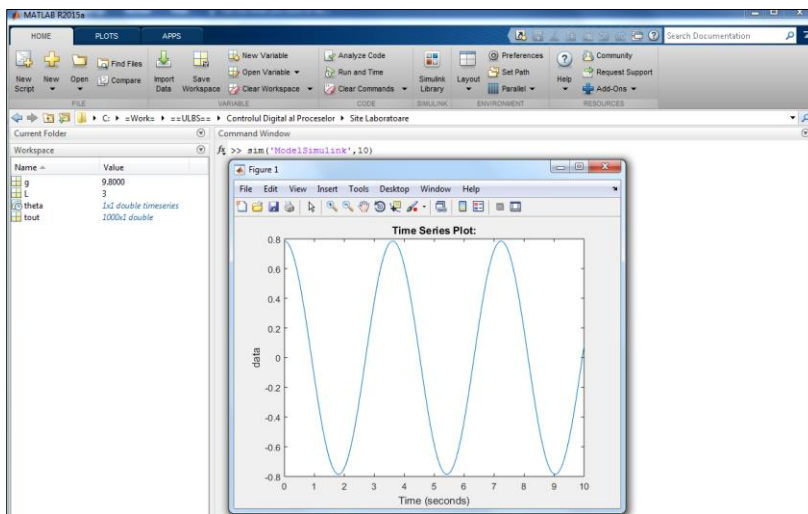
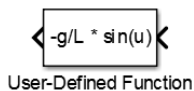


Figura 9. Variația unghiului θ în timp (Matlab plot)

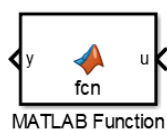
Probleme și aplicații

1. Să se realizeze modelele pendulului atunci când bucla de reacție este construită prin:

a) definirea unei funcții de către utilizator



b) scrierea formulei direct în blocul *Fcn*



c) scrierea formulei într-un fișier *Matlab Script Function*

Să se compare graficele (*Scope*) obținute cu cele 3 modele.

2. Să se modeleze în *Simulink* un amplificator ideal cu factor de amplificare k , care aplică funcția matematică:

$$U_{out} = k \cdot U_{in}$$

Să se ruleze simularea pentru următoarele semnale de intrare:

- semnal constant cu valoarea 1
- semnal rampă cu panta 1
- semnal sinusoidal
- semnal dreptunghiular
- semnal aleator

Să se afișeze pe același grafic (*Scope*) semnalul de intrare și cel de ieșire.

Notă: blocul Mux ar putea fi util pentru multiplexarea semnalelor înainte ca acestea să intre în Scope.

3. Redresorul monoalternanță ideal poate fi privit ca un diport cu funcționare de comutator comandat de semnul semnalului de intrare. Dacă semnul este pozitiv, comutatorul este închis și semnalul de intrare se regăsește la ieșire. Dacă este negativ, comutatorul este deschis și ieșirea devine 0.

$$U_{out} = \begin{cases} U_{in}, & \text{daca } \text{sgn}(U_{in}) \geq 0 \\ 0, & \text{daca } \text{sgn}(U_{in}) < 0 \end{cases}$$

Redresorul bialternanță ideal poate fi considerat un diport care aplică funcția matematică “modul”.

$$U_{out} = |U_{in}|$$

- a) să se modeleze în *Simulink* (în fișiere separate) un redresor monoalternanță și un redresor bialternanță, ambele având câte un semnal sinusoidal la intrare.
- b) să se genereze un fișier *.m care să cheme modelele ambelor redresoare și să afișeze ieșirile grafic (*plot*).

Proprietăți ale sistemelor

Obiective: Verificarea unor proprietăți de bază ale sistemelor (liniaritate, cauzalitate, etc.), atât în cazul în care este cunoscută relația dintre semnalele de intrare și cele de ieșire, cât și în cazul în care această relație nu este cunoscută.

Noțiuni teoretice

În funcție de relația dintre semnalele de intrare și cele de ieșire, sistemele prezintă șase proprietăți utile pentru analiza sau proiectarea lor:

- *Stabilitate:* proprietate esențială pentru a nu exista riscul ca sistemul să genereze semnale crescătoare, ce tind la infinit (această proprietate va fi discutată pe larg în altă lucrare de laborator).
- *Cauzalitate:* efectul urmează cauzei, niciodată invers. Consecința: nu vom opera niciodată cu mărimi viitoare ci doar cu cele trecute și cu cele prezente.
- *Invarianța în timp:* o cauză produce același efect, indiferent de momentul la care cauza este adusă la intrarea sistemului.
- *Liniaritate:* proprietate care permite aplicarea principiului superpoziției iar aparatul matematic pentru rezolvare este accesibil.
- *Reversibilitate:* proprietatea ca sistemul să poată fi inversat. Se poate construi un alt sistem cu o funcție de transfer inversă (pentru reconstituirea intrării din valorile ieșirii).
- *Memorie:* prin integrare, într-un sistem cauzal, unei valori imediat anterioare îi este însumată valoarea prezentă. Deci orice integrator are nevoie de o celulă de memorie care să ”țină minte” valoarea anterioară.

Cauzalitate

Un sistem este cauzal dacă pentru orice două semnale de intrare $x_1(t)$ și $x_2(t)$ identice până la momentul t_0 , semnalele de ieșire $y_1(t)$ și $y_2(t)$ sunt identice până la momentul t_0 .

$$x_1(t) = x_2(t), \forall t \leq t_0 \Rightarrow y_1(t) = y_2(t), \forall t \leq t_0 \quad (1)$$

Cu alte cuvinte, un sistem este cauzal dacă și numai dacă la orice moment de timp t_0 semnalul de ieșire $y(t_0)$ depinde doar de valoarea curentă și/sau valorile anterioare ale semnalului de intrare $x(t)$.

O definiție echivalentă poate fi construită și pentru sisteme discrete:

$$x_1(n) = x_2(n), \forall n \leq n_0 \Rightarrow y_1(n) = y_2(n), \forall n \leq n_0 \quad (2)$$

Sisteme necauzale pe de altă parte au nevoie de valori viitoare ale semnalelor de intrare (adică au nevoie de glob de cristal să prevadă viitorul semnalului de intrare ☺). Cauzalitatea este o condiție necesară (nu și suficientă) ca un sistem să poată exista fizic, să poată fi construit și să poată fi folosit în timp real.

Invarianța în timp

Un sistem este invariant în timp dacă și numai dacă o întârziere cu t_0 a semnalului de intrare produce o ieșire identică dar întârziată cu același t_0 .

$$x(t) \rightarrow y(t) \Rightarrow x(t - t_0) \rightarrow y(t - t_0) \quad (3)$$

O definiție echivalentă poate fi construită și pentru sisteme discrete:

$$x(n) \rightarrow y(n) \Rightarrow x(n - n_0) \rightarrow y(n - n_0) \quad (4)$$

Invarianța în timp asigură faptul că un sistem funcțional se va comporta la fel indiferent când este testat; dacă sistemul este testat în două zile consecutive, semnalele de ieșire generate vor fi identice, dar cel de-al doilea semnal de ieșire va fi generat cu o zi mai târziu decât primul deoarece și semnalele de intrare au fost decalate cu o zi.

Pentru a verifica invarianța în timp pornind de la relațiile sistemului, trebuie verificat că operațiile de trecere prin sistem (S) și de întârziere (D) produc același rezultat.

Pentru sisteme invariante în timp:

$$\left. \begin{array}{l} x_1(t) \xrightarrow{S} y_1(t) \xrightarrow{D} y_1(t - t_0) \\ x_1(t) \xrightarrow{D} x_1(t - t_0) \xrightarrow{S} y_2(t) \end{array} \right\} \Rightarrow y_2(t) = y_1(t - t_0) \quad (5)$$

Liniaritate

Un sistem este liniar dacă și numai dacă, pentru orice două semnale de intrare $x_1(t)$ și $x_2(t)$ pentru care ieșirile sistemului sunt $y_1(t)$ și $y_2(t)$, răspunsul sistemului la orice combinație liniară a intrărilor este aceeași combinație liniară a ieșirilor.

$$\begin{aligned} x_1(t) \rightarrow y_1(t), \quad x_2(t) \rightarrow y_2(t) &\Rightarrow \\ \alpha \cdot x_1(t) + \beta \cdot x_2(t) \rightarrow \alpha \cdot y_1(t) + \beta \cdot y_2(t) &\end{aligned} \quad (6)$$

O definiție echivalentă poate fi construită și pentru sisteme discrete:

$$x_1(n) \rightarrow y_1(n), \quad x_2(n) \rightarrow y_2(n) \Rightarrow \alpha \cdot x_1(n) + \beta \cdot x_2(n) \rightarrow \alpha \cdot y_1(n) + \beta \cdot y_2(n) \quad (7)$$

De remarcat faptul că este suficientă verificarea proprietății de liniaritate pentru două semnale, pentru ca aceasta să fie validă pentru suma oricâtor semnale. Sistemele liniare permit așadar folosirea superpoziției, răspunsul generat de către mai mulți stimuli aplicați simultan este egal cu suma răspunsurilor generate de fiecare stimul în parte. De asemenea, dacă semnalul de intrare se poate descompune într-o sumă de semnale mai simple, răspunsul sistemului va fi aceeași combinație a răspunsurilor sistemului la semnalele mai simple, proprietate foarte utilă în analiza sistemelor.

Reversibilitate

Un sistem este reversibil dacă și numai dacă intrări diferite produc ieșiri diferite.

$$x_1(t) \neq x_2(t) \Rightarrow y_1(t) \neq y_2(t) \quad (8)$$

O definiție echivalentă poate fi construită și pentru sisteme discrete:

$$x_1(n) \neq x_2(n) \Rightarrow y_1(n) \neq y_2(n) \quad (9)$$

Cu alte cuvinte, reversibilitatea este proprietatea să existe (cel puțin teoretic) un sistem invers, care să poată reconstitui semnalul de intrare pornind de la semnalul de ieșire, indiferent dacă acesta este sau nu cauzal, liniar sau stabil.

Memorie

Un sistem este **static (fără memorie)** dacă, la orice moment de timp, valoarea semnalului de ieșire depinde doar de valoarea semnalului de intrare la acel moment de timp și nu depinde de valori anterioare sau ulterioare ale semnalului de intrare. Un astfel de sistem este fără memorie, deoarece trebuie doar să proceseze valoarea curentă a semnalului de intrare, fără a avea nevoie să rețină alte valori ale acestuia.

$$x_1(t_0) = x_2(t_0) \Rightarrow y_1(t_0) = y_2(t_0) \quad (10)$$

O definiție echivalentă poate fi construită și pentru sisteme discrete:

$$x_1(n_0) = x_2(n_0) \Rightarrow y_1(n_0) = y_2(n_0) \quad (11)$$

Toate sistemele statice sunt automat cauzale.

Un sistem este **dinamic** dacă valorile semnalului de ieșire depind de valori din trecut sau viitor ale semnalului de intrare. Sistemele dinamice au nevoie de memorie pentru a stoca valori trecute sau pentru a putea prezice valori viitoare ale semnalului.

Desfășurarea lucrării

Atunci când se cunosc formulele care leagă semnalul de intrare de semnalul de ieșire, se poate încerca demonstrarea relațiilor care definesc proprietățile (sau construirea contra-exemplilor pentru proprietățile care nu sunt îndeplinite).

Aplicație 1. Fie sistemul definit prin relația:

$$x, y: \mathbb{R} \rightarrow \mathbb{R}, \quad y(t) = 100 \cdot x^3(t)$$

Să se determine dacă sistemul este cauzal, invariant în timp, liniar, reversibil și cu memorie.

Cauzalitate, memorie: semnalul de ieșire $y(t)$ depinde la orice moment de timp doar de valoarea semnalului de intrare la acel moment de timp, de aceea sistemul este fără memorie și implicit cauzal.

Invarianța în timp:

$$x_1(t) \xrightarrow{S} y_1(t) = 100 \cdot x_1^3(t) \xrightarrow{D} y_1(t - t_0) = 100 \cdot x_1^3(t - t_0)$$

$$x_1(t) \xrightarrow{D} x_2(t) = x_1(t - t_0) \xrightarrow{S} y_2(t) = 100 \cdot x_2^3(t) = 100 \cdot x_1^3(t - t_0)$$

Deoarece $y_2(t) = y_1(t - t_0) = 100 \cdot x_1^3(t - t_0)$, sistemul este invariant în timp

Liniaritate:

$$x_1(t) \rightarrow y_1(t) = 100 \cdot x_1^3(t)$$

$$x_2(t) \rightarrow y_2(t) = 100 \cdot x_2^3(t)$$

$$x_3(t) = \alpha \cdot x_1(t) + \beta \cdot x_2(t) \rightarrow y_3(t) = 100 \cdot (\alpha \cdot x_1(t) + \beta \cdot x_2(t))^3$$

Dar combinația liniară a ieșirilor este diferită de răspunsul sistemului la combinația liniară a intrărilor.

$$\begin{aligned} \alpha \cdot y_1(t) + \beta \cdot y_2(t) &= 100 \cdot x_1^3(t) + 100 \cdot x_2^3(t) \neq \\ &\neq 100 \cdot (\alpha \cdot x_1(t) + \beta \cdot x_2(t))^3 = y_3(t) \end{aligned}$$

Așadar sistemul nu este liniar.

Reversibilitate: Sistemul este reversibil deoarece pentru oricare două semnale $x_1(t) \neq x_2(t)$ răspunsurile sistemului sunt diferite:

$$y_1(t) = 100 \cdot x_1^3(t) \neq 100 \cdot x_2^3(t) = y_2(t)$$

Fiind o funcție simplă, se poate găsi ușor sistemul invers (care reface semnalul $x(t)$ din semnalul $y(t)$):

$$x(t) = \frac{1}{100} \sqrt[3]{y(t)}$$

Aplicație 2. Fie sistemul definit prin relația:

$$x, y: \mathbb{R}^+ \rightarrow \mathbb{R}, \quad y(t) = \int_0^t x(\tau) d\tau$$

Să se determine dacă sistemul este cauzal, invariant în timp, liniar, reversibil și cu memorie.

Cauzalitate, memorie: semnalul de ieșire $y(t)$ depinde la orice moment de timp doar de valorile semnalului de intrare de până la acel moment de timp, de aceea sistemul este cauzal și cu memorie.

Invarianța în timp:

$$x_1(t) \xrightarrow{S} y_1(t) = \int_0^t x_1(\tau) d\tau \xrightarrow{D} y_1(t - t_0) = \int_0^{t-t_0} x_1(\tau) d\tau$$

$$x_1(t) \xrightarrow{D} x_2(t) = x_1(t - t_0) \xrightarrow{S} y_2(t) = \int_0^t x_1(\tau - t_0) d\tau \xrightarrow{u=\tau-t_0}$$

$$y_2(t) = \int_{-t_0}^{t-t_0} x(u) du \xrightarrow{x,y:\mathbb{R}^+\rightarrow\mathbb{R}} y_2(t) = \int_0^{t-t_0} x(u) du$$

Deoarece $y_2(t) = y_1(t - t_0)$, sistemul este invariant în timp.

Liniaritate:

$$x_1(t) \rightarrow y_1(t) = \int_0^t x_1(\tau) d\tau$$

$$x_2(t) \rightarrow y_2(t) = \int_0^t x_2(\tau) d\tau$$

$$\begin{aligned} x_3(t) = \alpha \cdot x_1(t) + \beta \cdot x_2(t) &\rightarrow y_3(t) = \int_0^t (\alpha \cdot x_1(\tau) + \beta \cdot x_2(\tau)) d\tau = \\ &= \alpha \cdot \int_0^t (x_1(\tau)) d\tau + \beta \cdot \int_0^t (x_2(\tau)) d\tau = \alpha \cdot y_1(t) + \beta \cdot y_2(t) \end{aligned}$$

Deci sistemul este liniar.

Reversibilitate: Sistemul este reversibil deoarece pentru oricare două semnale $x_1(t) \neq x_2(t)$, diferența semnalelor de ieșire nu este semnalul $zero(t) = 0, \forall t$ în domeniul de definiție:

$$y_1(t) - y_2(t) = \int_0^t (x_1(\tau)) d\tau - \int_0^t (x_2(\tau)) d\tau = \int_0^t (x_1(\tau) - x_2(\tau)) d\tau \neq zero(t)$$

Aplicație 3. Fie sistemul definit prin relația:

$$x, y: \mathbb{R} \rightarrow \mathbb{R}, \quad y(t) = \frac{dx(t)}{dt}$$

Să se determine dacă sistemul este cauzal, invariant în timp, liniar, reversibil și cu memorie.

Cauzalitate, memorie: dacă semnalul de ieșire la orice moment de timp τ este definit ca derivata la stânga:

$$y(\tau) = \frac{dx(t)}{dt} = \lim_{\substack{t \rightarrow \tau \\ t < \tau}} \frac{x(t) - x(\tau)}{t - \tau}$$

atunci valorile semnalului de ieșire depind de valorile semnalului de intrare de până la acel moment de timp, de aceea sistemul este cauzal și cu memorie.

Dacă sistemul ar fi fost definit ca derivata la dreapta:

$$y(\tau) = \frac{dx(t)}{dt} = \lim_{\substack{t \rightarrow \tau \\ t > \tau}} \frac{x(t) - x(\tau)}{t - \tau}$$

atunci valorile semnalului de ieșire ar depinde de valorile semnalului de intrare viitoare aceluși moment de timp, de aceea sistemul nu ar fi cauzal, dar ar rămâne cu memorie.

Invarianța în timp:

$$x_1(t) \xrightarrow{S} y_1(t) = \frac{dx(t)}{dt} \xrightarrow{D} y_1(t - t_0) = \frac{dx(t - t_0)}{dt}$$

$$x_1(t) \xrightarrow{D} x_2(t) = x_1(t - t_0) \xrightarrow{S} y_2(t) = \frac{dx(t - t_0)}{dt}$$

Deoarece $y_2(t) = y_1(t - t_0)$, sistemul este invariant în timp.

Liniaritate:

$$x_1(t) \rightarrow y_1(t) = \frac{dx_1(t)}{dt}$$

$$x_2(t) \rightarrow y_2(t) = \frac{dx_2(t)}{dt}$$

$$\begin{aligned} x_3(t) = \alpha \cdot x_1(t) + \beta \cdot x_2(t) \rightarrow y_3(t) &= \frac{d(\alpha \cdot x_1(t) + \beta \cdot x_2(t))}{dt} = \\ &= \frac{d(\alpha \cdot x_1(t))}{dt} + \frac{d(\beta \cdot x_2(t))}{dt} = \alpha \cdot y_1(t) + \beta \cdot y_2(t) \end{aligned}$$

Deci sistemul este liniar.

Reversibilitate: Sistemul nu este reversibil deoarece există două semnale $x_1(t) \neq x_2(t)$ pentru care ieșirile sunt identice. De exemplu:

$$\left. \begin{aligned} x_1(t) &= 2, \forall t \\ x_2(t) &= 3, \forall t \end{aligned} \right\} \Rightarrow x_1(t) \neq x_2(t)$$

Dar $y_1(t) = y_2(t) = 0, \forall t$. Pentru a demonstra că un sistem nu este reversibil, este suficient un exemplu de două semnale distincte care produc aceeași ieșire.

Probleme și aplicații

1. Fie sistemele definite prin relațiile:

a) $x, y: R \rightarrow R, \quad y(t) = x(t) + 1$

b) $x, y: R \rightarrow R, \quad y(t) = x(t + 1)$

c) $x, y: R \rightarrow R, \quad y(t) = 3 \cdot x^2(t)$

d) $x: R \rightarrow R^+, \quad y: R \rightarrow R, \quad y(t) = \ln(x(t))$

Să se determine dacă sistemele sunt cauzale, invariante în timp, liniare, reversibile, cu memorie.

2. Fie controller-ul *PID* definit prin relația:

$$x, y: R^+ \rightarrow R, \quad y(t) = k_P \cdot x(t) + k_I \cdot \int_0^t x(\tau) d\tau + k_D \cdot \frac{dx(t)}{dt}$$

Să se determine dacă sistemul este cauzal, invariant în timp, liniar, reversibil și cu memorie.

3. Fie sistemele discrete definite prin relațiile:

a) $x, y: N \rightarrow R, \quad y(n) = x(n) - 1$

b) $x, y: N \rightarrow R, \quad y(n) = x(n - 1)$

c) $x, y: N \rightarrow R, \quad y(n) = 5 \cdot x^4(n)$

d) $x, y: N \rightarrow R, \quad y(n) = -|x(n)|$

Să se determine dacă sistemele sunt cauzale, invariante în timp, liniare, reversibile, cu memorie.

4. Fie sistemele discrete definite în *Matlab* ca funcții precompilate *bbox1.p, ..., bbox7.p*, care primesc la intrare un vector de valori x și generează la ieșire un vector de aceeași dimensiune y . Să se verifice cauzalitatea, invarianța în timp, liniaritatea și memoria.

Notă: scopul este de a determina proprietățile sistemelor, nu de a încerca ghicirea funcției. Pentru fiecare proprietate, este nevoie să se genereze și alte semnale de intrare x , care să verifice acea proprietate și să se interpreteze ieșirile obținute y .

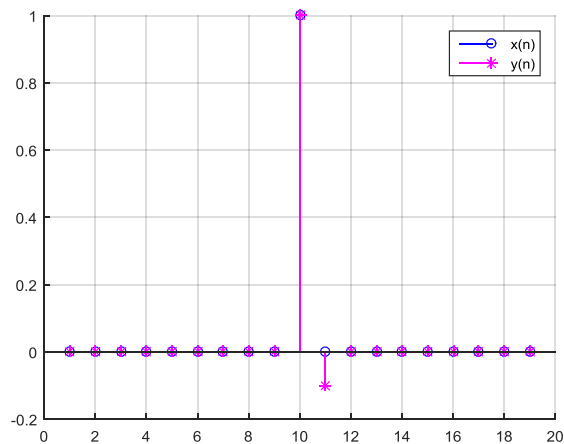
Pentru sisteme black-box (pentru care nu avem acces la funcția matematică sau componentele fizice care compun sistemul), nu se poate determina cauzalitatea, invarianța în timp, liniaritatea sau faptul că este un sistem fără memorie, ci doar negativele proprietăților, prin găsirea de contra-exemple.

De exemplu, secvența x_1 de mai jos trecută prin `bbox1` produce graficul din figură. Există informații utile în acest grafic referitor la vreuna dintre proprietăți?

```
x1 = [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0];
```

```
y1 = bbox1(x1);
```

```
figure, hold on, stem(x1, 'bo'), stem(y1, 'm*'), hold off
```



Reprezentarea și simularea modelelor sistemelor electrice

Obiective: Reprezentarea și simularea modelului unui circuit electric cu blocuri *Simulink*, pornind de la ecuațiile diferențiale care descriu sistemul.

Noțiuni teoretice

Modelul unui sistem (proces) este descrierea în limbaj matematic a evoluției în timp a aceluși sistem, adică sistemul de ecuații diferențiale rezultat prin aplicarea legilor fizicii care guvernează funcționarea aceluși sistem.

Mediul *Simulink* permite reprezentarea sub formă grafică (blocuri funcționale) a termenilor din ecuațiile diferențiale care descriu sistemul, și apoi permite simularea sistemului. Pașii sunt următorii:

1. Scrierea ecuațiilor diferențiale prin aplicarea și particularizarea legilor fizicii care guvernează evoluția sistemului.
2. Când ecuațiile scrise se constituie ca un sistem compatibil de ecuații diferențiale, **acest sistem de ecuații diferențiale va reprezenta modelul sistemului**. Deoarece majoritatea sistemelor de ecuații diferențiale nu pot fi rezolvate folosind metodele matematice obișnuite, singurul mod de rezolvare este transformarea (aproximarea) sistemului de ecuații diferențiale într-un sistem de ecuații cu diferențe finite, folosind pentru aceasta semnale numerice în locul semnalelor analogice originale. Acest sistem poate fi rezolvat prin metode numerice, soluția obținută fiind foarte apropiată de soluția analogică.
3. Realizarea modelului grafic al sistemului, de regulă în mediul *Simulink*.

Se pornește de la **model** (sistemul de ecuații diferențiale) și se obține **modelul grafic** (cu blocuri funcționale). Prin realizarea **modelului grafic** în *Simulink* se mai parcurge o

etapă importantă și anume transformarea *modelului analogic* (reprezentat de un sistem de ecuații diferențiale în timp continuu, ușor de recunoscut datorită variabilelor de tip t , dt , x , dx , etc) într-un *model numeric* caracterizat printr-un sistem de ecuații cu diferențe finite.

4. Simularea modelului grafic se face simplu prin apăsarea butonului RUN.

Simularea înseamnă de fapt rezolvarea numerică a sistemului original de ecuații diferențiale, transpus de *Simulink* într-un sistem de ecuații cu diferențe finite, prin metode numerice. Soluția poate fi vizualizată grafic și astfel poate fi înțeleasă evoluția în timp a sistemului.

Folosirea corectă a terminologiei aferente noțiunii de *model*, sintetizată în *Tabelul 1* de mai jos, va consolida cunoașterea teoretică a aspectelor legate de modelarea sistemelor.

Tabel 1. Terminologia aferentă denumirii de *model*

Denumire	<i>Model</i>	<i>Model grafic</i>
		<i>Model numeric</i>
Întâlnit și folosit în	Teorie: fizică, teoria sistemelor, etc	<i>Simulink, Matlab</i>
Definiție	Sistem de ecuații diferențiale rezultat din aplicarea legilor fizicii	Sistem de ecuații cu diferențe finite, rezultat din ecuațiile diferențiale
Tip	analogic	numeric
Semnale	Analogice Timp continuu (t , dt , x , dx , ...)	Numerice Timp discret (șir de numere)
Rezolvare	”cu creionul”	”cu calculatorul” folosind algoritmi numerici: ODE3, etc

Pentru realizarea modelului grafic în *Simulink*, trebuie ținut cont că:

- blocurile *Simulink* reprezintă **operații**
- firele (legăturile între blocuri) sunt **semnale**

Desfășurarea lucrării

Aplicație 1. Fie circuitul RC din *Figura 1*.

Să se scrie ecuațiile diferențiale care descriu circuitul RC serie.



Figura 1. (a) Reprezentarea circuitului electric; (b) Modelul abstract corespunzător

$$\begin{cases} v_{in}(t) = R \cdot i(t) + v_{out}(t) & (1) \\ i(t) = C \cdot \frac{dv_{out}(t)}{dt} & (2) \end{cases}$$

- $i(t)$ este curentul în circuit, considerat semnal intern al sistemului.
- $v_{in}(t)$ este tensiunea de alimentare, considerată semnal de intrare în sistem.
- $v_{out}(t)$ este tensiunea pe condensator, considerată semnal de ieșire din sistem.

Aplicație 2. Să se modeleze sistemul din *Figura 1* în *Simulink*. Să se afișeze pe grafic (Scope) tensiunea pe condensator și cea pe rezistor. $R = 4k\Omega$ și $C = 100\mu F$ sunt valori constante pe parcursul simulării.

Dacă variabilele R , C există în *Workspace* (de exemplu definite în *Command Window* sau prin rularea unui fișier *.m), modelul *Simulink* poate lucra cu ele. Alternativ, variabilele pot fi definite și în proprietățile modelului (*File* → *ModelProperties*), prin *InitFcn* callback. Modelul trebuie salvat după fiecare modificare.

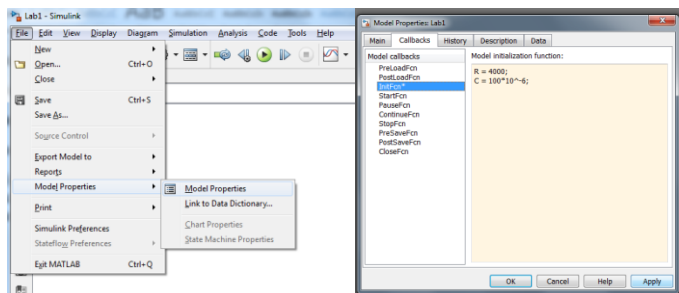


Figura 2. Definirea variabilelor în proprietățile modelului *Simulink*

Realizarea modelului pornind de la ecuațiile diferențiale începe prin adăugarea blocului *Scope* pentru vizualizarea semnalului de ieșire.

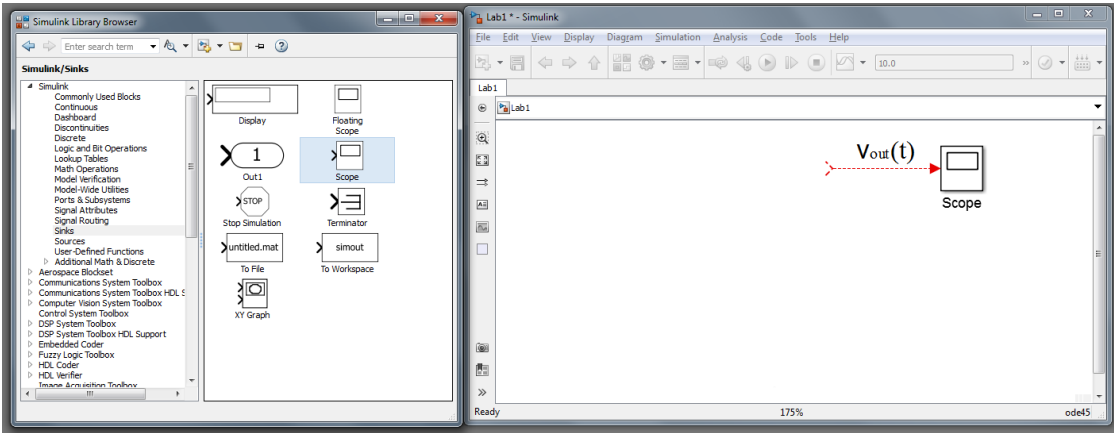


Figura 3. Modelarea parțială a sistemului în *Simulink*

Semnalul $v_{out}(t)$ este momentan “în aer” (linie roșie punctată), iar pentru a-l clarifica putem exprima $v_{out}(t)$ în funcție de $i(t)$ din ecuația (2):

$$v_{out}(t) = \frac{1}{C} \cdot \int i(t) dt \quad (3)$$

Pentru a obține $v_{out}(t)$, trebuie aplicate operațiile de integrare (*Integrator*) și înmulțire cu o constantă (*Gain*) asupra semnalului $i(t)$.

/Deoarece $\frac{1}{C} \cdot \int i(t) dt = \int \frac{1}{C} \cdot i(t) dt$, blocurile de integrare și amplificare pot fi interschimbate în schemă, însă nu mereu este posibil acest lucru.

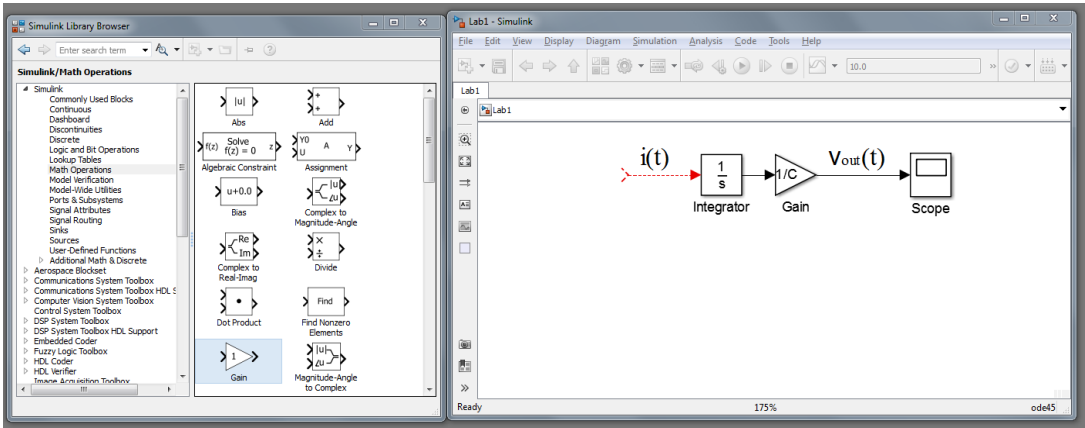


Figura 4. Modelarea parțială a sistemului în *Simulink*

Din ecuația (1), putem exprima semnalul $i(t)$ ca o sumă (diferență).

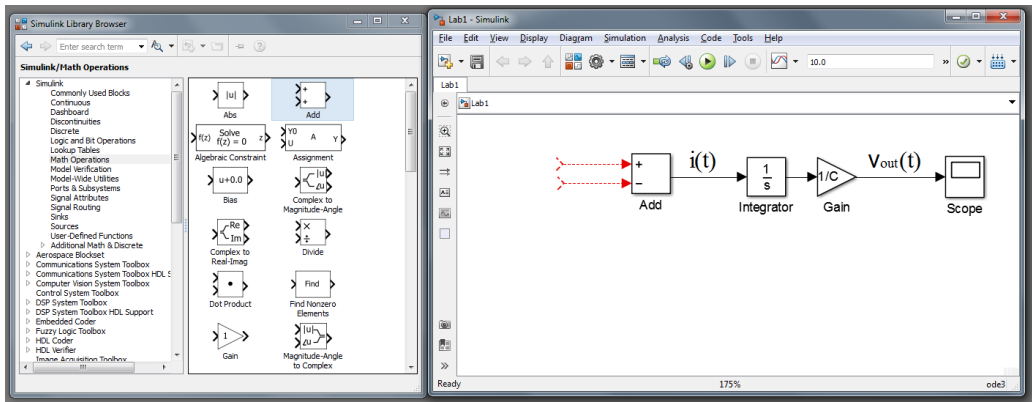


Figura 5. Modelarea parțială a sistemului în *Simulink*

Pe intrarea "+" este semnalul de intrare, înmulțit cu o constantă.

Pe intrarea "-" semnalul este cules de pe firul $v_{out}(t)$ și înmulțit cu o constantă.

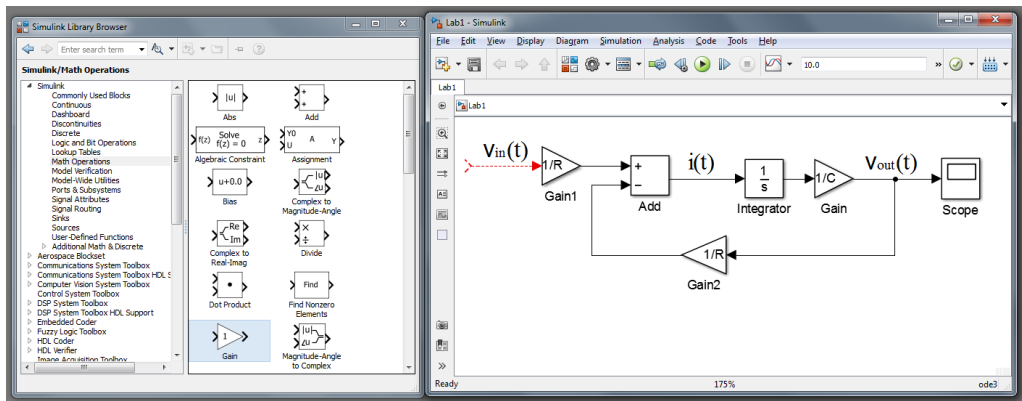


Figura 6. Modelarea parțială a sistemului în *Simulink*

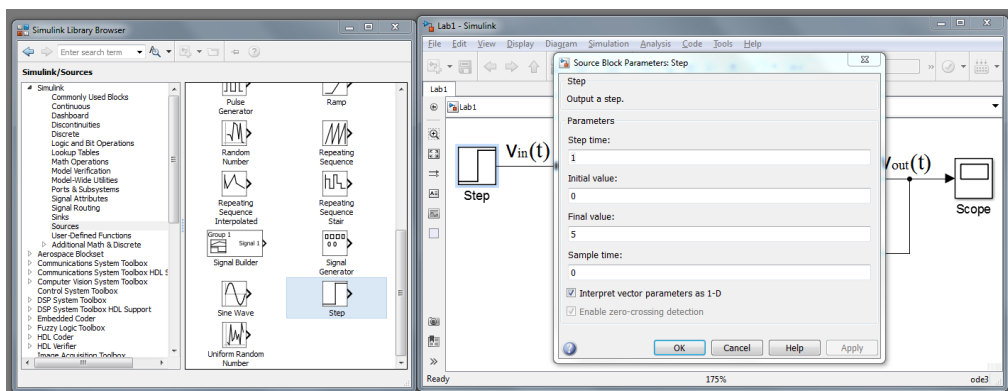


Figura 7. Modelarea sistemului în *Simulink* (configurarea semnalului de intrare)

Pentru a verifica încărcarea condensatorului, putem alege pentru tensiunea de alimentare un semnal treaptă.

Important! Înainte de rulare, trebuie ajustați parametrii de configurare ai modelului:

- timpul de rulare a simulării (*start & stop time*)
- tipul simulării (*fixed step, solver ODE3*), prin care se configurează metodele de rezolvare a modelului numeric. În spatele acestor setări stă foarte multă teorie din domeniul matematicii discrete, dar pentru simulările din această lucrare este suficient de știut că setările recomandate sunt: ” *step* ” = ” *fixed step* ” și ” *solver* ” = ” *solver ODE3* ”
- *step-size*, prin care se stabilește frecvența de eșantionare a semnalului numeric de intrare.

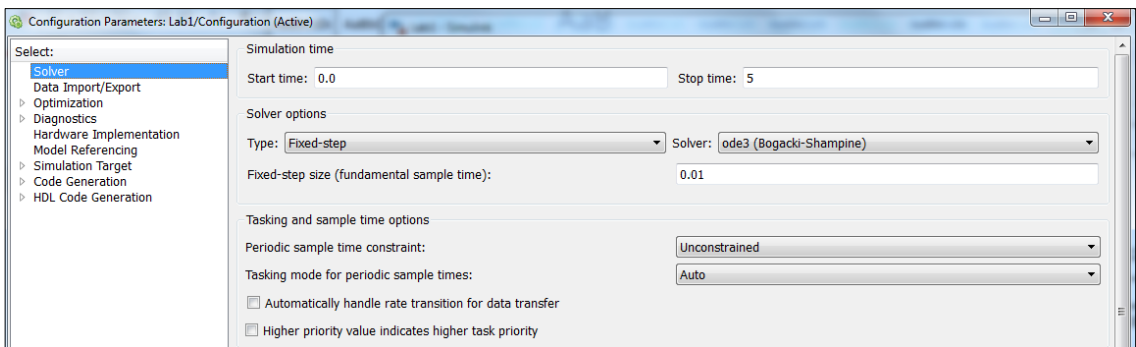


Figura 8. Parametrii de configurare ai sistemului în *Simulink*

Aplicație 3. Să se realizeze modelul circuitului RC serie folosind mai întâi ecuația (1) pentru exprimarea semnalului $v_{out}(t)$ în funcție de alte semnale.

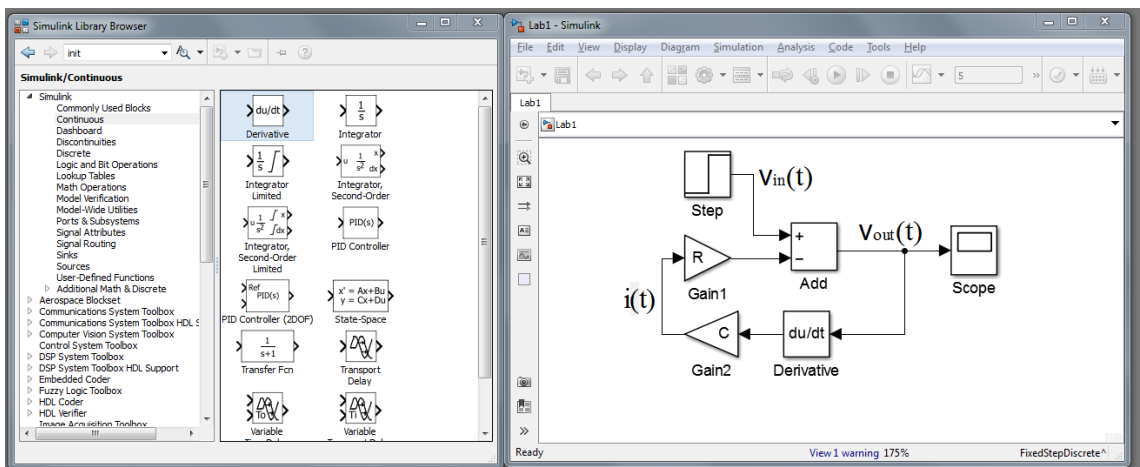
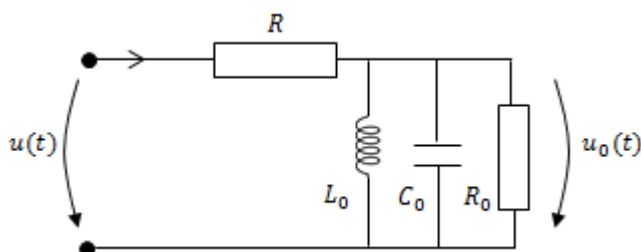


Figura 9. Modelare alternativă a sistemului în *Simulink*

Probleme și aplicații

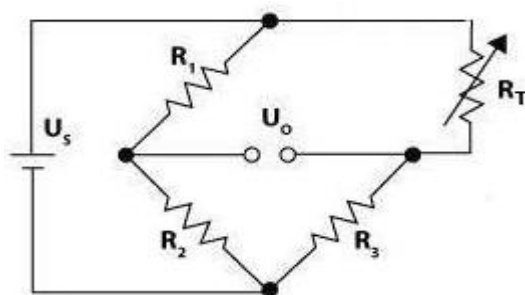
1. Să se realizeze modelul circuitului RC serie folosind blocul de integrare și un singur amplificator.

2. Fie circuitul de mai jos, în care tensiunea de alimentare $u(t)$ este semnal de intrare și tensiunea pe rezistența R_0 este semnal de ieșire.



Să se scrie sistemul de ecuații diferențiale care descriu circuitul și să se realizeze modelul pentru următorii parametri: $R = 1\text{k}\Omega$, $R_0 = 20\Omega$, $L_0 = 100\text{mH}$, $C_0 = 10\mu\text{F}$, $u(t) = \sin(2\pi ft + \varphi_0)$, $f = 2\text{Hz}$, $\varphi_0 = \pi/4$.

3. Fie circuitul de mai jos în care tensiunea de alimentare $u_s(t)$ este semnal de intrare și tensiunea $u_o(t)$ este semnal de ieșire.



a) să se scrie sistemul de ecuații diferențiale care descriu circuitul și să se realizeze modelul, pentru diverse combinații de valori ale rezistențelor.

b) ce se întâmplă dacă puntea este echilibrată?

c) dacă $R_1 = R_2 = R_3 = 100\Omega$, pentru ce valoare R_T semnalul de ieșire este mereu inversat și atenuat de 10 ori față de semnalul de intrare ($U_o = -0.1 \cdot U_s$)?

4. Să se realizeze modelul circuitului RLC serie ($R = 20\Omega$, $L = 100\text{mH}$, $C = 10\mu\text{F}$, $v_{in}(t) = \sin(2\pi ft + \varphi_0)$, $f = 2\text{Hz}$, $\varphi_0 = \pi/4$) folosind:

- a) doar blocuri de integrare
- b) doar blocuri de derivare
- c) un bloc de integrare și unul de derivare

Să se afișeze pe același grafic (*Scope*) tensiunea pe condensator, cea pe bobină și cea pe rezistor.

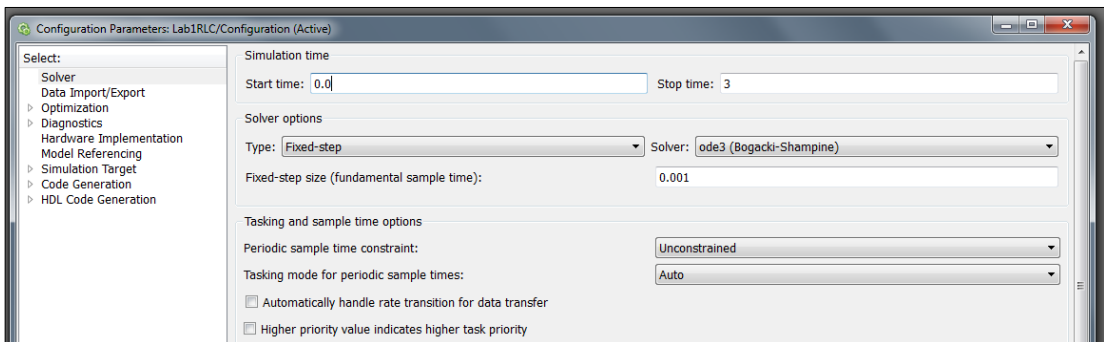


Figura 10. Parametrii de configurare ai simulării

Reprezentarea și simularea modelelor sistemelor mecanice

Obiective: construirea modelului unui sistem mecanic cu blocuri *Simulink*, pornind de la ecuațiile diferențiale care descriu sistemul.

Noțiuni teoretice

Ca și în cazul circuitelor electrice:

- blocurile *Simulink* reprezintă **operații**
- firele (legăturile între blocuri) sunt **semnale**

Modelarea sistemelor fizice urmează aceleași etape:

- scrierea ecuațiilor diferențiale care descriu sistemele
- adăugarea de fire, pornind de la semnalul de ieșire, și de blocuri, conform operațiilor matematice din ecuațiile diferențiale, până la completarea modelului.
- introducerea condițiilor inițiale

Spre deosebire de circuitele electrice, unde condițiile inițiale sunt nule în majoritatea cazurilor, la sisteme mecanice condițiile inițiale sunt foarte importante, în special pentru sistemele asupra cărora nu se acționează. De exemplu, un pendul lăsat în poziția de echilibru va rămâne în poziția de echilibru, rezultând o simulare neinteresantă comparativ cu cazul în care pendulul pornește de la un unghi θ_{init} .

Există de asemenea posibilitatea ca sistemul modelat să aibă atât condiții inițiale nenule, cât și semnal sau semnale de intrare.

Desfășurarea lucrării

Aplicație 1. Un corp de masă M legat de un resort ideal este așezat pe o suprafață perfect orizontală și netedă (fără frecări). Corpul este scos din poziția de echilibru și deplasat pe o distanță x_0 . Din această poziție corpul este lăsat liber. Să se scrie ecuațiile diferențiale care descriu sistemul.

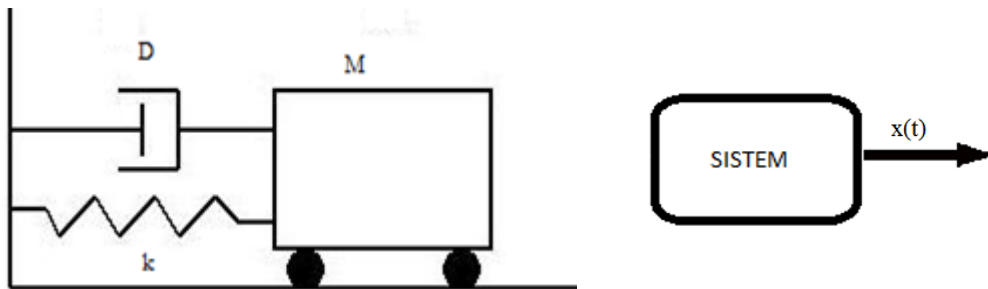


Figura 1. (a) Reprezentarea sistemului fizic și a modelului abstract

Ecuațiile diferențiale care descriu sistemul sunt:

$$\begin{cases} M \cdot a(t) = -k \cdot x(t) - D \cdot v(t) & (1) \text{ legea II a dinamicii} \\ v(t) = \frac{dx(t)}{dt} & (2) \text{ definiția vitezei} \\ a(t) = \frac{dv(t)}{dt} & (3) \text{ definiția accelerației} \end{cases}$$

unde:

- M = masa corpului (*constantă*)
- D = constanta de amortizare (*constantă*)
- k = constanta elastică (*constantă*)
- $a(t)$ = accelerația corpului (*semnal*)
- $v(t)$ = viteza corpului (*semnal*)
- $x(t)$ = poziția corpului, distanța față de poziția de echilibru (*semnal*)

Realizarea modelului pornind de la ecuațiile diferențiale începe prin adăugarea blocului *Scope* pentru vizualizarea semnalului de ieșire.

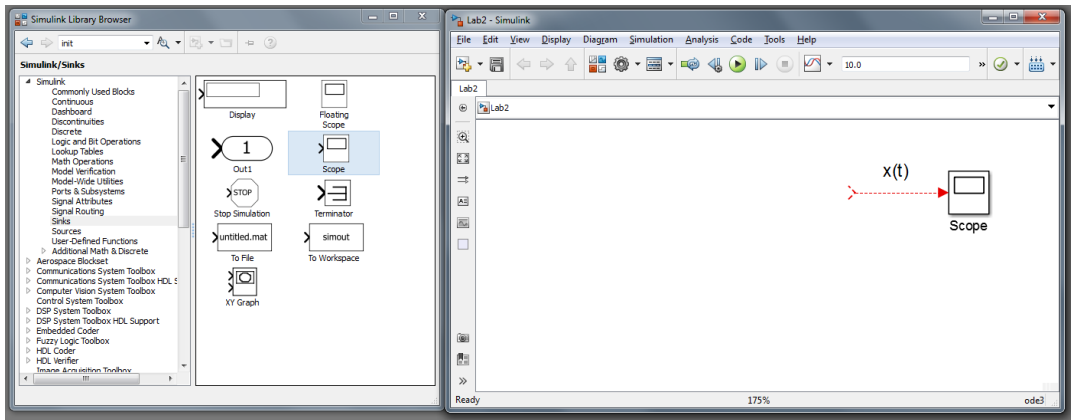


Figura 2. Modelarea parțială a sistemului în *Simulink*

Din ecuațiile (2) și (3) se construiește lanțul de legături între semnalele poziție, viteză și accelerație:

$$x(t) + x_0 = \int v(t)dt \quad (4)$$

$$v(t) + v_0 = \int a(t)dt \quad (5)$$

unde x_0 și v_0 sunt constantele de integrare și corespund condițiilor inițiale ale sistemului.

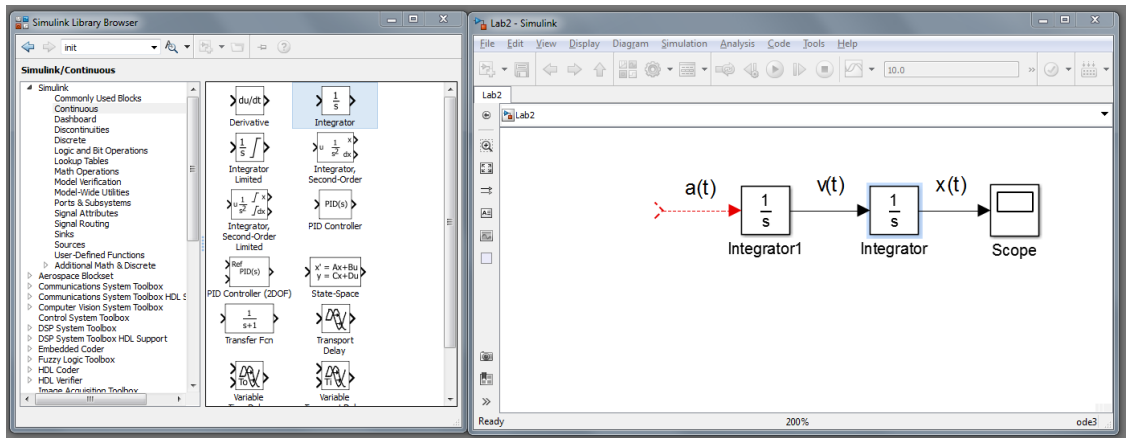


Figura 3. Modelarea parțială a sistemului în *Simulink*

Semnalul $a(t)$ este momentan “în aer” (linie roșie punctată), dar îl putem exprima în funcție de $v(t)$, $x(t)$ din ecuația (1):

$$a(t) = -\frac{k}{M} \cdot x(t) - \frac{D}{M} \cdot v(t) \quad (6)$$

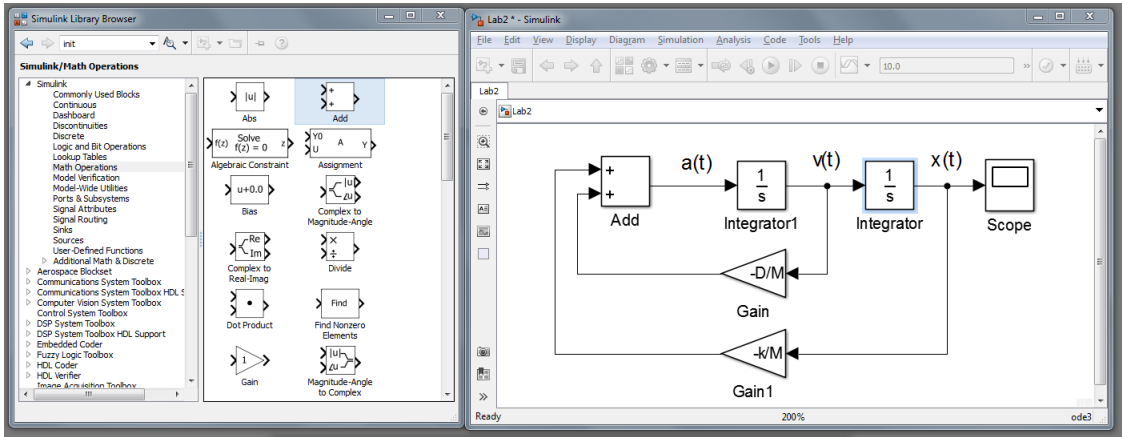


Figura 4. Modelarea sistemului în *Simulink*

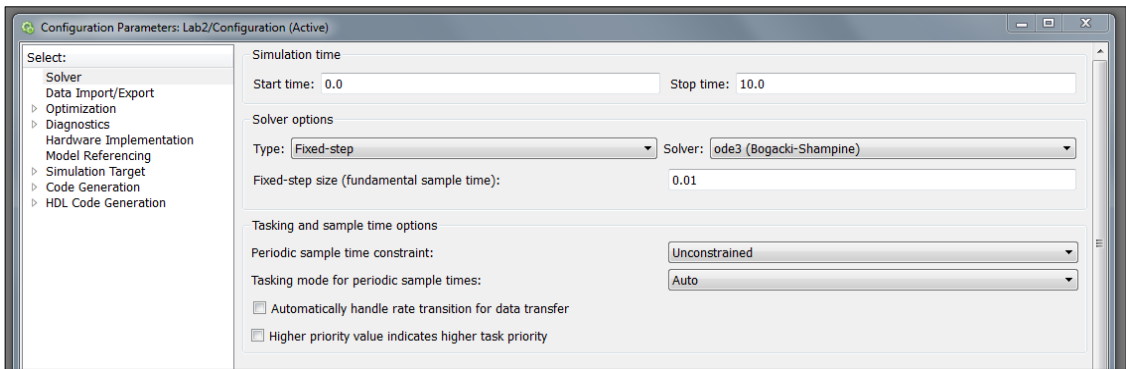


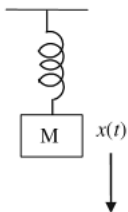
Figura 5. Parametrii de configurare ai sistemului în *Simulink*

Probleme și aplicații

1. Pentru $M = 5 \text{ kg}$, $k = 11 \frac{\text{N}}{\text{m}}$, $D = 0 \frac{\text{Ns}}{\text{m}}$ și condițiile inițiale $x_0 = 1, v_0 = 0$ (corpul este deplasat cu 1m înspre dreapta, apoi lăsat să oscileze liber):
- să se afișeze graficele semnalelor $x(t), v(t), a(t)$.
 - să se determine pe grafic perioada de oscilație a corpului. *Notă: pentru condiții inițiale 0, simularea confirmă intuiția că un corp lăsat în poziția de echilibru rămâne în poziția de echilibru $x(t) = 0, \forall t$.*
 - să se repete experimentul pentru deplasarea inițială a corpului spre stânga cu 1m.
 - să se repete experimentul pentru alte valori ale constantei elastice k
 - să se repete experimentul pentru cazul în care corpul este lăsat în poziția de echilibru, însă i se imprimă inițial o viteză de 4 m/s înspre dreapta.
 - să se repete experimentul pentru cazul în care corpul este deplasat din poziția de echilibru cu 1m spre dreapta și i se imprimă inițial o viteză de 4 m/s tot înspre dreapta.
 - să se repete experimentele pentru o constantă de amortizare $D > 0$.

2. Un corp cu masa $M = 2 \text{ kg}$ este legat de un resort vertical (cântar) având constanta elastică $k = 11 \frac{\text{N}}{\text{m}}$ și constanta de amortizare $D = 3 \frac{\text{Ns}}{\text{m}}$. Poziția inițială a corpului (în momentul în care este agățat de cântar) este $x(0) = 0 \text{ m}$ și ia valori pozitive dacă întinde resortul.

- să se implementeze modelul în *Simulink* și să se afișeze graficele semnalelor $x(t), v(t), a(t)$.
- să se determine poziția la care se stabilizează corpul și durata de stabilizare.



Notă: In echilibrul forțelor acum mai apare o componentă (gravitația).

$$M \cdot a(t) \pm M \cdot g = -k \cdot x(t) - D \cdot v(t)$$

Dacă semnul noii forțe este ales greșit, se modelează amortizarea de la mașină (pentru care punctul fix al resortului este pământul, iar corpul – caroseria mașinii – comprimă resortul).

3. O bilă este aruncată vertical (în sus) cu viteza inițială $v_{init} = 4\text{m/s}$ (frecarea cu aerul este ignorată). Să se implementeze modelul în *Simulink* și să se afișeze graficele poziției, vitezei și accelerației, adică semnalele $x(t)$, $v(t)$, $a(t)$.

4. O bilă este aruncată de la nivelul solului la un unghi $\theta = 30^\circ$ cu viteza inițială $v_{init} = 4\text{m/s}$. Să se implementeze modelul în *Simulink* și să se afișeze graficele poziției, vitezei și accelerației pe axele OX și OY, adică semnalele $x(t)$, $y(t)$, $v_x(t)$, $v_y(t)$, $a_x(t)$, $a_y(t)$, până când bila ajunge înapoi la nivelul solului (perfect orizontal). Prin rulări succesive, să se determine răspunsul la următoarele întrebări:

- pentru ce unghi bila se deplasează cel mai mult pe axa OX până când ajunge înapoi la nivelul solului? (să se păstreze viteza inițială dar să se varieze unghiul de aruncare).
 - cu cât trebuie mărită viteza inițială pentru ca bila să ajungă din nou la nivelul solului de două ori mai departe decât în problema originală? (să se păstreze unghiul $\theta = 30^\circ$).
-

5. O bilă este aruncată vertical (în sus) cu viteza inițială $v_{init} = 4\text{m/s}$ (frecarea cu aerul este ignorată). La contactul cu solul, bila sare (își modifică instantaneu direcția de deplasare, iar viteza de deplasare se înjumătățește). Să se implementeze modelul în *Simulink* și să se afișeze graficele poziției, vitezei și accelerației, adică semnalele $x(t)$, $v(t)$, $a(t)$, pentru primele 3 sărituri ale bilei și respectiv pentru primele 30 secunde.

Modelarea sistemelor cu ajutorul funcției de transfer

Obiective: determinarea funcției de transfer; modelarea în *Simulink* și *Matlab Script* a sistemelor cu ajutorul funcției de transfer.

Noțiuni teoretice

Funcția de transfer $H(s)$ este transformata Laplace a răspunsului la impuls al sistemului $h(t)$.

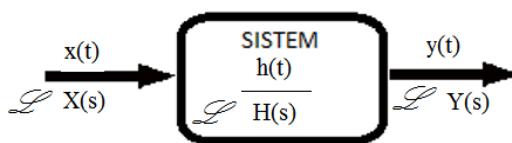


Figura 1. Sistem cu un semnal de intrare și unul de ieșire

$$\text{In domeniul timp: } y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) \cdot h(t - \tau) d\tau \quad (1)$$

$$\text{In domeniul } s \text{ (Laplace): } Y(s) = X(s) \cdot H(s) \Rightarrow H(s) = \frac{Y(s)}{X(s)} \quad (2)$$

Pentru a calcula funcția de transfer Laplace a unui sistem, trebuie parcurse următoarele etape:

- scrierea ecuațiilor diferențiale care descriu sistemul
- transformarea în domeniul s (Laplace) a semnalelor și a operațiilor
- prelucrarea ecuațiilor în domeniul s (Laplace) pentru a ajunge la o formă potrivită

In cazul sistemelor liniare (și invariante în timp), este util ca funcția de transfer $H(s)$ să fie scrisă sub forma unui raport de polinoame în variabila Laplace s :

$$H(s) = \frac{b_M s^M + \dots + b_1 s^1 + b_0}{a_N s^N + \dots + a_1 s^1 + a_0} = \frac{\sum_{k=0}^M b_k s^k}{\sum_{k=0}^N a_k s^k} \quad (3)$$

Desfășurarea lucrării

Fie circuitul RLC serie pentru care tensiunea de alimentare $u(t)$ se consideră semnal de intrare, iar curentul $i(t)$ se consideră semnal de ieșire.

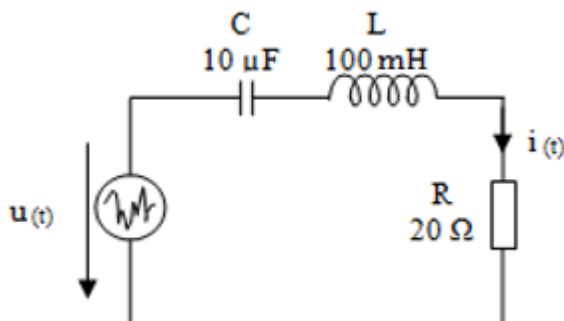


Figura 2. Circuit RLC serie

Aplicație 1. Să se determine funcția de transfer Laplace $H(s)$ pentru acest sistem.

Ecuțiile diferențiale care descriu sistemul sunt:

$$\begin{cases} u(t) = R \cdot i(t) + u_c(t) + u_L(t) \\ i(t) = C \cdot \frac{du_c(t)}{dt} \\ u_L(t) = L \cdot \frac{di(t)}{dt} \end{cases}$$

Tabel 1. Transformarea în domeniul s (Laplace) a operațiilor

Operație în domeniul timp		Operație în domeniul s (Laplace)	
Inmulțire cu constantă	$k \cdot x(t)$	$k \cdot X(s)$	Inmulțire cu constantă
Adunare/Scădere	$x(t) \pm y(t)$	$X(s) \pm Y(s)$	Adunare/Scădere
Integrare	$\int x(t) dt$	$\frac{1}{s} \cdot X(s)$	Inmulțire cu $1/s$
Derivare	$\frac{dx(t)}{dt}$	$s \cdot X(s)$	Inmulțire cu s
Convoluție	$x(t) * h(t)$	$X(s) \cdot H(s)$	Produs

Transformarea în domeniul s (Laplace) a semnalelor: fie $U(s)$ corespondentul Laplace al semnalului $u(t)$, $I(s)$ corespondentul Laplace al semnalului $i(t)$ și așa mai departe pentru toate semnalele.

$$u(t) \stackrel{L}{\leftrightarrow} U(s), u_c(t) \stackrel{L}{\leftrightarrow} U_c(s), u_L(t) \stackrel{L}{\leftrightarrow} U_L(s), i(t) \stackrel{L}{\leftrightarrow} I(s)$$

Sistemul de ecuații devine în domeniul Laplace:

$$\begin{cases} U(s) = R \cdot I(s) + U_c(s) + U_L(s) \\ I(s) = s \cdot C \cdot U_c(s) \\ U_L(s) = s \cdot L \cdot I(s) \end{cases}$$

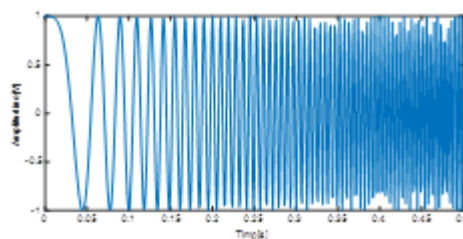
Eliminând semnalele interne, putem obține o relație între semnalul de intrare $U(s)$ și cel de ieșire $I(s)$:

$$U(s) = R \cdot I(s) + \frac{1}{s \cdot C} \cdot I(s) + s \cdot L \cdot I(s)$$

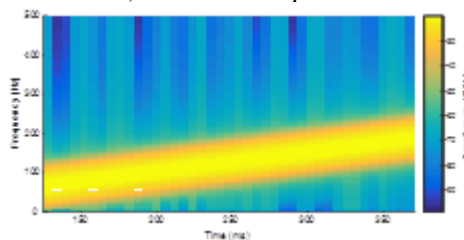
$$H(s) = \frac{I(s)}{U(s)} = \frac{1}{R + \frac{1}{s \cdot C} + s \cdot L} = \frac{s \cdot C}{s^2 \cdot L \cdot C + s \cdot R \cdot C + 1}$$

Aplicație 2. Să se modeleze sistemul în *Simulink* și să se afișeze pe grafic (*Scope*) semnalul de ieșire $i(t)$, pentru un semnal de intrare *chirp* liniar care pornește de la $t = 0s, f = 0\text{Hz}$ și ajunge la $t = 1s$ și $f = 500\text{Hz}$.

Semnalul *chirp* liniar este o sinusoidă pentru care frecvența se modifică liniar în timp.



a) Semnalul *chirp* liniar



b) Spectrograma semnalului *chirp* liniar

Figura 3. Semnalul *chirp* liniar

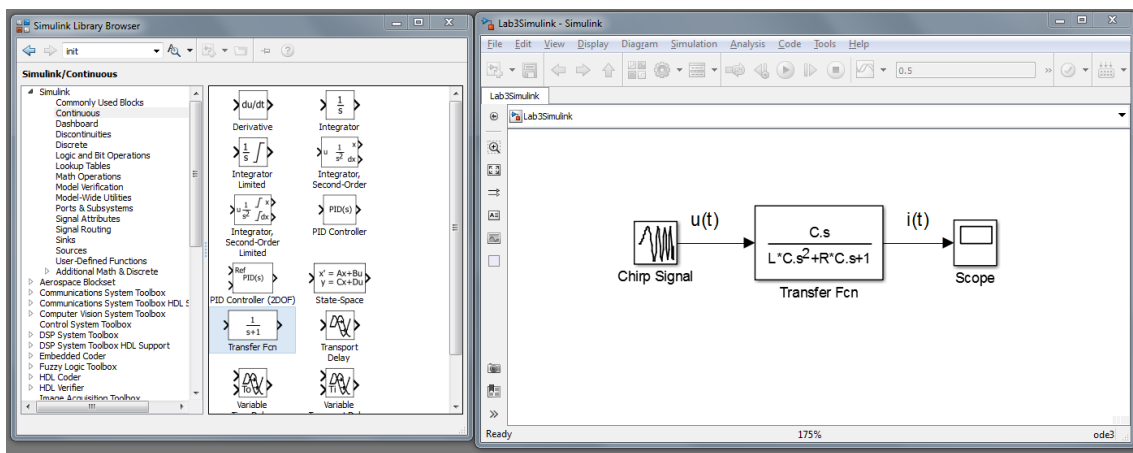
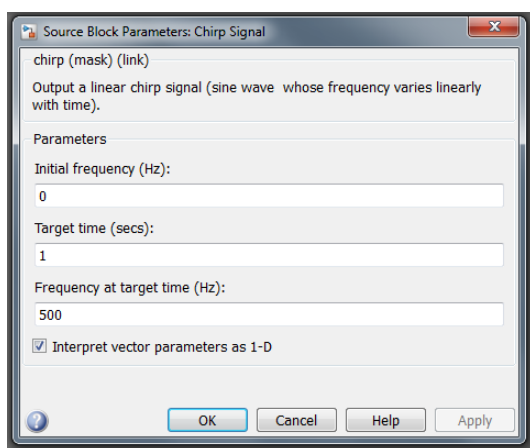
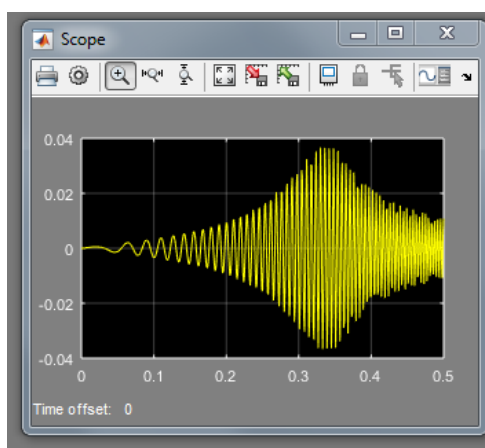


Figura 4. Modelarea circuitului RLC serie cu ajutorul funcției de transfer Laplace



a



b

Figura 5. Răspunsul sistemului la semnalul *chirp* liniar din modelarea *Simulink*

Aplicație 3. Să se modeleze sistemul în *Matlab Script* și să se afișeze pe grafic semnalul de ieșire $i(t)$, pentru un semnal de intrare *chirp* liniar care pornește de la $t = 0s, f = 0Hz$ și ajunge la $t = 1s, f = 500Hz$.

Pentru definirea unui sistem cu ajutorul funcției de transfer se poate folosi comanda TF, iar pentru rularea simulării (tregerii prin sistem a unui semnal de intrare) comanda LSIM.

```
system = tf(numarator,numitor);
```

```
y = lsim(system,x,t);
```


Aplicație 4. Să se analizeze sistemul în timp și frecvență și să se determine frecvența de rezonanță și amplificarea semnalului de ieșire la acea frecvență prin următoarele metode:

- analiza semnalului de ieșire atunci când intrarea este un semnal *chirp* liniar
- analiza diagramei *Bode* pentru sistem
- analiza diagramei poli-zero-uri a funcției de transfer $H(s)$
- calcul matematic

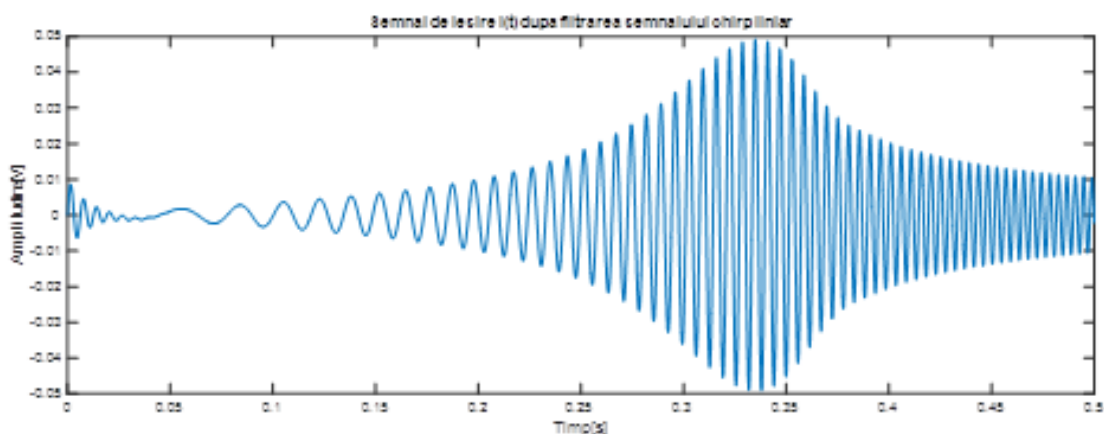


Figura 6. Răspunsul sistemului la semnalul *chirp* liniar din modelarea *Matlab Script*

Deoarece semnalul *chirp* liniar folosit are amplitudine 1, amplitudinea maximă a semnalului de ieșire este în același timp și amplificarea (atenuarea în acest caz) sistemului și se poate citi pe grafic $A = 0.492$. Amplitudinea maximă se găsește la $t = 0.335$ secunde. Dacă semnalul de intrare *chirp* pornește de la $t = 0s, f = 0Hz$ și ajunge la $t = 1s, f = 500Hz$, atunci la $t = 0.335s$ avea o frecvență $f_{rez} = 500 \cdot 0.335 = 167.5 Hz$. Ne așteptăm ca această valoare să fie puțin prea mare, deoarece nu se ține cont de întârzierea introdusă de sistem, însă este un estimat bun în lipsa analizei în frecvență.

Aplicație 5. Să se modeleze sistemul în *Simulink* cu ajutorul blocului *Zero-Pole* și să se afișeze pe grafic (*Scope*) semnalul de ieșire $i(t)$, pentru un semnal de intrare *chirp* liniar care pornește de la $t = 0s, f = 0Hz$ și ajunge la $t = 1s, f = 500Hz$.

Pentru s_1 și s_2 alese corespunzător:

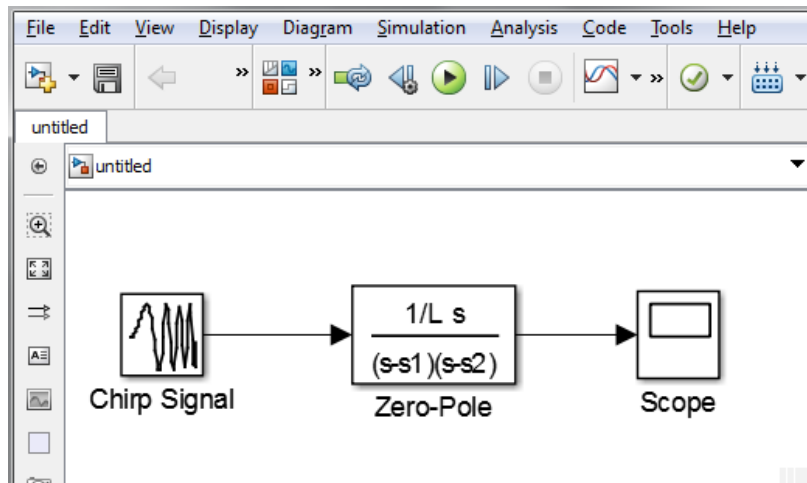


Figura 8. Modelarea circuitului RLC serie cu ajutorul funcției de transfer Laplace descrisă prin poli și zerouri

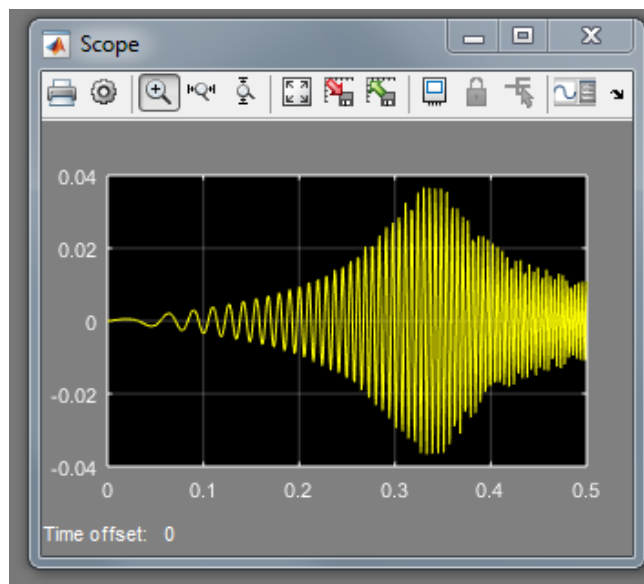
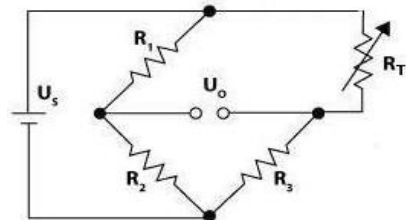
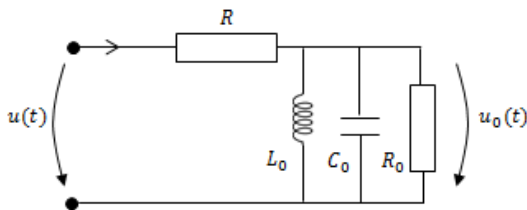


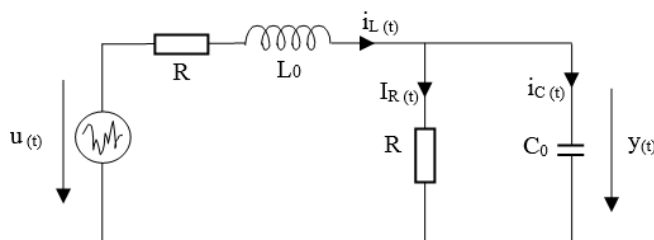
Figura 9. Răspunsul sistemului la semnalul *chirp* liniar din modelarea *Simulink*

Probleme și aplicații

1. Pentru sistemele din laboratorul *Modelarea circuitelor electrice*, să se determine funcțiile de transfer *Laplace* $H(s)$ și să se realizeze modelele în *Simulink* și *Matlab script*. Să se compare rezultatele cu simulările aceluia laborator.



2. Pentru sistemul de mai jos (semnal de intrare $u(t)$ sinusoidal de amplitudine 1V, frecvență 10Hz și durată 1s; $R = 10 \Omega$, $C_0 = 100\mu\text{F}$, $L_0 = 10\text{mH}$), să se determine funcția de transfer *Laplace* $H(s)$ și să se modeleze în *Simulink* și *Matlab script*.



3. Pentru circuitul RLC paralel, să se determine frecvența de rezonanță și atenuarea prin:

- utilizarea unui semnal de intrare *chirp* liniar
- utilizarea diagramei *Bode*
- utilizarea diagramei *poli-zero*
- matematic, din funcția de transfer $H(s)$

4. Pentru problemele din laboratorul *Modelarea sistemelor fizice*, să se determine funcția de transfer *Laplace* $H(s)$ și să se realizeze modelele în *Simulink* și *Matlab script* pentru cazul în care accelerația gravitațională $g(t)$ este un semnal de intrare (variabil în timp).

Modelarea sistemelor cu ajutorul variabilelor de stare

Obiective: determinarea matricelor din modelul cu variabile de stare, modelarea în *Simulink* și *Matlab Script* a sistemelor cu ajutorul ecuațiilor de stare, determinarea controlabilității și observabilității sistemului.

Noțiuni teoretice

Orice sistem linear poate fi reprezentat prin modelul cu variabile de stare (*state-space model*) descris de ecuațiile:

$$\begin{cases} \frac{dx(t)}{dt} = A \cdot x(t) + B \cdot u(t) & (1) \\ y(t) = C \cdot x(t) + D \cdot u(t) & (2) \end{cases}$$

Prima ecuație a modelului se numește *ecuație de stare*, iar ecuația a doua se numește *ecuație de ieșire*.

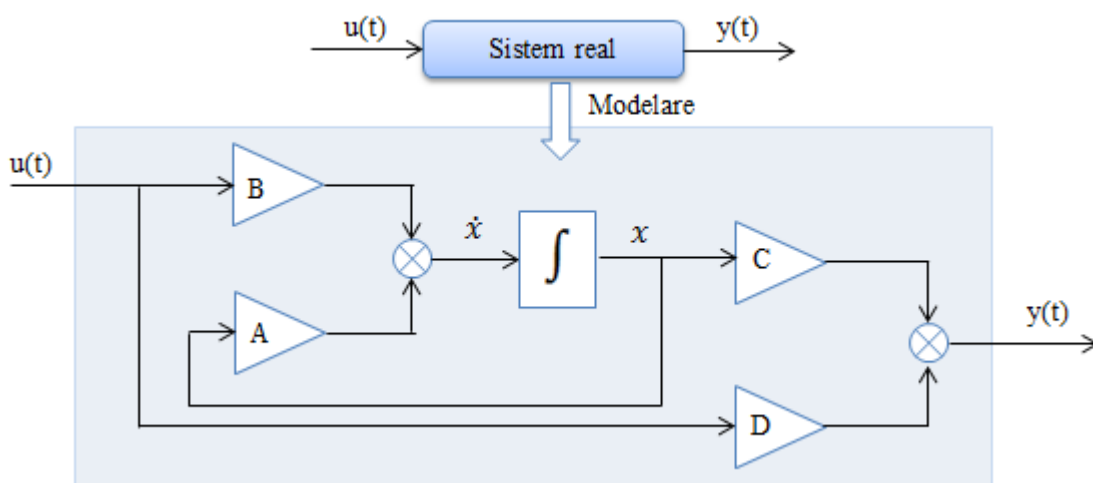


Figura 1. Schema bloc (*flowchart*) pentru modelul cu ecuații de stare

Pentru a determina modelul cu variabile de stare a unui sistem, trebuie parcurse următoarele etape:

- scrierea ecuațiilor diferențiale care descriu sistemul și identificarea intrărilor, ieșirilor și a variabilelor de stare din sistem.
- prelucrarea ecuațiilor diferențiale în domeniul timp pentru a ajunge la ecuațiile modelului cu variabile de stare.
- identificarea coeficienților/matricelor A, B, C, D

Variabilele de stare sunt elementele “cu memorie” din sistem (cele care implică schimburi energetice). Acestea pot fi identificate și în sistemul de ecuații diferențiale, fiind semnalele care se derivează.

Dacă sistemul are un semnal de intrare $u(t)$, un semnal de ieșire $y(t)$ și o variabilă de stare $x(t)$, atunci va fi descris de cele două ecuații de mai sus, unde A, B, C, D sunt coeficienți (constante).

Dacă sistemul are M semnale de intrare, N variabile de stare și Q semnale de ieșire, va fi descris prin aceleași două ecuații, însă acestea devin ecuații matriceale:

$$\begin{cases} \dot{\mathbf{X}}_{N \times 1} = \mathbf{A}_{N \times N} \cdot \mathbf{X}_{N \times 1} + \mathbf{B}_{N \times M} \cdot \mathbf{U}_{M \times 1} & (3) \\ \mathbf{Y}_{Q \times 1} = \mathbf{C}_{Q \times N} \cdot \mathbf{X}_{N \times 1} + \mathbf{D}_{Q \times M} \cdot \mathbf{U}_{M \times 1} & (4) \end{cases}$$

Tabel 1. Vectorii și matricele modelului cu variabile de stare

Matricea	Structura	Dim.	Denumire
X	$\begin{bmatrix} x_1(t) \\ x_1(t) \\ \dots \\ x_N(t) \end{bmatrix}$	$N \times 1$	matricea de stare (matricea de tranziție a stărilor)
$\frac{dX}{dt} = \dot{X}$	$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dots \\ \dot{x}_N(t) \end{bmatrix}$	$N \times 1$	matricea derivatei variabilelor de stare
U	$\begin{bmatrix} u_1(t) \\ u_2(t) \\ \dots \\ u_M(t) \end{bmatrix}$	$M \times 1$	matricea variabilelor de intrare
Y	$\begin{bmatrix} y_1(t) \\ y_2(t) \\ \dots \\ y_Q(t) \end{bmatrix}$	$Q \times 1$	matricea variabilelor de ieșire
A	$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix}$	$N \times N$	matricea sistemului (<i>system matrix</i>)
B	$\begin{bmatrix} b_{11} & b_{12} & \dots & b_{1M} \\ b_{21} & b_{22} & \dots & b_{2M} \\ \dots & \dots & \dots & \dots \\ b_{N1} & b_{N2} & \dots & b_{NM} \end{bmatrix}$	$N \times M$	matricea de intrare (<i>input matrix</i>)
C	$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1N} \\ c_{21} & c_{22} & \dots & c_{2N} \\ \dots & \dots & \dots & \dots \\ c_{Q1} & c_{Q2} & \dots & c_{QN} \end{bmatrix}$	$Q \times N$	matricea de ieșire (<i>output matrix</i>)
D	$\begin{bmatrix} d_{11} & d_{12} & \dots & d_{1M} \\ d_{21} & d_{22} & \dots & d_{2M} \\ \dots & \dots & \dots & \dots \\ d_{Q1} & d_{Q2} & \dots & d_{QM} \end{bmatrix}$	$Q \times M$	matricea de transfer direct (<i>direct transmittance matrix</i>)

Desfășurarea lucrării

Aplicație 1. Fie circuitul de mai jos, pentru care tensiunea de alimentare $u(t)$ se consideră semnal de intrare, iar tensiunea pe condensator $y(t)$ se consideră semnal de ieșire. Să se determine matricile A, B, C, D din modelul cu variabile de stare pentru acest sistem.

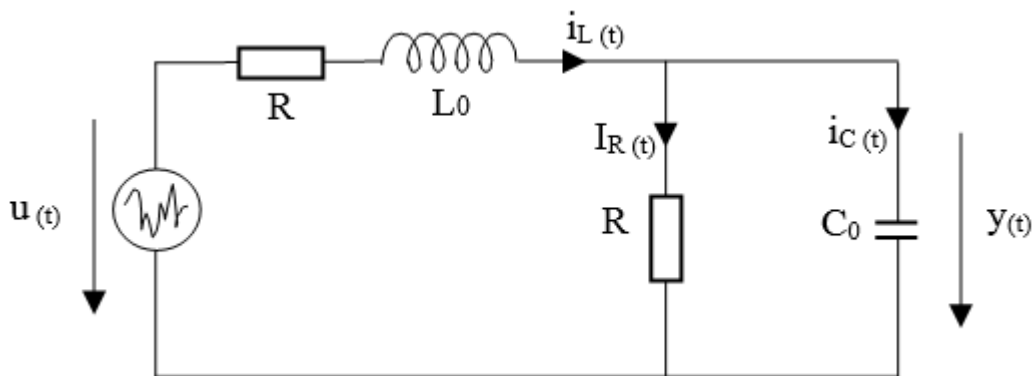


Figura 2. Circuitul propus

Ecuțiile diferențiale care descriu sistemul sunt:

$$\begin{cases} u(t) = R \cdot i_L(t) + u_L(t) + y(t) & (1) \\ i_L(t) = i_R(t) + i_C(t) & (2) \\ y(t) = R \cdot i_R(t) & (3) \\ u_L(t) = L_0 \cdot \frac{di_L(t)}{dt} & (4) \\ i_C(t) = C_0 \cdot \frac{dy(t)}{dt} & (5) \end{cases}$$

Matricea variabilelor de intrare: $\mathbf{U} = [u(t)] \quad M = 1$

Matricea variabilelor de stare: $\mathbf{X} = \begin{bmatrix} i_L(t) \\ y(t) \end{bmatrix} \quad N = 2$

Matricea variabilelor de ieșire: $\mathbf{Y} = [y(t)] \quad Q = 1$

După cum se observă și din ecuațiile diferențiale, variabilele de stare pentru un circuit electric sunt **curenții prin bobine și tensiunile pe condensatoare**.

Înainte de prelucrarea ecuațiilor diferențiale este util să știm la ce dorim să ajungem.

Ecuația de stare din model $\dot{X} = A \cdot X + B \cdot U$ poate fi scrisă desfășurată astfel:

$$\begin{aligned} \begin{bmatrix} di_L(t)/dt \\ dy(t)/dt \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} i_L(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} \cdot u(t) \Rightarrow \\ \Rightarrow \begin{cases} di_L(t)/dt = a_{11} \cdot i_L(t) + a_{12} \cdot y(t) + b_{11} \cdot u(t) \\ dy(t)/dt = a_{21} \cdot i_L(t) + a_{22} \cdot y(t) + b_{21} \cdot u(t) \end{cases} \end{aligned}$$

Adică derivatele variabilelor de stare trebuie exprimate în funcție de variabilele de stare și intrări. Din ecuația (4) $\rightarrow \frac{di_L(t)}{dt} = \frac{1}{L_0} \cdot u_L(t)$. Deoarece $u_L(t)$ nu este nici variabilă de stare și nici semnal de intrare, trebuie înlocuit conform ecuației (1):

$$\frac{di_L(t)}{dt} = \frac{1}{L_0} \cdot (u(t) - R \cdot i_L(t) - y(t))$$

După desfacerea parantezelor și re-aranjarea termenilor, se pot identifica coeficienții a_{11} , a_{12} , b_{11} :

$$\begin{aligned} \frac{di_L(t)}{dt} &= \frac{-R}{L_0} \cdot i_L(t) + \frac{-1}{L_0} \cdot y(t) + \frac{1}{L_0} \cdot u(t) \\ a_{11} &= \frac{-R}{L_0}, \quad a_{12} = \frac{-1}{L_0}, \quad b_{11} = \frac{1}{L_0} \end{aligned}$$

Din ecuația (5) $\rightarrow \frac{dy(t)}{dt} = \frac{1}{C_0} \cdot i_C(t)$. Deoarece $i_C(t)$ nu este nici variabilă de stare și nici semnal de intrare, trebuie înlocuit conform ecuațiilor (2), (3) pentru identificarea coeficienților a_{21} , a_{22} , b_{21} :

$$\begin{aligned} \frac{dy(t)}{dt} &= \frac{1}{C_0} \cdot (i_L(t) - i_R(t)) = \frac{1}{C_0} \cdot \left(i_L(t) - \frac{y(t)}{R} \right) \\ &= \frac{1}{C_0} \cdot i_L(t) + \frac{-1}{C_0 \cdot R} \cdot y(t) + 0 \cdot u(t) \\ a_{21} &= \frac{1}{C_0}, \quad a_{22} = \frac{-1}{C_0 \cdot R}, \quad b_{21} = 0 \end{aligned}$$

Matricea sistemului A și matricea de intrare B sunt așadar:

$$A = \begin{bmatrix} \frac{-R}{L_0} & \frac{-1}{L_0} \\ 1 & \frac{-1}{C_0 \cdot R} \end{bmatrix} \quad B = \begin{bmatrix} 1/L_0 \\ 0 \end{bmatrix}$$

Ecuția de ieșire din model $Y = C \cdot X + D \cdot U$ poate fi scrisă desfășurat astfel:

$$y(t) = [c_{11} \ c_{12}] \cdot \begin{bmatrix} i_L(t) \\ y(t) \end{bmatrix} + [d_{11}] \cdot u(t)$$

Deoarece $y(t)$ este în același timp semnal de ieșire și variabilă de stare, rezultă:

$$y(t) = 0 \cdot i_L(t) + 1 \cdot y(t) + 0 \cdot u(t)$$

$$C = [0 \ 1] \quad D = [0]$$

Aplicație 5. Să se modeleze sistemul în *Simulink* și să se afișeze pe grafic (*Scope*) semnalul de ieșire $y(t)$, pentru un semnal de intrare $u(t)$ sinusoidal de amplitudine 1V, frecvența 10Hz și durata 1s. Se cunosc $R=10 \ \Omega$, $C_0=100\mu\text{F}$, $L_0=10\text{mH}$.

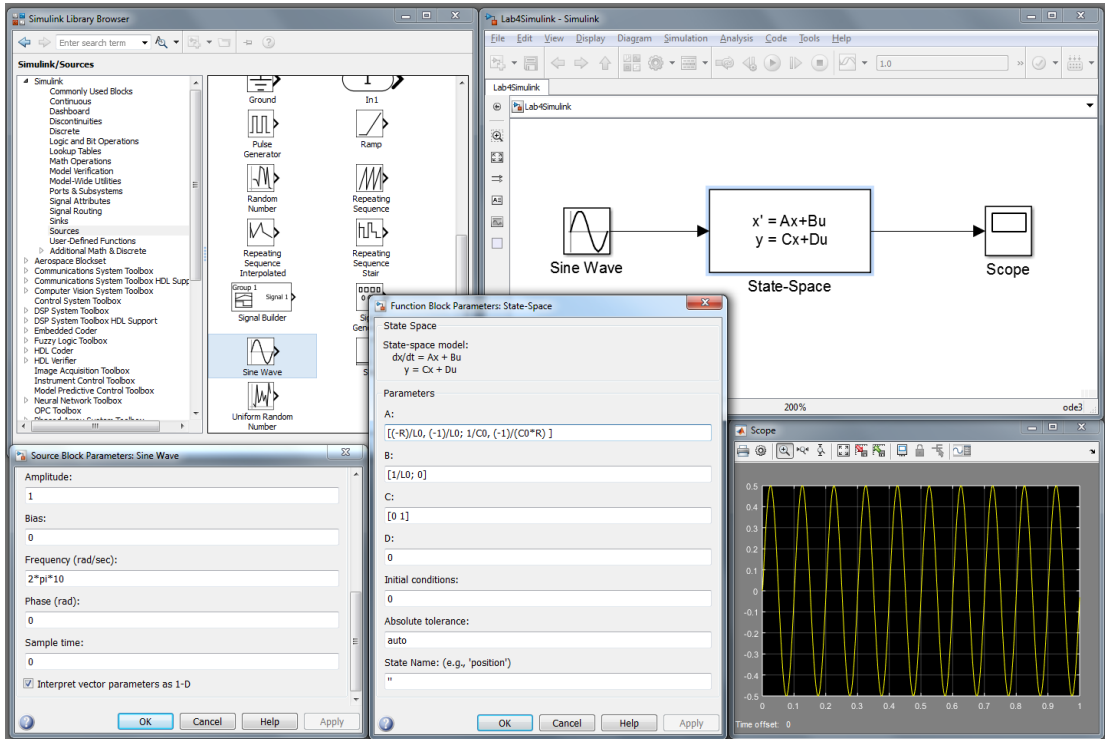


Figura 3. Implementarea *Simulink* a modelului cu variabile de stare

Aplicație 6. Să se modeleze sistemul în *Matlab Script* și să se afișeze pe grafic semnalul de ieșire $y(t)$, pentru un semnal de intrare $u(t)$ sinusoidal de amplitudine 1V, frecvența 10Hz și durata 1s. Se cunosc $R=10 \Omega$, $C_0=100\mu\text{F}$, $L_0=10\text{mH}$.

Pentru definirea unui sistem cu ajutorul modelului cu variabile de stare se poate folosi comanda `ss`, iar pentru rularea simulării (tregerii prin sistem a unui semnal de intrare) se poate folosi comanda `lsim`:

```
sistem = ss(A,B,C,D);  
y = lsim(sistem,x,t);
```

Aplicație 7. Să se determine relația între valorile R , C_0 și L_0 pentru ca sistemul să fie controlabil.

Sistemul este controlabil dacă valorile variabilelor de stare pot fi controlate prin semnalele de intrare ale sistemului. Pentru a determina condiția ca un sistem să fie controlabil se definește *matricea de controlabilitate*, o matrice pătrată de dimensiune $N \times N$.

$$M_C = [B, \quad AB, \quad A^2B, \dots, \quad A^{N-1}B]$$

Sistemul este controlabil numai dacă rangul matricei de controlabilitate este N (determinantul $\Delta M_C \neq 0$).

$$M_C = [B, AB] = \begin{bmatrix} 1 & -R \\ L_0 & \frac{L_0^2}{1} \\ 0 & L_0 C_0 \end{bmatrix}, \Delta M_C \neq 0, \forall R, L_0, C_0$$

Sistemul este controlabil indiferent de valorile R, L_0, C_0 .

Aplicație 8. Să se determine relația între valorile R , C_0 și L_0 pentru ca sistemul să fie observabil.

Sistemul este observabil dacă valorile inițiale ale variabilelor de stare pot fi determinate cu ajutorul semnalelor de intrare U și ieșire Y ale sistemului.

Pentru a determina condiția ca un sistem să fie observabil se definește *matricea de observabilitate*, o matrice pătrată de dimensiune $N \times N$.

$$M_O = \begin{bmatrix} C \\ CA \\ \dots \\ CA^{N-1} \end{bmatrix}$$

Sistemul este observabil numai dacă rangul matricei de controlabilitate este N (determinantul $\Delta M_O \neq 0$).

$$M_O = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{1}{C_0} & \frac{-1}{C_0 \cdot R} \end{bmatrix}$$

$$\Delta M_O \neq 0, \forall R, L_0, C_0$$

Sistemul este observabil indiferent de valorile R, L_0, C_0 .

Aplicație 9. Să se determine funcția de transfer Laplace $H(s)$ pentru acest sistem, pornind de la modelul cu variabile de stare. Să se verifice rezultatul (pentru $R=10\Omega$, $C_0=100\mu\text{F}$, $L_0=10\text{mH}$) prin funcțiile *Matlab* de conversie între modele.

Pentru a calcula funcția de transfer Laplace a unui sistem, trebuie parcurse următoarele etape:

- scrierea ecuațiilor diferențiale care descriu sistemul
- transformarea Laplace a semnalelor și a operațiilor
- prelucrarea ecuațiilor în domeniul Laplace pentru a ajunge la o formă potrivită

Sistemul de ecuații diferențiale matriceale din modelul cu variabile de stare este:

$$\begin{cases} \frac{dX(t)}{dt} = A \cdot X(t) + B \cdot U(t) \\ Y(t) = C \cdot X(t) + D \cdot U(t) \end{cases}$$

Unde $X(t)$, $U(t)$, $Y(t)$ sunt vectori, iar (t) a fost introdus pentru evidențierea faptului că suntem încă în domeniul timp.

Transformarea Laplace a semnalelor:

$$u(t) \xleftrightarrow{L} U(s), y(t) \xleftrightarrow{L} Y(s)$$

$$X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \dots \\ x_M(t) \end{bmatrix} \xleftrightarrow{L} X(s) = \begin{bmatrix} X_1(s) \\ X_2(s) \\ \dots \\ X_M(s) \end{bmatrix}$$

Prin transformarea *Laplace* a operațiilor și ținând cont că nu există condiții inițiale, sistemul devine:

$$\begin{cases} s \cdot X(s) = A \cdot X(s) + B \cdot U(s) \\ Y(s) = C \cdot X(s) + D \cdot U(s) \end{cases}$$

Eliminând variabilele de stare, putem obține o relație între semnalul de intrare $U(s)$ și cel de ieșire $Y(s)$. *Atenție la operațiile cu matrice!*

$$s \cdot X(s) = A \cdot X(s) + B \cdot U(s)$$

$$(s \cdot I_N - A) \cdot X(s) = B \cdot U(s)$$

$$X(s) = (s \cdot I_N - A)^{-1} \cdot B \cdot U(s)$$

$$Y(s) = C \cdot X(s) + D \cdot U(s) = \left(C \cdot (s \cdot I_N - A)^{-1} \cdot B + D \right) \cdot U(s)$$

Pentru problema dată:

$$D = 0$$

$$s \cdot I_N - A = \begin{bmatrix} s + \frac{R}{L_0} & \frac{1}{L_0} \\ -\frac{1}{C_0} & s + \frac{1}{C_0 \cdot R} \end{bmatrix}$$

$$(s \cdot I_N - A)^{-1} = \frac{1}{\left(s + \frac{R}{L_0}\right) \cdot \left(s + \frac{1}{C_0 \cdot R}\right) + \frac{1}{L_0 C_0}} \cdot \begin{bmatrix} s + \frac{1}{C_0 \cdot R} & -\frac{1}{L_0} \\ \frac{1}{C_0} & s + \frac{R}{L_0} \end{bmatrix}$$

$$= \frac{1}{\Delta} \cdot \begin{bmatrix} s + \frac{1}{C_0 \cdot R} & -\frac{1}{L_0} \\ \frac{1}{C_0} & s + \frac{R}{L_0} \end{bmatrix}$$

$$C \cdot (s \cdot I_N - A)^{-1} = \frac{1}{\Delta} \cdot [0 \ 1] \cdot \begin{bmatrix} s + \frac{1}{C_0 \cdot R} & -\frac{1}{L_0} \\ \frac{1}{C_0} & s + \frac{R}{L_0} \end{bmatrix} = \frac{1}{\Delta} \cdot \left[\frac{1}{C_0} \quad s + \frac{R}{L_0} \right]$$

$$C \cdot (s \cdot I_N - A)^{-1} \cdot B = \frac{1}{\Delta} \cdot \left[\frac{1}{C_0} \quad s + \frac{R}{L_0} \right] \cdot \begin{bmatrix} 1/L_0 \\ 0 \end{bmatrix} = \frac{1}{\Delta} \cdot \frac{1}{L_0 C_0}$$

Deoarece sistemul are un semnal de intrare și un semnal de ieșire, se poate calcula funcția de transfer:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{1}{\Delta} \cdot \frac{1}{L_0 C_0} = \frac{1}{s^2 \cdot L_0 \cdot C_0 + s \cdot (R \cdot C_0 + L_0/R) + 2}$$

Pentru *Matlab*, există funcții care să facă conversia între diverse modele ale sistemului, precum cele din *Tabel 2*.

Tabel 2. Funcții *Matlab* de conversie între modele

Conversie din / Conversie în	TF	ZP	SS
TF	-	zp2tf	ss2tf
ZP	tf2zp	-	ss2zp
SS	tf2ss	zp2ss	-

TF: modelarea sistemului prin funcția de transfer

ZP: modelarea sistemului prin precizarea poziției polilor și zerourilor funcției de transfer

SS: modelarea sistemului cu ajutorul variabilelor de stare

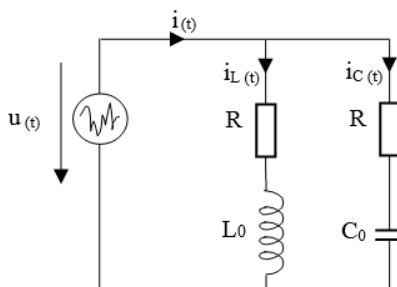
Probleme și aplicații

1. Fie circuitul de mai jos, pentru care tensiunea de alimentare $u(t)$ se consideră semnal de intrare, iar curentul $i(t)$ se consideră semnal de ieșire.

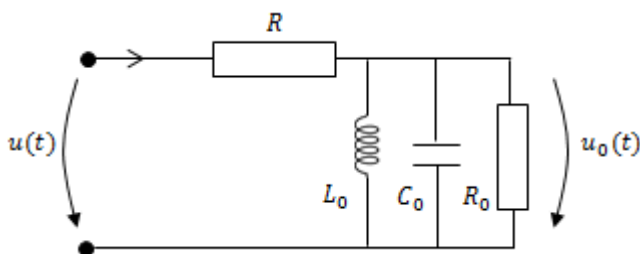
a) să se determine matricele A, B, C, D din modelul cu variabile de stare pentru acest sistem.

b) să se modeleze sistemul în *Simulink* și *Matlab Script*.

c) să se determine relația dintre valorile R, C_0 și L_0 pentru ca sistemul să fie controlabil/observabil.



2. Pentru circuitul din imagine:



a) să se scrie sistemul de ecuații diferențiale care descrie circuitul.

b) să se determine matricele A, B, C, D din modelul cu variabile de stare considerând semnal de intrare $u(t)$ și semnal de ieșire $u_o(t)$.

c) să se discute observabilitatea și controlabilitatea sistemului.

3. Să se arate că circuitele RLC serie și RLC paralel sunt observabile și controlabile pentru orice valori pozitive ale parametrilor R, L și C .

4. Să se scrie o funcție care să primească la intrare oricare dintre modelele unui sistem liniar (TF, ZP, SS) și să afișeze funcția de transfer, diagrama poli-zero și matricele A , B , C , D .

Stabilitatea sistemelor liniare

Obiective: Determinarea stabilității sistemelor liniare prin diverse criterii.

Noțiuni teoretice

Stabilitatea unui sistem este esențială, deoarece un sistem instabil duce de regulă la distrugerea sa. Un sistem este stabil în sens BIBO (Bounded Input Bounded Output = Intrare Mărginită Ieșire Mărginită) dacă și numai dacă pentru orice semnal $x(t)$ mărginit de M_x , există o valoare $M_y < \infty$ care să mărginească semnalul de ieșire.

$$\forall x(t) \text{ a. î. } \|x(t)\| \leq M_x < \infty \Rightarrow \|y(t)\| \leq M_y < \infty$$

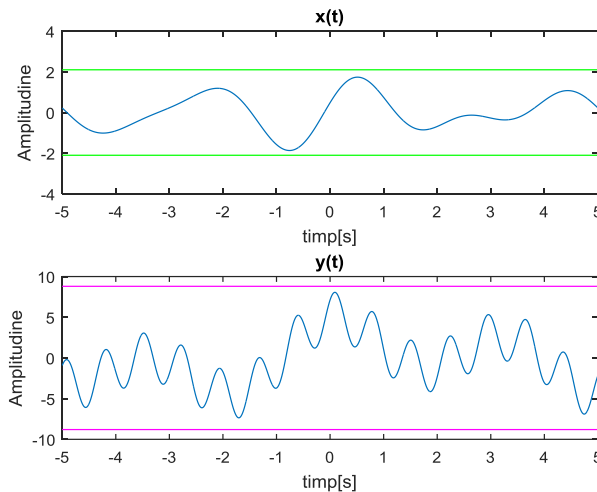


Figura 1. Exemplu de semnal de ieșire mărginit pentru un semnal de intrare mărginit

Cu alte cuvinte, sistemele stabile sunt cele pentru care atunci când intrările rămân în intervalul $[-M_x, M_x]$, ieșirile nu se duc “în bălării” (la $\pm\infty$), adică rămân în intervalul $[-M_y, M_y]$. De remarcat că nu contează dacă M_x este mult mai mic/mare față de M_y , ci doar ca acestea să fie finite. De asemenea, pentru criteriul matematic de stabilitate nu contează ce se întâmplă cu semnalele nemărginite de intrare la trecerea prin sistem.

Testele de stabilitate sunt foarte importante în proiectarea sistemelor, de aceea pe lângă metode de a verifica direct definiția matematică s-au dezvoltat criterii de stabilitate, în special pentru sisteme liniare și invariante în timp. Nu toate criteriile de stabilitate sunt mereu echivalente cu definiția, fiind mai degrabă condiții suficiente, dar nu neapărat și necesare pentru stabilitatea matematică; dacă un astfel de criteriu este îndeplinit, atunci sistemul este stabil BIBO, însă nu toate sistemele stabile BIBO respectă criteriul.

Criterii legate de funcția de transfer $H(s)$

Condiția necesară și suficientă pentru ca un sistem liniar caracterizat de funcția de transfer $H(s) = \frac{\text{polinom}(s)}{(s-s_1)(s-s_2)\dots(s-s_N)}$ să fie stabil este ca toți polii funcției de transfer s_1, s_2, \dots, s_N să fie situați în semiplanul stâng al planului complex. Altfel spus, toți polii funcției de transfer trebuie să aibă partea reală negativă și diferită de zero.

$$\text{Re}\{s_1\} < 0, \text{Re}\{s_2\} < 0, \dots, \text{Re}\{s_N\} < 0$$

Criteriul Hurwitz

Criteriul Hurwitz folosește coeficienții polinomului caracteristic al sistemului (polinomul de la numitor) și determină dacă toate rădăcinile au partea reală negativă, fără a calcula rădăcinile în mod explicit. Acest criteriu se poate aplica doar dacă polinomul caracteristic are toți coeficienții strict pozitivi (> 0).

Fie funcția de transfer $H(s) = \frac{\text{polinom}(s)}{a_N \cdot s^N + a_{N-1} \cdot s^{N-1} + \dots + a_1 \cdot s^1 + a_0}$, pentru care polinomul caracteristic este $P(s) = a_N \cdot s^N + a_{N-1} \cdot s^{N-1} + \dots + a_1 \cdot s^1 + a_0$, $a_k > 0, \forall k$

Matricea Hurwitz, de dimensiune $N \times N$, se completează astfel:

- Pe diagonala principală se scriu coeficienții polinomului caracteristic, începând cu a_{N-1} .

$$M_H = \begin{bmatrix} a_{N-1} & \blacksquare & \blacksquare \\ \blacksquare & \ddots & \blacksquare \\ \blacksquare & \blacksquare & a_0 \end{bmatrix}$$

- Pe fiecare coloană, se completează, pornind de la coeficienții deja completați la pasul anterior, coeficienții corespunzători puterilor lui s mai mari în jos și respectiv corespunzători puterilor lui s mai mici în sus.

$$M_H = \begin{bmatrix} a_{N-1} & a_{N-3} & \dots & 0 \\ a_N & a_{N-2} & \dots & 0 \\ 0 & a_{N-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_0 \end{bmatrix}$$

- Se extrag minorii Hurwitz de-a lungul diagonalei principale.

$$\Delta_1 = |a_{N-1}|, \Delta_2 = \begin{vmatrix} a_{N-1} & a_{N-3} \\ a_N & a_{N-2} \end{vmatrix}, \Delta_3 = \begin{vmatrix} a_{N-1} & a_{N-3} & a_{N-5} \\ a_N & a_{N-2} & a_{N-4} \\ 0 & a_{N-1} & a_{N-3} \end{vmatrix}, \dots, \Delta_N = |M_H|$$

Condiția suficientă de stabilitate a unui sistem este ca toți minorii Hurwitz Δ_k să aibă același semn (să fie pozitivi).

Criteriul Routh

Criteriul Routh folosește tot coeficienții polinomului caracteristic, din care se construiește tabloul:

$$\begin{array}{c|ccc} N & a_N & a_{N-2} & a_{N-4} & \dots \\ N-1 & a_{N-1} & a_{N-3} & a_{N-5} & \dots \\ N-2 & b_1 & b_2 & b_3 & \dots \\ N-3 & c_1 & c_2 & c_3 & \dots \\ \dots & & & & \dots \\ 0 & q & & & \end{array}$$

unde:

$$b_1 = \frac{a_{N-1} \cdot a_{N-2} - a_N \cdot a_{N-3}}{a_{N-1}}, b_2 = \frac{a_{N-1} \cdot a_{N-4} - a_N \cdot a_{N-5}}{a_{N-1}}, \dots,$$

$$c_1 = \frac{b_1 \cdot a_{N-3} - a_{N-1} \cdot b_2}{b_1}, c_2 = \frac{b_1 \cdot a_{N-5} - a_{N-1} \cdot b_3}{b_1}, \dots$$

Numerele din partea stângă nu au decât rolul de a contoriza liniile din tabel, calculele făcându-se exclusiv cu valorile din dreapta; pe măsură ce se înaintează în completarea tabelului, numărul de elemente pe linie scade, până când ajunge la o singură valoare la linia din dreptul valorii “0”.

Coefficienții lipsă de care este nevoie pentru calcule se pun 0.

Sistemul este stabil conform criteriului Routh dacă prima coloană $\{a_N, a_{N-1}, b_1, c_1, \dots, q\}$ conține valori având același semn. Numărul de schimbări de semn dă numărul de rădăcini ale polinomului caracteristic în semiplanul drept.

Criteriul Mihailov

Criteriul Mihailov folosește tot polinomul caracteristic, însă este un criteriu grafic mai degrabă decât un criteriu algebric.

Fie funcția de transfer $H(s) = \frac{\text{polinom}(s)}{a_N \cdot s^N + a_{N-1} \cdot s^{N-1} + \dots + a_1 \cdot s^1 + a_0}$, pentru care polinomul caracteristic este $P(s) = a_N \cdot s^N + a_{N-1} \cdot s^{N-1} + \dots + a_1 \cdot s^1 + a_0$. Polinomul caracteristic se evaluează pe axa frecvențelor:

$$\begin{aligned} P(j\omega) &= P(s)|_{s=j\omega} = a_N \cdot (j\omega)^N + a_{N-1} \cdot (j\omega)^{N-1} + \dots + a_1 \cdot (j\omega)^1 + a_0 \\ &= R(\omega) + j \cdot Q(\omega) \end{aligned}$$

Pentru ca un sistem să fie stabil este suficient ca vectorul complex $P(j\omega)$ să se deplaseze începând din punctul de abscisă $P(j0) = a_0$ situat pe axa reală pozitivă, numai în sens trigonometric, și să parcurgă un număr de cadrane (în ordinea I, II, III, IV, I,...) egal cu ordinul polinomului caracteristic N , atunci când $\omega \in (0, \infty)$. Acest grafic se numește **hodograful Mihailov**.

Desfășurarea lucrării

Aplicație 1. Fie un sistem cu funcția de transfer:

$$H(s) = \frac{1}{s^5 + 5 \cdot s^4 + 10 \cdot s^3 + 10 \cdot s^2 + 5 \cdot s + 1} = \frac{1}{(s+1)^5}.$$

Să se verifice stabilitatea acestui sistem prin criteriile Hurwitz, Routh și Mihailov.

Sistemul are 5 poli, toți în $s = -1$, ceea ce înseamnă că sistemul este stabil conform criteriului Laplace.

Criteriul Hurwitz

Matricea Hurwitz este:

$$M_H = \begin{bmatrix} 5 & 10 & 1 & 0 & 0 \\ 1 & 10 & 5 & 0 & 0 \\ 0 & 5 & 10 & 1 & 0 \\ 0 & 1 & 10 & 5 & 0 \\ 0 & 0 & 5 & 10 & 1 \end{bmatrix}$$

Folosind funcția *Matlab* `det`, se pot calcula determinanții (minorii):

$$\Delta_1 = 5 > 0, \Delta_2 = \begin{vmatrix} 5 & 10 \\ 1 & 10 \end{vmatrix} = 40 > 0$$

$$\Delta_3 = \begin{vmatrix} 5 & 10 & 1 \\ 1 & 10 & 5 \\ 0 & 5 & 10 \end{vmatrix} = 280 > 0$$

$$\Delta_4 = \begin{vmatrix} 5 & 10 & 1 & 0 \\ 1 & 10 & 5 & 0 \\ 0 & 5 & 10 & 1 \\ 0 & 1 & 10 & 5 \end{vmatrix} = 1024 > 0$$

$$\Delta_5 = |M_H| = 1024 > 0$$

Criteriul Routh

Tabelul pentru criteriul Routh este:

$$\begin{array}{c|ccc} 5 & 1 & 10 & 5 \\ 4 & 5 & 10 & 1 \\ 3 & 8 & 4.8 & \\ 2 & 7 & 1 & \\ 1 & 3.6 \dots & 0 & \\ 0 & 1 & & \end{array}$$

Deoarece pe prima coloană toți termenii sunt pozitivi, sistemul este stabil.

Criteriul Mihailov

Pentru criteriul Mihailov trebuie generată reprezentarea grafică:

```
jomega = j* [0:0.01:3.3];  
P = (jomega+1).^5;  
figure(1),  
hold on  
plot([0,0], [-50,50], 'g-');  
plot([-50,400], [0,0], 'g-');  
plot(real(P), imag(P))  
hold off  
axis([-50,400,-50,50])  
title('P(j\omega)')  
grid on
```

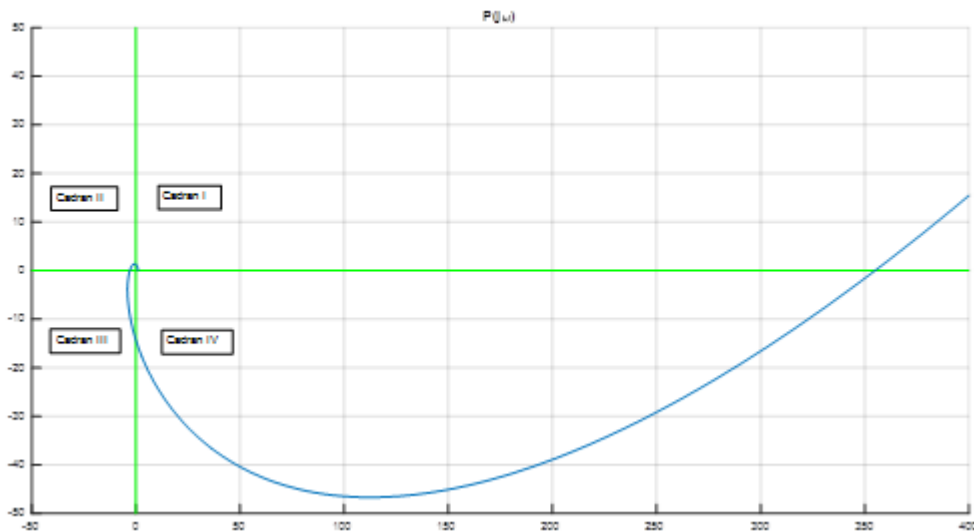


Figura 2. Hodograful Mihailov pentru sistemul din exemplul 1

Graficul pornește de la punctul $1 + j \cdot 0$ (pe axa reală) și parcurge pe rând cadranele I, II, III, IV, I, în total 5 cadrane, de unde rezultă că sistemul este stabil.

Aplicație 2. Fie un sistem cu funcția de transfer $H(s) = \frac{1}{s^3 + s^2 + s^1 + 2}$. Să se verifice stabilitatea acestui sistem prin criteriile Hurwitz, Routh și Mihailov.

Criteriul Hurwitz

Matricea Hurwitz este:

$$M_H = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

$$\Delta_1 = 1 > 0, \Delta_2 = \begin{vmatrix} 1 & 2 \\ 1 & 1 \end{vmatrix} = -1 < 0, \Delta_3 = |M_H| = -2 < 0$$

Sistemul nu este stabil.

Criteriul Routh

Tabelul pentru criteriul Routh este:

$$\begin{array}{c|cc} 3 & 1 & 1 \\ 2 & 1 & 2 \\ 1 & -1 & \\ 0 & 2 & \end{array}$$

Sunt două schimbări de semn, deci două rădăcini au parte reală pozitivă; sistemul nu este stabil.

Criteriul Mihailov

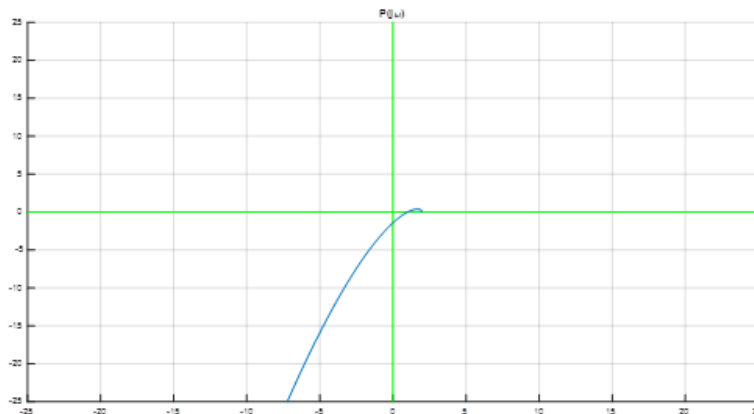


Figura 3. Hodograful Mihailov pentru sistemul din exemplul 2

Graficul merge în sens trigonometric, însă nu parcurge cadranele în ordinea corectă (I, II, III), deci sistemul este instabil.

Intr-adevăr, rădăcinile polinomului caracteristic sunt aproximativ $s_1 \approx -1.35$ și $s_{2,3} \approx 0.17 \pm 1.2j$, rădăcinile complexe fiind cele care generează instabilitatea.

Probleme și aplicații

1. Să se scrie o funcție care primește la intrare vectorul coeficienților polinomului caracteristic și întoarce valorile determinanților minorilor și concluzia dacă sistemul este stabil sau nu.

2. Să se scrie o funcție care primește la intrare vectorul coeficienților polinomului caracteristic și întoarce tabelul conform criteriului Routh și concluzia dacă sistemul este stabil sau nu.

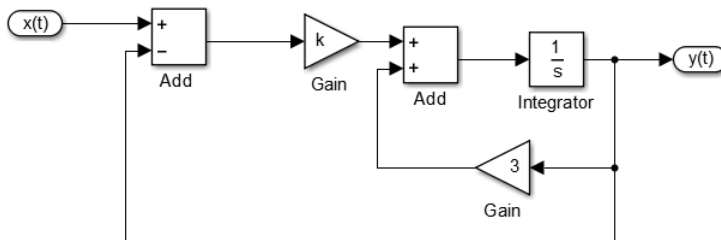
3. Să se verifice stabilitatea sistemelor de mai jos prin criteriile Hurwitz, Routh și Mihailov.

a)
$$H(s) = \frac{s^2 - 1}{s^5 + 4 \cdot s^4 + 7 \cdot s^3 + 3 \cdot s^2 + 8 \cdot s + 1}$$

b)
$$H(s) = \frac{s + 1}{s^4 + s^3 + 5s^2 + s + 1}$$

4. Să se demonstreze că circuitul RLC serie (semnal de intrare – tensiunea de alimentare $u(t)$, semnal de ieșire – curentul prin componente $i(t)$) este stabil pentru orice valori pozitive ale componentelor R, L, C .

5. Să se determine funcția de transfer a sistemului cu reacție de mai jos. Pentru ce valori k ale amplificatorului, sistemul este stabil? Dar instabil?



6. Să se verifice stabilitatea sistemelor din laboratorul *Modelarea sistemelor fizice* pentru cazul în care accelerația gravitațională $g(t)$ este un semnal de intrare (variabil în timp).

Controlul sistemelor și ajustarea parametrilor controller-ului *PID*

Obiective: Construirea unui sistem cu buclă de reacție negativă și controller *PID*; determinarea parametrilor controller-ului *PID* prin metodele *Ziegler-Nichols* și *Cohen-Coon*.

Noțiuni teoretice

Pentru implementarea sistemelor cu control este necesară o schimbare de perspectivă și anume: în analiza sistemelor, problema de bază era determinarea semnalului de ieșire $y(t)$, știindu-se semnalul de intrare $x(t)$ și răspunsul la impuls al sistemului $h(t)$ sau echivalentele lor Laplace $X(s)$ și funcția de transfer $H(s)$.

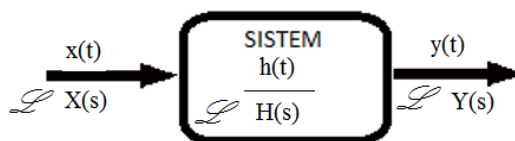


Figura 1. Reprezentarea abstractă a unui sistem

În controlul sistemelor, se cunoaște semnalul dorit la ieșirea sistemului, iar problema constă în a convinge sistemul (numit *Plant*) să producă acest semnal (sau unul cât mai apropiat de el). Semnalul de ieșire al sistemului $y(t)$ devine mărimea controlată.

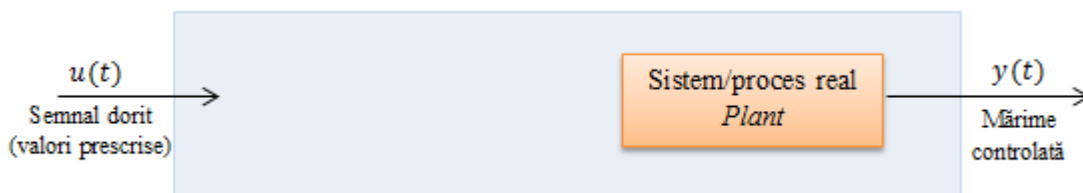


Figura 2. Poziționarea sub-sistemului (*Plant*) într-un sistem cu control

De aceea, sistemul *Plant* este inclus într-un sistem mai mare cu buclă de reacție negativă, pentru a putea determina **semnalul de eroare** $\varepsilon(t)$ între valorile dorite și cele obținute și a genera un **semnal de control** $x(t)$ pentru sistemul *Plant*.

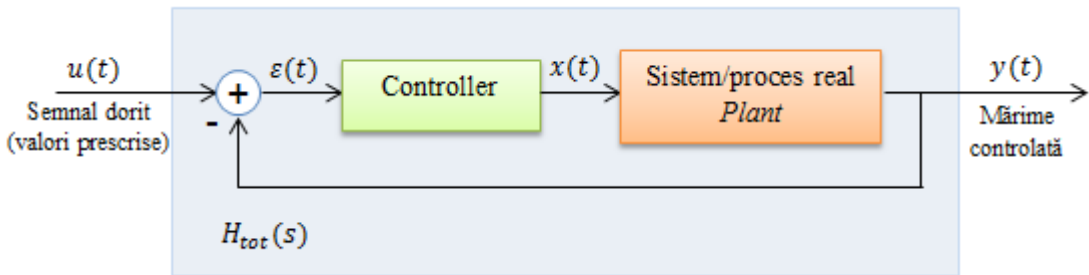
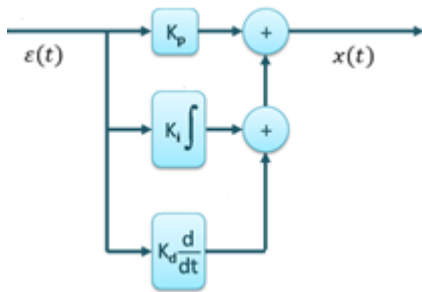


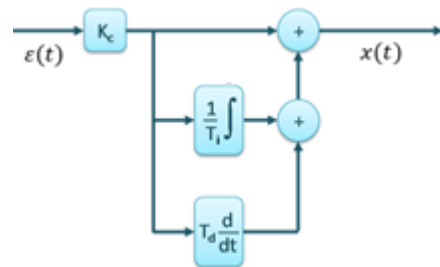
Figura 3. Sistem cu buclă de reacție și controller pe calea directă

Controller-ul *PID* (**P**roportional-**I**ntegral-**D**erivativ), cel mai folosit în industrie datorită simplității lui (doar 3 parametri) și capacității de a genera semnale de comandă adecvate unei game largi de sisteme, poate fi implementat în mai multe moduri, precum cele descrise în figurile *Figura 4.a* și *Figura 4.b*.



$$x(t) = K_p \cdot \varepsilon(t) + K_i \cdot \int \varepsilon(t) dt + K_d \cdot \frac{d\varepsilon(t)}{dt}$$

a) Implementare paralelă



$$x(t) = K_C \cdot \left(\varepsilon(t) + \frac{1}{T_i} \cdot \int \varepsilon(t) dt + T_d \cdot \frac{d\varepsilon(t)}{dt} \right)$$

b) Implementare standard

Figura 4. Implementare *PID*

Deoarece cele două implementări descriu același sistem (anume controller-ul *PID*), legătura între coeficienți se poate determina prin identificare:

$$\begin{cases} K_p = K_C \\ K_i = K_C / T_i \\ K_d = K_C \cdot T_d \end{cases}$$

Determinarea parametrilor controller-ului *PID* prin metoda *Ziegler-Nichols* în buclă închisă

- Se transformă controller-ul *PID* într-un controller *P* pur prin anularea constantelor de integrare ($K_i = 0$) și de derivare ($K_d = 0$).
- Se stabilește factorul de amplificare limită K_0 , care asigură funcționarea auto-oscilantă a sistemului în buclă închisă.
- Se determină perioada de oscilare T_0 pentru acest regim.
- Se ajustează valorile controller-ului *PID* conform parametrilor din *Tabel 1*.

Tabel 1. Parametrii controller-ului *PID* conform metodei *Ziegler-Nichols*

Tip Controller	K_c	T_i	T_d
P	$0.5K_0$	∞	0
PI	$0.45K_0$	$0.83T_0$	0
PID	$0.6K_0$	$0.5T_0$	$0.125T_0$

- În caz de supracreștere (*overshoot*) inacceptabil de mare, se ajustează mai fin parametrii *PID*.

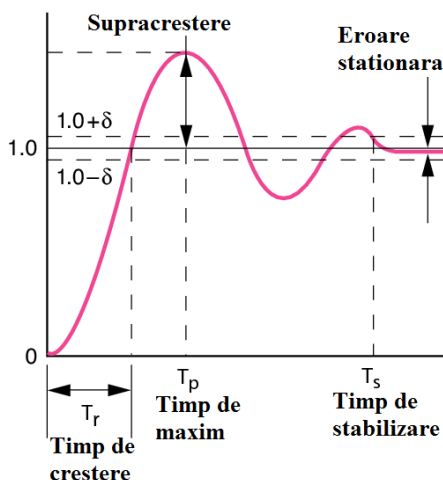


Figura 5. Răspunsul generic al unui sistem cu control la semnalul treaptă unitate pentru evidențierea conceptelor de supracreștere, timp de creștere și timp de stabilizare

Determinarea parametrilor controller-ului *PID* prin metoda *Cohen-Coon* în buclă deschisă

- Se face testul în buclă deschisă, aducând la intrarea sistemului *Plant* un semnal treaptă unitate.

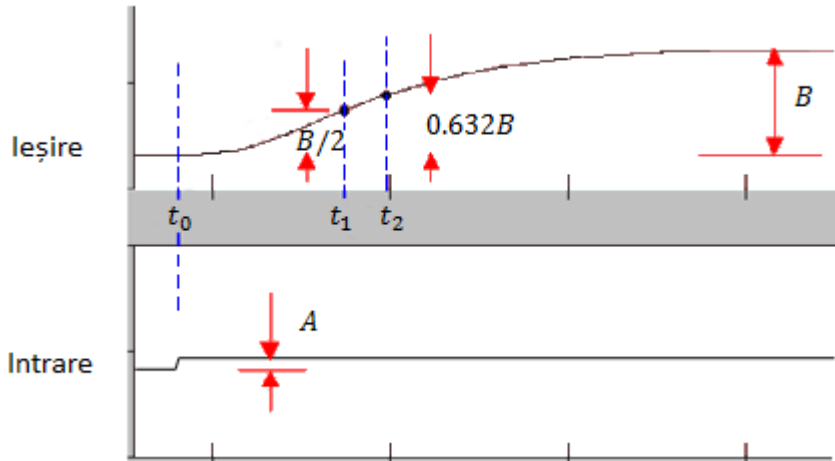


Figura 6. Răspunsul sistemului în buclă deschisă la treapta unitate, pentru determinarea parametrilor de proces conform metodei *Cohen-Coon*

- Se calculează parametrii de proces:

$$t_1 = \frac{t_2 - \ln(2) \cdot t_3}{1 - \ln(2)}$$

$$\tau = t_3 - t_1$$

$$\tau_{del} = t_1 - t_0$$

$$r = \tau_{del} / \tau$$

$$K = B/A$$

- Se ajustează valorile controller-ului *PID* conform parametrilor din *Tabel 2*.

Tabel 2. Parametrii controller-ului *PID* conform metodei *Cohen-Coon*

Tip Controller	K_c	T_i	T_d
P	$\frac{1}{rK} \left(1 + \frac{r}{3}\right)$		
PI	$\frac{1}{rK} \left(0.9 + \frac{r}{12}\right)$	$\tau_{del} \frac{30 + 3r}{9 + 20r}$	
PID	$\frac{1}{rK} \left(\frac{4}{3} + \frac{r}{4}\right)$	$\tau_{del} \frac{32 + 6r}{13 + 8r}$	$\tau_{del} \frac{4}{11 + 2r}$

Desfășurarea lucrării

Aplicație 1. Fie sistemul *Plant* cu funcția de transfer $H(s)$. Să se determine funcția de transfer $H_{tot}(s)$ a întregului sistem cu buclă de reacție negativă și controller cu funcție de transfer $C(s)$.

Pentru a calcula funcția de transfer Laplace a unui sistem, trebuie parcurse următoarele etape:

- scrierea ecuațiilor (diferențiale) care descriu sistemul
- transformarea Laplace a semnalelor și a operațiilor
- prelucrarea ecuațiilor în domeniul Laplace pentru a ajunge la o formă potrivită

Ecuțiile (diferențiale) care descriu sistemul sunt ecuațiile care leagă intrările și ieșirile sub-sistemelor (produsele de convoluție), precum $y(t) = x(t) * h(t)$, și legăturile dintre sub-sisteme, de exemplu $\varepsilon(t) = x(t) - y(t)$.

Atunci când se assemblează sub-sisteme, operațiile de derivare sau integrare sunt rare, iar operațiile obișnuite (înmulțirea cu o constantă, adunarea și scăderea) sunt identice în domeniile *timp* și *Laplace* , așa că de obicei se sare peste etapa scrierii ecuațiilor în timp.

Transformarea Laplace a semnalelor: fie $U(s)$ corespondentul Laplace al semnalului $u(t)$, și așa mai departe pentru toate semnalele:

$$u(t) \stackrel{L}{\leftrightarrow} U(s), \varepsilon(t) \stackrel{L}{\leftrightarrow} E(s), x(t) \stackrel{L}{\leftrightarrow} X(s), y(t) \stackrel{L}{\leftrightarrow} Y(s)$$

Ecuțiile care caracterizează sistemul cu reacție devin în domeniul Laplace:

$$\begin{cases} E(s) = U(s) - Y(s) \\ X(s) = C(s) \cdot E(s) \\ Y(s) = H(s) \cdot X(s) \end{cases}$$

Eliminând semnalele interne, putem obține o relație între semnalul de intrare $U(s)$ și cel de ieșire $Y(s)$:

$$Y(s) = H(s) \cdot X(s) = H(s) \cdot C(s) \cdot E(s) = H(s) \cdot C(s) \cdot (U(s) - Y(s))$$

$$H_{tot}(s) = \frac{Y(s)}{U(s)} = \frac{H(s) \cdot C(s)}{1 + H(s) \cdot C(s)}$$

Aplicație 2. Fie sistemul *Plant* cu funcția de transfer $H(s) = \frac{1}{s \cdot (s+1) \cdot (s+2)}$. Să se determine empiric parametrii controller-ului *PID* prin metoda *Ziegler-Nichols* în buclă închisă. Să se afișeze grafic (*Scope*) răspunsul sistemului cu buclă de reacție (și controller cu parametrii astfel determinați) la semnalul de intrare $u(t)$ treaptă unitate.

Deoarece funcția de transfer este dată în această formă, putem folosi din *Matlab* blocul *Zero-Pole* pentru simularea sistemului *Plant*.

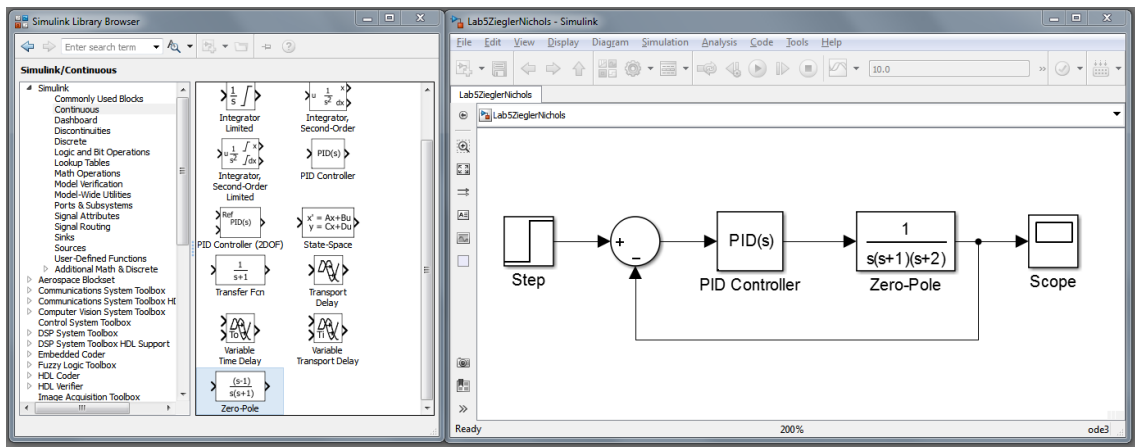


Figura 7. Modelarea *Simulink* a sistemului cu control

Alegând la intrare un semnal treaptă unitate cu trecerea din 0 în 1 la $t = 0s$ și anulând constantele de integrare ($K_i = 0$) și de derivare ($K_d = 0$), se modifică constanta proporțională până se observă funcționarea auto-oscilantă a sistemului.

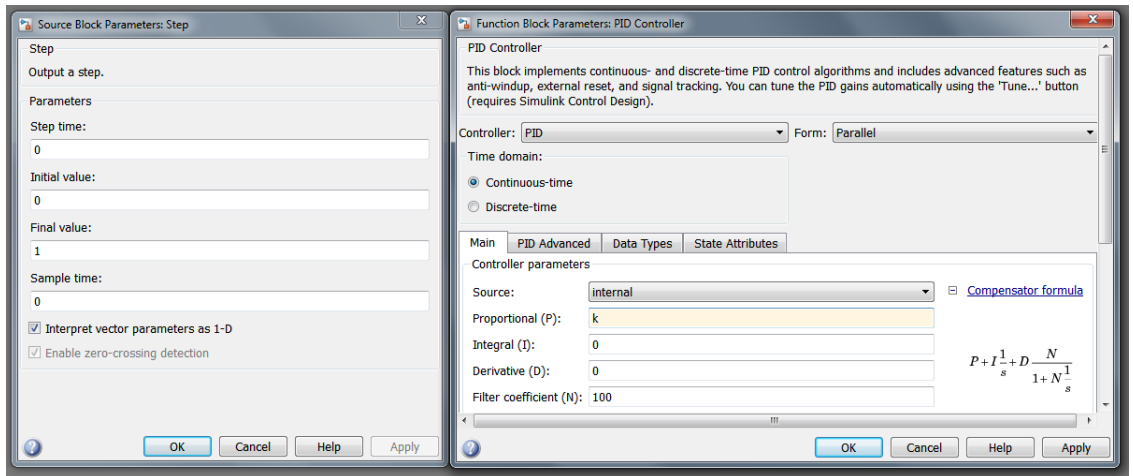


Figura 8. Parametrii blocurilor *Step* și *PID* din *Simulink*

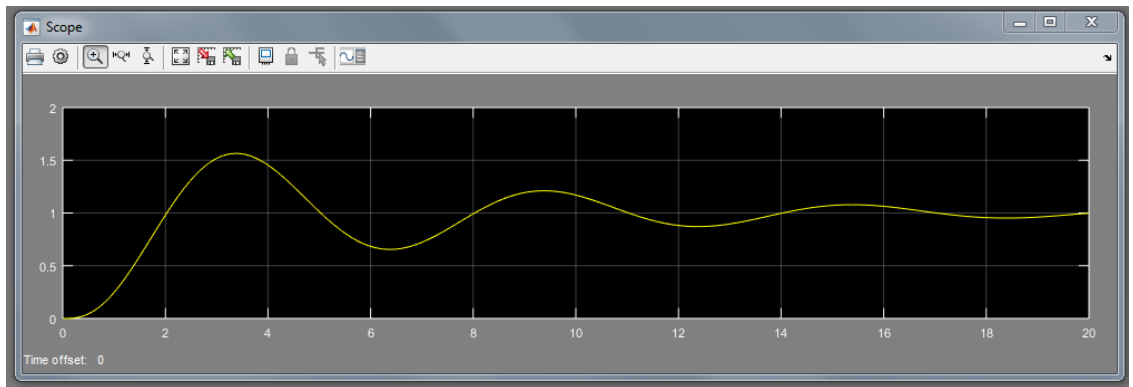


Figura 9. Exemplu de funcționare stabilă a sistemului (constantă proporțională prea mică)

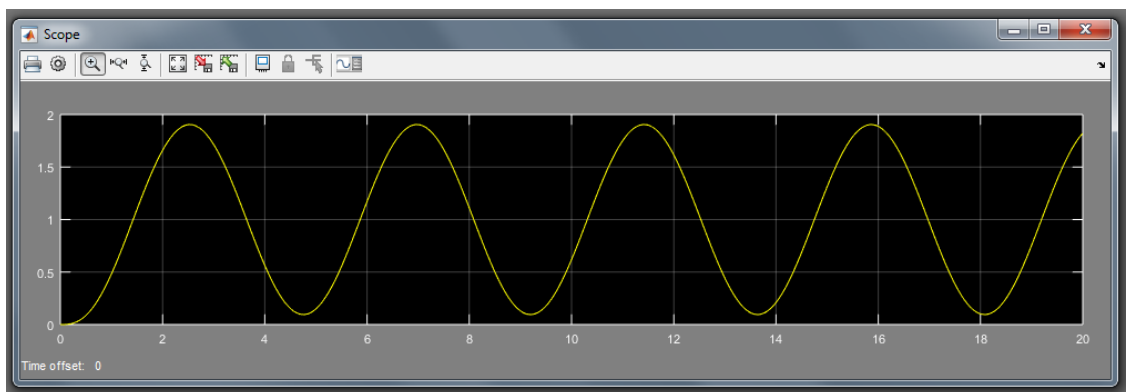


Figura 10. Exemplu de funcționare auto-oscilantă a sistemului (obținută cu K_0 căutat)

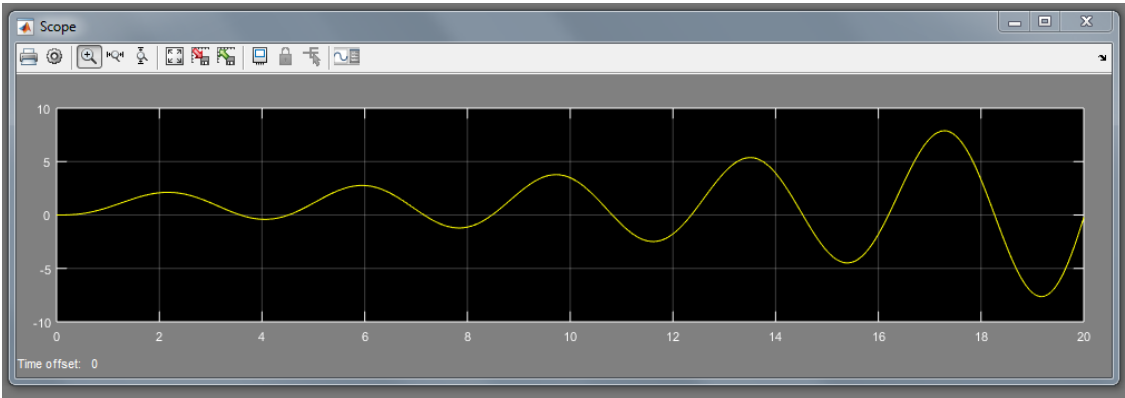


Figura 11. Exemplu de funcționare instabilă a sistemului (constantă proporțională prea mare)

Pe graficul sistemului la limita stabilității se măsoară perioada de oscilare T_0 , iar din *Tabel 1* se citesc parametri K_c, T_i, T_d ai controller-ului *PID*. Pentru că blocul *Simulink* cere parametri K_p, K_i, K_d trebuie făcută conversia. Pentru semnalul de intrare $u(t)$ treaptă unitate, se obține graficul de mai jos, în care se observă că sistemul se stabilizează însă are o supracreștere de ~60%.

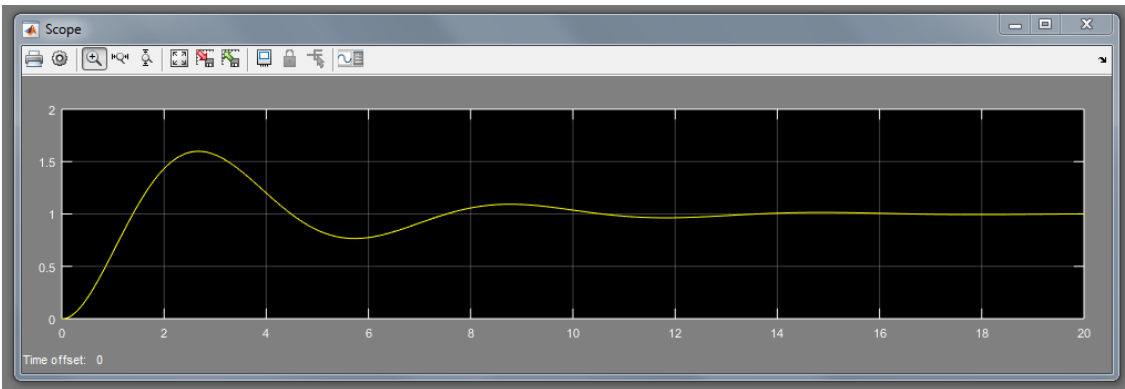


Figura 12. Stabilizarea sistemului la treapta unitate cu coeficienții *Ziegler-Nichols* obținuți

Aplicație 3. Fie sistemul *Plant* cu funcția de transfer $H(s) = \frac{1}{s \cdot (s+1) \cdot (s+2)}$. Să se determine matematic K_0 și T_0 .

Atunci când în controller-ul *PID* se anulează componentele integrală și derivativă, acesta devine un bloc de amplificare (*Gain*).

Atunci când constanta de proporționalitate asigură funcționarea auto-oscilantă a sistemului, funcția de transfer a controllerului este $C(s) = K_0$, iar funcția de transfer a întregului sistem cu buclă de reacție este:

$$H_{tot}(s) = \frac{Y(s)}{U(s)} = \frac{H(s) \cdot C(s)}{1 + H(s) \cdot C(s)} = \frac{K_0}{s^3 + 3 \cdot s^2 + 2 \cdot s + K_0}$$

Putem folosi metodele *Hurwitz* sau *Routh* pentru determinarea valorii K_0 pentru care sistemul ajunge să fie la limita stabilității.

$$Hurwitz = \begin{bmatrix} 3 & K_0 & 0 \\ 1 & 2 & 0 \\ 0 & 3 & K_0 \end{bmatrix}$$

$$\Delta_1 = 3 > 0; \Delta_2 = 6 - K_0 \rightarrow K_0 = 6; \Delta_3 = 6 \cdot K_0 - K_0^2 \rightarrow K_0 = 6 \text{ sau } K_0 = 0$$

Soluția $K_0 = 0$ este soluția banală (care anulează orice semnal de comandă pentru sistemul *Plant*) și poate fi ignorată. Rămâne soluția $K_0 = 6$, pentru care funcția de transfer a sistemului cu buclă de reacție este:

$$H_{K_0}(s) = \frac{6}{s^3 + 3 \cdot s^2 + 2 \cdot s + 6}$$

Criteriul de stabilitate conform diagramei poli-zero-uri este ca polii să se găsească în semiplanul Laplace stâng. Polii din semiplanul drept produc instabilitate în sistem. Deci polii care duc la o funcționare auto-oscilantă trebuie să se găsească pe axa $j\omega$.

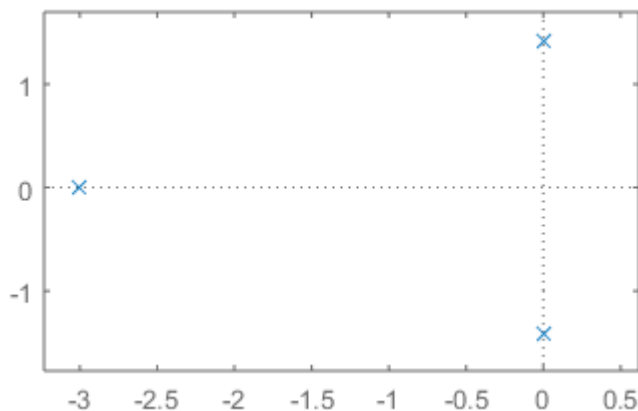


Figura 13. Diagrama poli-zero-uri (axa OX: axa reală; axa OY: axa imaginară)

Aplicație 5. Fie sistemul *Plant* cu funcția de transfer $H(s) = \frac{2}{(s+1) \cdot (s+2)}$. Să se determine empiric parametrii controller-ului *PID* prin metoda *Cohen-Coon* în buclă deschisă.

Pentru determinarea parametrilor controller-ului *PID* prin metoda *Cohen-Coon*, se face testul în buclă deschisă, aducând la intrarea sistemului *Plant* un semnal treaptă unitate cu trecerea din 0 în 1 la $t_0 = 1s$.

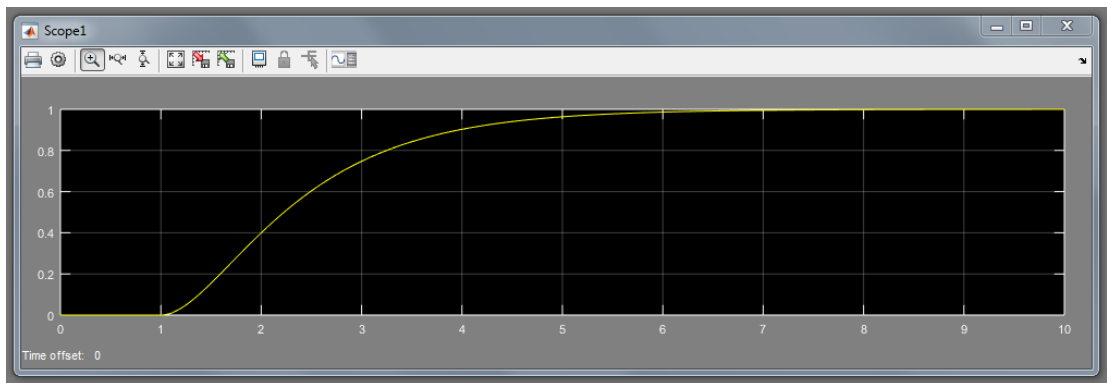


Figura 14. Răspunsul sistemului în buclă deschisă la treapta unitate

Se determină pe grafic $t_2 = 2.23s$, $t_3 = 2.584s$, $B = 1$ și se calculează parametrii de proces:

$$t_1 = \frac{t_2 - \ln(2) \cdot t_3}{1 - \ln(2)} \approx 1.43$$

$$\tau = t_3 - t_1 \approx 1.154$$

$$\tau_{del} = t_1 - t_0 \approx 0.43$$

$$r = \tau_{del} / \tau \approx 0.373$$

$$K = B/A = 1/1 = 1$$

Din *Tabel 2* se citesc parametrii K_c, T_i, T_d ai controller-ului *PID*. Pentru că blocul *Simulink* cere parametrii K_p, K_i, K_d trebuie făcută conversia. Pentru semnalul de intrare $u(t)$ treaptă unitate, se obține:

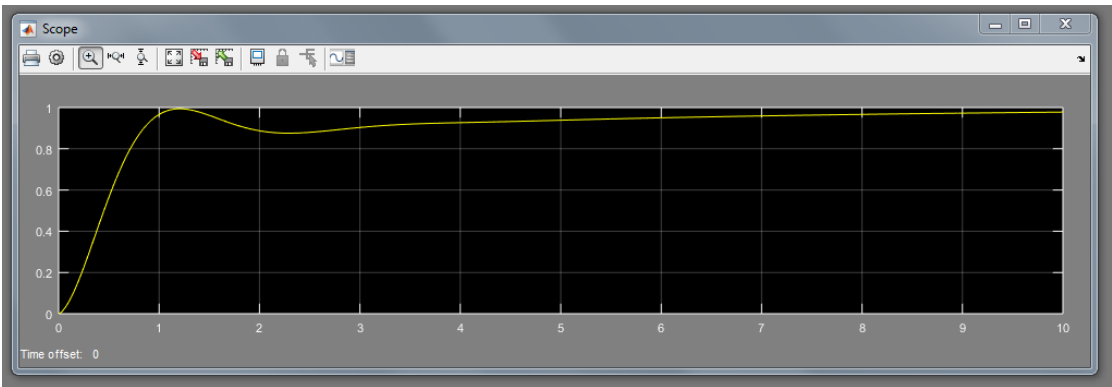


Figura 15. Stabilizarea sistemului la treapta unitate cu coeficienții *Cohen-Coon* obținuți

Aplicație 6. Pornind de la parametri determinați empiric prin metoda *Cohen-Coon*, să se determine parametri optimi în fereastra interactivă *Tune*.

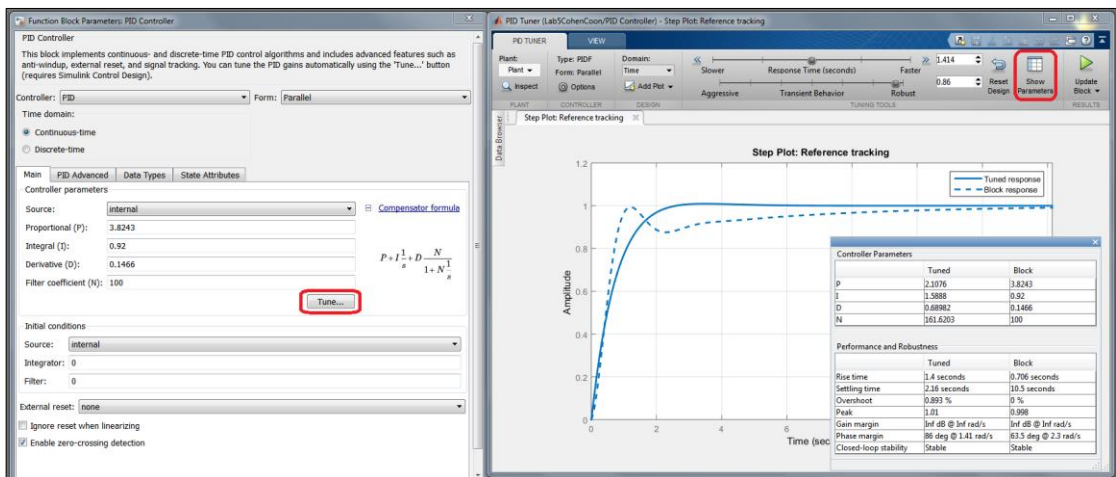


Figura 16. Fereastra interactivă *Tune*

Probleme și aplicații

1. Fie sistemele *Plant* cu funcțiile de transfer de mai jos.

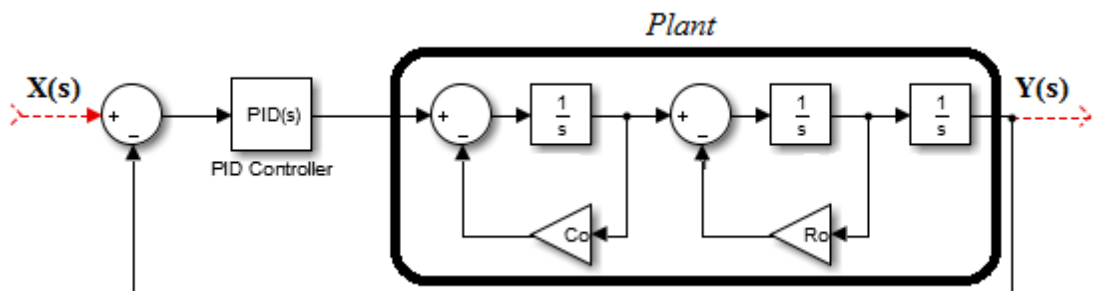
$$H_1(s) = \frac{1}{s^3 + 2 \cdot s^2 + 3 \cdot s}$$

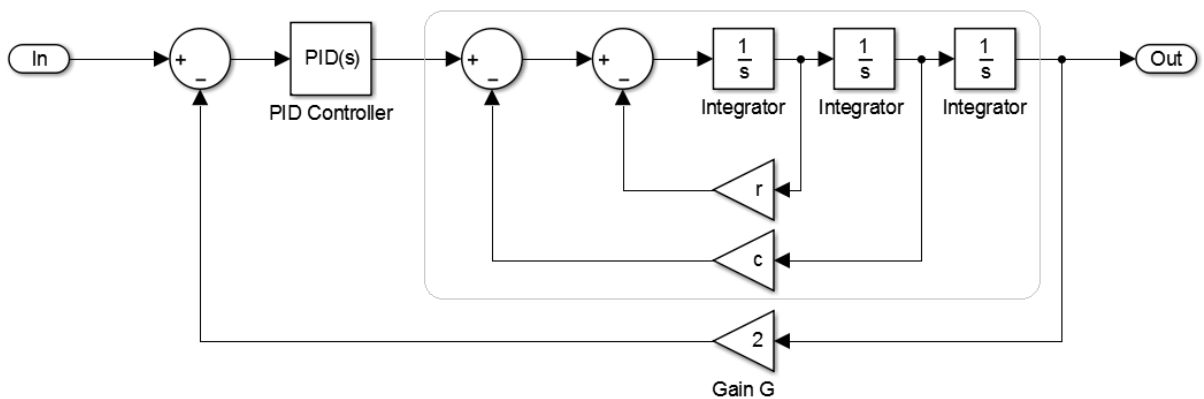
$$H_\tau(s) = \frac{1}{(\tau_1 s + 1) \cdot (\tau_2 s + 1) \cdot (\tau_3 s + 1)}, \quad \text{unde } \tau_1, \tau_2, \tau_3 > 0$$

- a) să se determine empiric și matematic parametrii controller-ului *PID* prin metoda *Ziegler-Nichols* în buclă închisă.
- b) să se afișeze grafic (*Scope*) răspunsul fiecărui sistem cu buclă de reacție (și controller cu parametrii astfel determinați) la semnalul de intrare $u(t)$ treaptă unitate.
- c) să se ajusteze parametrii controller-ului *PID* K_p, K_i, K_d cu ajutorul ferestrei interactive *Tune* astfel încât:
 - supracreșterea să nu fie mai mare de 5%.
 - timpul de stabilizare (pentru $\delta = 0.01$) să fie sub 10s.

Pentru fiecare combinație de parametri încercată, să se noteze într-un tabel similar *Tabelului 3* supracreșterea, timpul de creștere și timpul de stabilizare.

2. Fie sistemele cu reacție și controller *PID* de mai jos.





- a) să se determine empiric și matematic parametrii controller-ului *PID* prin metoda *Ziegler-Nichols* în buclă închisă.
- b) să se afișeze grafic (*Scope*) răspunsul fiecărui sistem cu buclă de reacție (și controller cu parametrii astfel determinați) la semnalul de intrare $u(t)$ treaptă unitate.
- c) să se ajusteze parametrii controller-ului *PID* K_p, K_i, K_d cu ajutorul ferestrei interactive *Tune* astfel încât:
 - supracreșterea să nu fie mai mare de 5%.
 - timpul de stabilizare (pentru $\delta = 0.01$) să fie sub 10s.

Pentru fiecare combinație de parametri încercată, să se noteze într-un tabel similar *Tablelului 3* supracreșterea, timpul de creștere și timpul de stabilizare.

3. Să se determine empiric parametrii controller-ului *PID* prin metoda *Cohen-Coon* în buclă deschisă.

$$H_{\tau}(s) = \frac{1}{(\tau_1 s + 1) \cdot (\tau_2 s + 1) \cdot (\tau_3 s + 1)}, \quad \text{unde } \tau_1, \tau_2, \tau_3 > 0$$

4. Pentru toate sistemele cu buclă de reacție și controller *PID*, să se refacă experimentele după introducerea unui semnal perturbator între controller și *Plant*.

Modelarea sistemelor în timp discret

Obiective: Construirea modelului în timp discret pentru circuite electrice și sisteme fizice.

Noțiuni teoretice

Transformata Z este echivalentul transformatei $Laplace$ pentru semnale discrete.

Tabel 1. Transformata Z și transformata $Laplace$

Transformata $Laplace$	Transformata Z
Aplicată semnalelor în timp continuu: $x(t): R^+ \rightarrow R$	Aplicată semnalelor în timp discret. $x_d(n): N \rightarrow R$
$X(s) = \int_0^{\infty} x(t) \cdot e^{-st} dt$	$X(z) = \sum_{n=0}^{\infty} x_d(n) \cdot z^{-n}$

Dacă semnalul discret se obține prin eșantionarea semnalului continuu cu frecvența de eșantionare F_s (și cuantizare), atunci:

$$x_d(n) = x(n \cdot T_s), \quad n \in N, \quad T_s = 1/F_s$$

Tabel 2. Transformarea Z a operațiilor

Operație în timp		Operație în Z	
Inmulțire cu constantă	$k \cdot x_d(n)$	$k \cdot X(z)$	Inmulțire cu constantă
Adunare/Scădere	$x_d(n) \pm y_d(n)$	$X(z) \pm Y(z)$	Adunare/Scădere
Intârziere	$x_d(n - k)$	$X(z) \cdot z^{-k}$	Inmulțire cu z^{-1}
Convoluție	$x_d(n) * h_d(n)$	$X(z) \cdot H(z)$	Produs

Pentru a calcula funcția de transfer Z a unui sistem direct din ecuațiile diferențiale care descriu sistemul, trebuie parcurse următoarele etape:

- Scrierea ecuațiilor diferențiale care descriu sistemul
- Discretizarea semnalelor și a operațiilor
- Transformarea Z a semnalelor și a operațiilor
- Prelucrarea ecuațiilor în domeniul Z pentru a ajunge la o formă potrivită

Pentru a calcula funcția de transfer Z a unui sistem pornind de la funcția de transfer Laplace, trebuie făcută transformarea din domeniul Laplace s în domeniul z . Legătura dintre variabila Laplace s și variabila z este:

$$z = e^{sT_s} \Leftrightarrow s = F_s \cdot \ln(z) \quad (1)$$

Pentru a se evita folosirea logaritmilor se preferă aproximări precum

- *metodele Euler:*

$$\text{(Forward)} \quad s \approx \frac{z-1}{T_s} \quad \text{(Backward)} \quad s \approx \frac{1}{T_s} \cdot \frac{z-1}{z}$$

- *metoda Tustin:*

$$s \approx \frac{2}{T_s} \cdot \frac{z-1}{z+1} \quad (2)$$

Regiunea de stabilitate (corespunzătoare semiplanului stâng al diagramei *Laplace*) este transferată astfel pe diagrama Z :

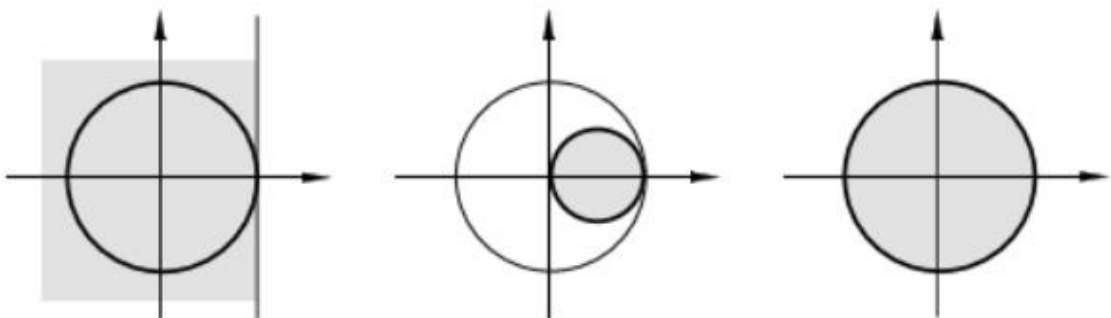


Figura 1. Maparea regiunii de stabilitate a diagramei *Laplace* pe diagrama Z conform metodelor *Euler Forward* (stânga), *Euler Backward* (centru) și *Tustin* (dreapta)

În toate metode de modelare, este util ca funcția de transfer $H(z)$ să fie scrisă sub forma unui raport de polinoame în variabila z^{-1} :

$$H(z) = \frac{b_0 + b_1z^{-1} + \dots + b_Nz^{-N}}{1 - a_1z^{-1} - \dots - a_Mz^{-M}} = \frac{\sum_{k=0}^N b_kz^{-k}}{1 - \sum_{k=1}^M a_kz^{-k}} \quad (3)$$

- atunci când $a_k = 0 \quad \forall k$, se obține un sistem (sau filtru) FIR (cu răspuns finit la impuls).
- dacă există vreun $a_k \neq 0$, se obține un sistem (sau filtru) IIR (cu răspuns infinit la impuls).

Implementarea sistemului discret se poate face cu blocuri de întârziere (*delay*) și de amplificare (*gain*).

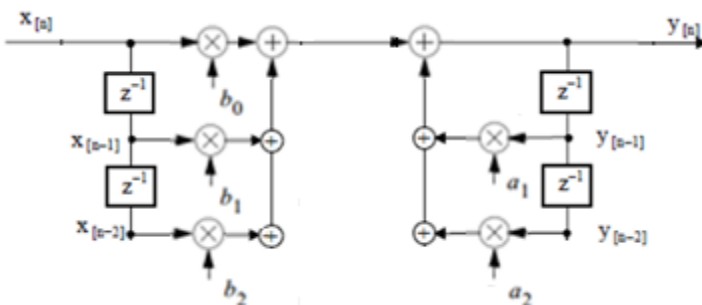


Figura 2. Implementarea sistemelor discrete în forma directă I

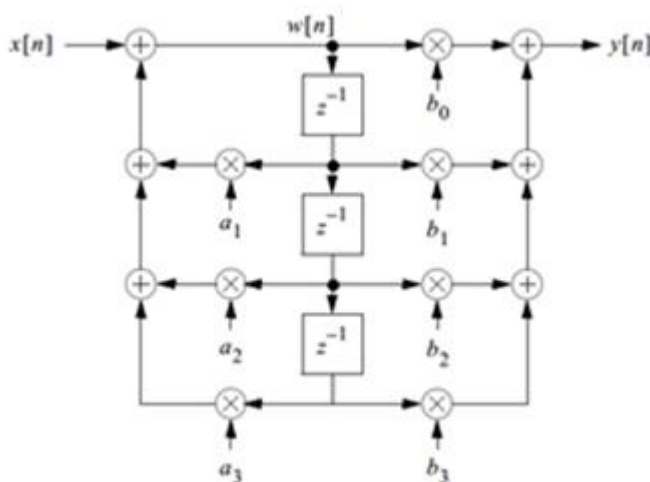
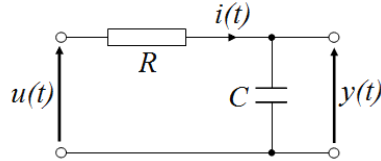


Figura 3. Implementarea sistemelor discrete în forma directă II

Desfășurarea lucrării

Aplicație 1. Să se scrie ecuațiile diferențiale care descriu circuitul RC serie din imagine.



$$\begin{cases} u(t) = R \cdot i(t) + y(t) \\ i(t) = C \cdot \frac{dy(t)}{dt} \end{cases}$$

Aplicație 2. Să se determine funcția de transfer Laplace $H(s)$.

$$u(t) = R \cdot C \cdot \frac{dy(t)}{dt} + y(t)$$

$$U(s) = s \cdot R \cdot C \cdot Y(s) + Y(s)$$

$$H(s) = \frac{Y(s)}{U(s)} = \frac{1}{s \cdot R \cdot C + 1}$$

Aplicație 3. Să se determine funcția de transfer $H_1(z)$ a modelului discret direct din ecuațiile diferențiale.

Transformarea Z a semnalelor: fie $U(z)$ corespondentul Z al semnalului $u_d(n)$ și $Y(z)$ corespondentul Z al semnalului $y_d(n)$:

$$u_d(n) \stackrel{Z}{\leftrightarrow} U(z)$$

$$y_d(n) \stackrel{Z}{\leftrightarrow} Y(z)$$

Primul pas în determinarea funcției de transfer Z este discretizarea (prin eșantionare și cuantizare) a:

- semnalelor

$$u_d(n) = u(n \cdot T_s)$$

$$i_d(n) = i(n \cdot T_s)$$

$$y_d(n) = y(n \cdot T_s)$$

- operațiilor

$$R \cdot i(t) \rightarrow R \cdot i_d(n)$$

$$\frac{dy(t)}{dt} \rightarrow \frac{y_d(n) - y_d(n-1)}{T_s}$$

Pentru aproximarea derivatei s-a folosit formula derivatei numerice în două puncte la stânga.

Sistemul de ecuații devine așadar:

$$\begin{cases} u_d(n) = R \cdot i_d(n) + y_d(n) \\ i_d(n) = C \cdot \frac{y_d(n) - y_d(n-1)}{T_s} \end{cases}$$

Sau eliminând semnalul intern $i_d(n)$:

$$u_d(n) = R \cdot C \cdot \frac{y_d(n) - y_d(n-1)}{T_s} + y_d(n)$$

Ecuția care caracterizează sistemul devine:

$$U(z) = R \cdot C \cdot \frac{Y(z) - Y(z) \cdot z^{-1}}{T_s} + Y(z)$$

$$H_1(z) = \frac{Y(z)}{U(z)} = \frac{1}{\left(\frac{R \cdot C}{T_s} + 1\right) - \frac{R \cdot C}{T_s} z^{-1}}$$

Aplicație 4. Să se determine funcția de transfer $H_2(z)$ a modelului discret pornind de la funcția de transfer Laplace, prin aproximarea *Tustin*.

- funcția de transfer Laplace este:

$$H(s) = \frac{1}{s \cdot R \cdot C + 1}$$

- aproximarea Tustin este:

$$s \approx \frac{2}{T_s} \cdot \frac{z - 1}{z + 1}$$

Rezultă funcția de transfer Z:

$$H_2(z) = H(s) \Big|_{s=\frac{2}{T_s} \frac{z-1}{z+1}} = \frac{1}{\frac{2}{T_s} \cdot \frac{z-1}{z+1} \cdot R \cdot C + 1}$$

$$H_2(z) = \frac{z + 1}{\frac{2}{T_s} \cdot R \cdot C \cdot (z - 1) + z + 1}$$

$$H_2(z) = \frac{z + 1}{\left(\frac{2 \cdot R \cdot C}{T_s} + 1\right) \cdot z - \left(\frac{2 \cdot R \cdot C}{T_s} - 1\right)}$$

$$H_2(z) = \frac{1 + z^{-1}}{\left(\frac{2 \cdot R \cdot C}{T_s} + 1\right) - \left(\frac{2 \cdot R \cdot C}{T_s} - 1\right) \cdot z^{-1}}$$

Aplicație 5. Să se modeleze sistemul folosind cele 3 funcții de transfer și să se compare ieșirile, pentru $R = 4\text{k}\Omega$, $C = 150\mu\text{F}$, $F_s = 10\text{Hz}$.

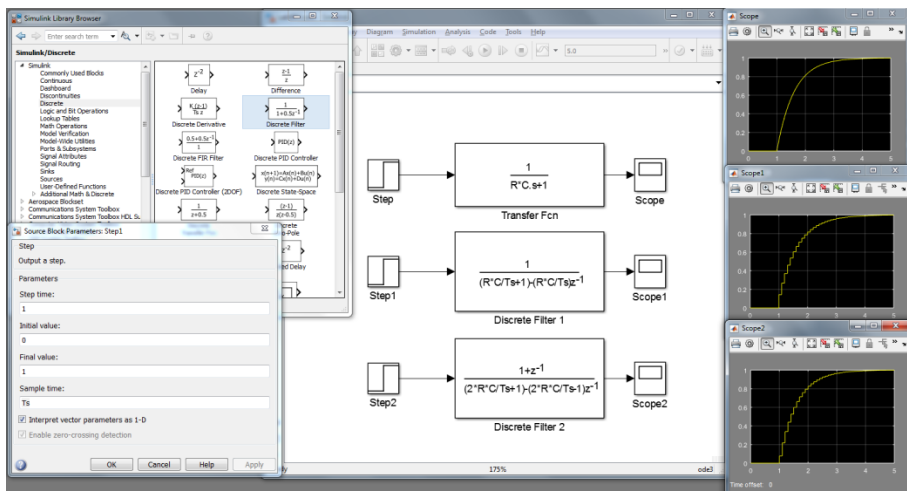


Figura 4. Modelarea Simulink a sistemului discret

Probleme și aplicații

1. Să se realizeze modelul circuitului RLC serie ($R = 20\Omega$, $L = 100\text{mH}$, $C = 10\mu\text{F}$, $v_{in}(t) = \sin(2\pi ft + \varphi_0)$, $F = 3\text{Hz}$, $\varphi_0 = \pi/4$) folosind:

a) funcția de transfer Laplace $H(s)$

b) funcția de transfer $H_1(z)$ determinată direct din ecuațiile diferențiale ($F_s = 70\text{Hz}$)

c) funcția de transfer $H_2(z)$ determinată prin aproximarea *Tustin* ($F_s = 70\text{Hz}$)

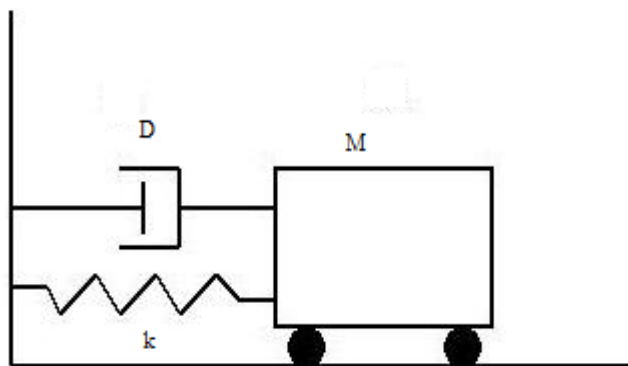
Să se afișeze (*Scope*) tensiunea pe condensator, cea pe bobină și cea pe rezistență pentru cele 3 metode de modelare.

2. Un corp de masă M legat de un resort ideal este așezat în poziția de echilibru pe o suprafață perfect orizontală și netedă (fără frecări). Asupra corpului se acționează cu o forță orizontală $F(t)$ variabilă în timp. Se cunosc $m = 5\text{ kg}$, $k = 11\frac{\text{N}}{\text{m}}$, $D = 2\frac{\text{Ns}}{\text{m}}$, $F(t) = \text{sign}(\sin(2\pi ft + \varphi_0))$, $f = 3\text{Hz}$, $\varphi_0 = \pi/4$. Să se realizeze modelul sistemului folosind:

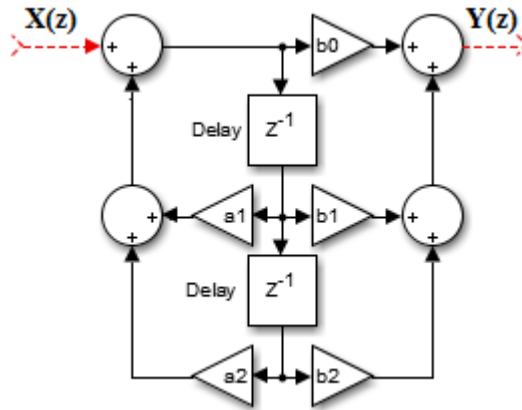
a) funcția de transfer Laplace $H(s)$

b) funcția de transfer $H_1(z)$ determinată direct din ecuațiile diferențiale ($F_s = 70\text{Hz}$)

c) funcția de transfer $H_2(z)$ determinată prin aproximarea *Tustin* ($F_s = 70\text{Hz}$)



3. Fie sistemul discret $H(z) = \frac{2(z+z_0)(z+z_0^*)}{2z^2-z+1}$, unde $z_0 = 1 + 3j$ și $z_0^* = 1 - 3j$
- a) să se construiască diagrama *poli-zerouri* pentru $H(z)$
- b) să se determine coeficienții a_i și b_i pentru implementarea sistemului ca mai jos:



4. Fie un sistem cu funcția de transfer:

$$H(s) = \frac{s^3 - 8}{s^3 + 8}$$

- a) să se determine sistemul discret corespondent $H(z)$ obținut cu ajutorul aproximării *Tustin*, pentru o perioadă de eșantionare $T_s = 1$ secundă.
- b) să se reprezinte diagrama *poli-zerouri* (Z) pentru sistemul discret corespondent.
- c) să se deseneze implementarea în forma directă I și II a sistemului discret.
- d) să se modeleze sistemul folosind funcțiile de transfer *Laplace* și Z și să se compare ieșirile.

5. Fie un sistem cu funcția de transfer:

$$H(s) = \frac{s - 1}{s^2 + s + 1}$$

- a) să se găsească sistemele discrete corespondente cu ajutorul funcției *Matlab* `c2d`, folosind toate metodele suportate și $T_s = 0.01$.
- b) să se modeleze toate aceste sisteme discrete și să se compare ieșirile.

Controller-ul *Deadbeat*

Obiective: Determinarea funcției de transfer $H(z)$ a controller-ului *deadbeat* pentru un sistem digital.

Noțiuni teoretice

Spre deosebire de controllerele discrete a căror proiectare este o extensie a metodelor de proiectare în timp continuu, precum controller-ul *PID* discret, controller-ul *deadbeat* este specific sistemelor discrete și nu are echivalent analogic. De asemenea, controller-ul *deadbeat* nu are o formă generală, el depinde de sub-sistemul care trebuie controlat (*Plant*).

Constrângerile de proiectare ale controller-ului *deadbeat* sunt:

- Eroare de poziționare 0
- Timpul în care se ajunge la valoarea dorită să fie finit și cât mai scurt
- Sistemul să fie realizabil (cauzal)

Observație: Controller-ul *deadbeat* este specific controlului sistemelor discrete, însă dacă este folosit în sisteme mixte, în care să controleze sisteme *Plant* continue, controller-ul *deadbeat* asigură valorile dorite doar la momente de timp multipli ai perioadei de eșantionare; însă nu poate asigura că între aceste momente nu există ripluri în răspunsul continuu al sistemului.

Fie sistemul discret cu reacție de mai jos, format dintr-un subsistem *Plant* (cu funcție de transfer $P(z)$ cunoscută, având toți polii și zerourile în interiorul cercului unitate), un controller (care va fi proiectat) și bucla de reacție negativă.

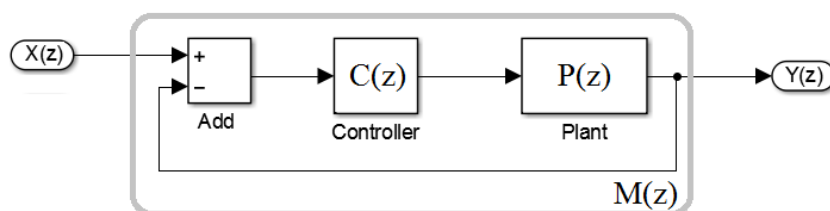


Figura 1. Sistem digital cu reacție

Pentru ca sistemul cu reacție să genereze un semnal de ieșire $y(n)$ identic (dar întârziat) cu semnalul de intrare $x(n)$, ar trebui ca:

$$y(n) = x(n - k) \xrightarrow{\text{Transf. } Z} \quad (1)$$

$$Y(z) = z^{-k} \cdot X(z) \quad (2)$$

$$M(z) = \frac{Y(z)}{X(z)} = z^{-k} = \frac{1}{z^k} \quad (3)$$

Funcția de transfer a întregului sistem este:

$$M(z) = \frac{C(z) \cdot P(z)}{1 + C(z) \cdot P(z)} \quad (4)$$

Controller-ul ar trebui așadar să aibă funcția de transfer:

$$C(z) = \frac{1}{P(z)} \cdot \frac{M(z)}{1 - M(z)} = \frac{1}{P(z)} \cdot \frac{1}{z^k - 1} \quad (5)$$

Alegerea valorii k se face din constrângerea de realizabilitate. Pentru ca un sistem discret cu funcția de transfer $H(z) = \frac{\text{numarator}(z)}{\text{numitor}(z)}$ să fie cauzal, trebuie ca ordinul numitorului să fie cel puțin egal cu ordinul numărătorului. În cazul controller-ului *deadbeat* având funcția de transfer $C(z)$, k trebuie să acopere diferența între ordinele numitorului și numărătorului funcției de transfer a sub-sistemului *Plant*.

$$k \geq \text{ord}(\text{numitor}(P(z))) - \text{ord}(\text{numarator}(P(z))) \quad (6)$$

Pentru obținerea celui mai rapid răspuns, se preferă egalitatea în relația de mai sus.

Dacă se dorește răspunsul corect al sistemului la o rampă (față de treapta unitate considerată mai sus), funcția de transfer a sistemului trebuie să fie:

$$M(z) = \frac{(k + 1) \cdot z - k}{z^{k+1}} \Rightarrow \quad (7)$$

$$C(z) = \frac{1}{P(z)} \cdot \frac{M(z)}{1 - M(z)} = \frac{1}{P(z)} \cdot \frac{(k + 1) \cdot z - k}{z^{k+1} - (k + 1) \cdot z + k} \quad (8)$$

Parametrul k are aceeași valoare și semnificație ca mai sus.

În cazul în care sub-sistemul *Plant* conține poli sau zerouri în afara cercului unitate, aceștia trebuie avuți în vedere când se construiește funcția de transfer dorită pentru sistemul cu reacție.

Fie ca sistemul să aibă K zerouri (z_i) și L poli (p_j) în afara cercului unitate, iar polii și zerourile din interiorul cercului unitate să fie grupați în $B(z)$:

$$P(z) = \frac{\prod_{i=1}^K (1 - z_i \cdot z^{-1})}{\prod_{j=1}^L (1 - p_j \cdot z^{-1})} \cdot B(z) \quad (9)$$

Funcția de transfer a controller-ului este în continuare:

$$C(z) = \frac{1}{P(z)} \cdot \frac{M(z)}{1 - M(z)} = \frac{\prod_{j=1}^L (1 - p_j \cdot z^{-1})}{\prod_{i=1}^K (1 - z_i \cdot z^{-1})} \cdot \frac{1}{B(z)} \cdot \frac{M(z)}{1 - M(z)} \quad (10)$$

Însă polii și zerourile din afara cercului unitate trebuie anulați/anulate cu ajutorul funcției de transfer globale $M(z)$. Astfel, pentru a anula zerourile, funcția de transfer globală trebuie să fie de forma:

$$M(z) = \prod_{i=1}^K (1 - z_i \cdot z^{-1}) \cdot (m_n \cdot z^{-n} + m_{n+1} \cdot z^{-n-1} + \dots) \quad \text{unde } n \geq k \quad (11)$$

De asemenea, pentru a anula polii, trebuie ca:

$$1 - M(z) = \prod_{j=1}^L (1 - p_j \cdot z^{-1}) \cdot (1 - z^{-1})^p \cdot (1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots) \quad (12)$$

unde p este numărul de poli în $z = 1$ din $P(z)$ sau din $X(z)$; se alege valoarea mai mare.

Pentru treapta unitate, $X(z) = \frac{1}{1-z^{-1}}$, deci p va avea valoarea 1 dacă $P(z)$ nu conține mai mulți poli în $z = 1$.

Trunchierea sumelor se face astfel încât $M(z)$ și $1 - M(z)$ să aibă același ordin și numărul de necunoscute să fie egal cu ordinul lui $M(z)$, pentru a se putea forma un sistem determinat de ecuații din care să se calculeze coeficienții.

Desfășurarea lucrării

Aplicație 1. Fie sistemul din *Figura 1* cu funcția de transfer a sub-sistemului *Plant*.

$$P(z) = \frac{z + 0.6}{3 \cdot z^2 - z - 1}$$

Să se determine funcția de transfer a controller-ului *deadbeat* și să se modeleze sistemul cu reacție în *Simulink* (cu pasul de eșantionare $T_s = 0.01$ și semnal de intrare treaptă unitate întârziată cu 0.1s, vezi *Figura 2*).

În primul rând se observa că atât poliile cât și zerourile sistemului *Plant* sunt în interiorul cercului unitate.

$$\text{zero: } z_0 = -0.6$$

$$\text{poli: } z_{1,2} = \frac{1 \pm \sqrt{13}}{2 \cdot 3}$$

Diferența între ordinele numitorului și numărătorului funcției de transfer a sub-sistemului *Plant* este:

$$k = 2 - 1 = 1.$$

Așadar funcția de transfer a controller-ului este:

$$M(z) = \frac{1}{z^k} = \frac{1}{z} \Rightarrow C(z) = \frac{1}{P(z)} \cdot \frac{M(z)}{1 - M(z)} = \frac{3 \cdot z^2 - z - 1}{(z + 0.6) \cdot (z - 1)} = \frac{3 \cdot z^2 - z - 1}{z^2 - 0.4z - 0.6}$$

Pentru implementarea în *Simulink* trebuie avut grijă ca pasul de eșantionare să fie respectat de fiecare bloc. Răspunsul sistemului este treapta unitate întârziată cu o perioadă de eșantionare.

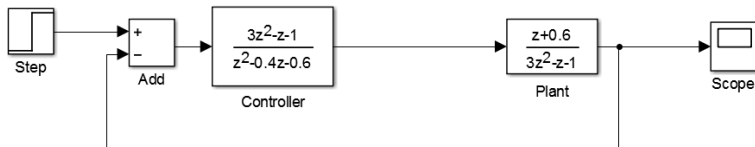


Figura 1. Implementarea în *Simulink* a sistemului din *Aplicație 1*

Dacă s-ar fi cerut funcția de transfer a controller-ului care să genereze un răspuns perfect (după un număr de tacte) pentru semnalul de intrare rampă, aceasta ar fi fost:

$$C(z) = \frac{1}{P(z)} \cdot \frac{M(z)}{1 - M(z)} = \frac{3 \cdot z^2 - z - 1}{(z + 0.6)} \cdot \frac{2 \cdot z - 1}{z^2 - 2 \cdot z + 1}$$

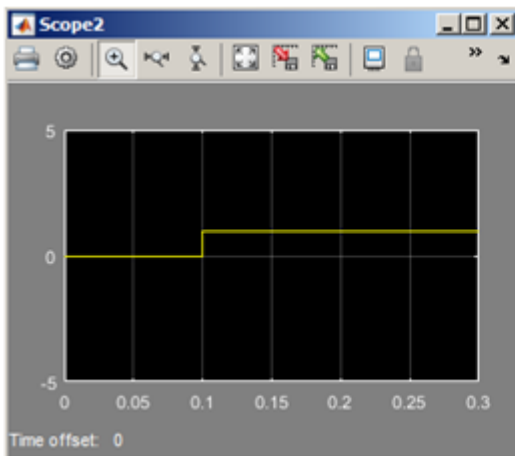


Figura 2. Treaptă unitate întârziată cu 0.1s

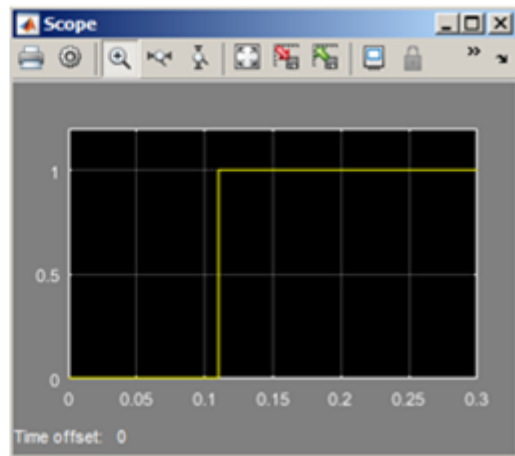


Figura 3. Răspunsul sistemului din *Aplicație 1* la semnalul treaptă unitate (întârziat cu 0.1s)

Aplicație 2. Fie sistemul din *Figura 1* cu funcția de transfer a sub-sistemului *Plant*.

$$P(z) = \frac{(z + 0.2) \cdot (z + 3)}{z \cdot (z - 1) \cdot (z - 0.4) \cdot (z - 0.8)} = \frac{z^{-2} \cdot (1 + 0.2 \cdot z^{-1}) \cdot (1 + 3 \cdot z^{-1})}{(1 - z^{-1}) \cdot (1 - 0.4 \cdot z^{-1}) \cdot (1 - 0.8 \cdot z^{-1})}$$

Să se determine funcția de transfer a controller-ului *deadbeat* și să se modeleze sistemul cu reacție în *Simulink* (cu pasul de eșantionare $T_s = 0.01$ și semnal de intrare treapta unitate întârziată cu 0.1s, vezi *Figura 2*).

$P(z)$ conține un zero în -3 , deci $M(z)$ trebuie să aibă termenul $(1 + 3 \cdot z^{-1})$.

Atât $P(z)$ cât și $X(z)$ conțin un singur pol în $z = 1$, deci $1 - M(z)$ trebuie să aibă termenul $(1 - z^{-1})^1$.

Diferența între ordinea numitorului și numărătorului funcției de transfer a sub-sistemului *Plant* este:

$$k = 4 - 2 = 2$$

Deoarece ordinul lui $M(z)$ este 3 și $k = 2$, există o singură necunoscută m și două necunoscute a :

$$\begin{cases} M(z) = (1 + 3 \cdot z^{-1}) \cdot m_2 \cdot z^{-2} \\ 1 - M(z) = (1 - z^{-1}) \cdot (1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}) \end{cases}$$

$$\begin{aligned}
1 - (1 + 3 \cdot z^{-1}) \cdot m_2 \cdot z^{-2} &= (1 - z^{-1}) \cdot (1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}) \\
1 - m_2 \cdot z^{-2} - 3 \cdot m_2 \cdot z^{-3} &= 1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} - z^{-1} - a_1 \cdot z^{-2} - a_2 \cdot z^{-3} \\
1 - m_2 \cdot z^{-2} - 3 \cdot m_2 \cdot z^{-3} &= 1 + (a_1 - 1) \cdot z^{-1} + (a_2 - a_1) \cdot z^{-2} - a_2 \cdot z^{-3} \\
\begin{cases} 0 = a_1 - 1 \\ m_2 = a_1 - a_2 \\ 3 \cdot m_2 = a_2 \end{cases} &\Rightarrow \begin{cases} a_1 = 1 \\ a_2 = 3/4 = 0.75 \\ m_2 = 1/4 = 0.25 \end{cases}
\end{aligned}$$

Funcția de transfer a controller-ului este așadar:

$$\begin{aligned}
C(z) &= \frac{(1 - z^{-1})}{(1 + 3 \cdot z^{-1})} \cdot \frac{(1 - 0.4 \cdot z^{-1}) \cdot (1 - 0.8 \cdot z^{-1})}{z^{-2} \cdot (1 + 0.2 \cdot z^{-1})} \cdot \frac{(1 + 3 \cdot z^{-1}) \cdot 0.25 \cdot z^{-2}}{(1 - z^{-1}) \cdot (1 + z^{-1} + 0.75 \cdot z^{-2})} \\
C(z) &= \frac{0.25 \cdot z \cdot (z - 0.4) \cdot (z - 0.8)}{(z + 0.2) \cdot (z^2 + z + 0.75)}
\end{aligned}$$

Pentru implementarea în *Simulink* trebuie avut grijă ca pasul de eșantionare să fie respectat de fiecare bloc. Răspunsul sistemului este semnalul treaptă unitate întârziat cu trei perioade de eșantionare. Ar fi fost întârziat cu doar două perioade de eșantionare dacă nu ar fi avut poli și zerouri pe sau în afara cercului unitate.

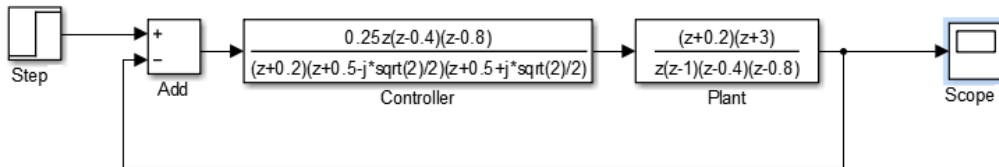


Figura 4. Implementarea în *Simulink* a sistemului din *Aplicație 2*

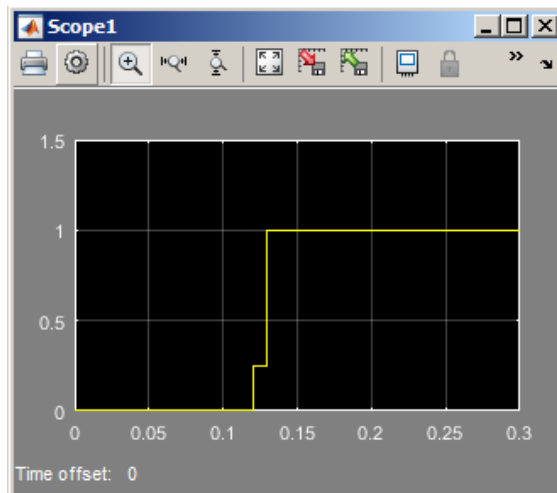


Figura 5. Răspunsul sistemului din *Aplicație 2* la semnalul treaptă unitate (întârziat cu 0.1s)

Probleme și aplicații

1. Fie sistemul din *Figura 1* cu funcția de transfer a sub-sistemului *Plant*:

$$P(z) = \frac{z + 0.5}{(z - 0.9) \cdot (z - 0.7) \cdot (z - 0.5)}$$

Să se determine funcția de transfer a controller-ului *deadbeat* și să se modeleze sistemul cu reacție în *Simulink* (cu pasul de eșantionare $T_s = 0.01$ și semnal de intrare treaptă unitate întârziată cu 0.1s, vezi *Figura 2*).

2. Fie sistemul din *Figura 1* cu funcția de transfer a sub-sistemului *Plant*:

$$P(z) = \frac{0.01 \cdot (z - 1.03)}{(z - 0.97)^2}$$

Să se determine funcția de transfer a controller-ului *deadbeat* și să se modeleze sistemul cu reacție în *Simulink* (cu pasul de eșantionare $T_s = 0.01s$ și semnal de intrare treaptă unitate întârziată cu 0.1s, vezi *Figura 2*).

Bibliografie

- [1] Miha P. Ioan, *Controlul Digital al Proceselor - Note de curs*, Universitatea Lucian Blaga, Sibiu, 2016
- [2] Richard C. Dorf , Robert H. Bishop, *Modern Control Systems*, Prentice-Hall, Inc., Upper Saddle River, NJ, 2000
- [3] Chi-Tsong Chen, *Analog and Digital Control System Design: Transfer-function, State-space, and Algebraic Methods*, Oxford University Press, Inc., New York, NY, 1995
- [4] P.E. Wellstead, *Introduction to Physical System Modelling*, Academic Press, New York (1979)
- [5] Benjamin C. Kuo, *Automatic Control Systems*, Prentice Hall PTR, Upper Saddle River, NJ, 1987
- [6] *Simulink, User's Guide, R2015a*, The MathWorks, Inc
- [7] *MATLAB, Programming Fundamentals, R2015a*, The MathWorks, Inc