Upoznavanje sa paketom *ControlSystems*

Nedeljko Stojaković
Marko Pejić

Novembar, 2021.

Cilj ovog dokumenta je da se upoznamo sa funkcijama paketa *ControlSystems* i načinima njihove upotrebe.

1. Uvod

Paket *ControlSystems* sadrži funkcije pomoću kojih je moguće definisati matematički model sistema u obliku funkcije prenosa i modela u prostoru stanja, kao i funkcije za simulaciju, odnosno za određivanja odziva sistema na različite pobude. Takođe, tu su funkcije koje su veoma korisne kod opisa složenih sistema. Pored toga, sadrži funkcije koje su nezaobilazne u analizi upravljanja sistemom, poput analize stabilnosti sistema.

2. Pregled osnovnih funkcija

U ovom poglavlju dat je pregled korisnih funkcija iz paketa *ControlSystems*, kako bi se ilustrovale najbitnije mogućnosti ovog paketa. Bitno je napomenuti da neke od dodatnih funkcionalnosti paketa nisu ni navedene, kao što se ne razmatraju svi oblici i/ili parametri funkcija, kako bi se fokus zadržao na najznačajnijim mogućnostima.

2.1 Kreiranje modela sistema

Paket *ControlSystems* podržava rad sa linearnim vremenski nepromenljivim modelima opisanim kao:

- model u prostoru stanja - ***ss***
- funkcija prenosa opisana količnikom polinoma - ***tf***
- funkcija prenosa opisana preko nula, polova i pojačanja - ***zpk***

Vremenski diskretni modeli imaju dodatni parametar *Ts*, koji predstavlja vreme odabiranja. Model se opisuje jednim objektom (promenljivom), čiji je tip određen prilikom kreiranja objekta. Svi navedeni tipovi (*ss*, *tf*, *zpk*) izvedeni su iz *LTI* (*Linear Time Invariant*) tipa objekta, zbog čega dele neke zajedničke osobine.

U nastavku je dat primer opisa linearnog modela u prostoru stanja (vremenski kontinualnog).

```
A = [0 1; 0 -12.5]
B = [0; 38.9]
C = [0 1]
D = 0
m = ss(A, B, C, D)
```

Prilikom opisa modela na ovaj način, funkciji *ss* prosleđuju se četiri parametra: matrice A, B, C i D, dok je rezultat smešten u promenljivu m.

```
StateSpace{Continuous, Float64}
A =
 0.0    1.0
 0.0  -12.5
B =
  0.0
 38.9
C =
 0.0  1.0
D =
 0.0

Continuous-time state-space model
```

Linearan vremenski diskretan model u prostoru stanja opisuje se na sličan način, uz dodatni parametar Ts, koji predstavlja periodu odabiranja.

```
Ts = 0.025
E = [1 0.02147; 0 0.73160]
F = [0.01098; 0.83520]
C = [0 1]
D = 0
md = ss(E, F, C, D, Ts)
```

Rezultat izvršavanja ovog koda prikazan je na sledećoj slici, gde se može primetiti da se radi o vremenski diskretnom modelu u prostoru stanja, periode odabiranja $0.025\,s$.

```
StateSpace{Discrete{Float64}, Float64}
A =
 1.0  0.02147
 0.0  0.7316
B =
 0.01098
 0.8352
C =
 0.0  1.0
D =
 0.0

Sample Time: 0.025 (seconds)
Discrete-time state-space model
```

Još jedan od načina za definisanje modela nekog sistema jeste upotrebom funkcije prenosa opisane količnikom polinoma $P(s)$ i $Q(s)$. Polinomi $P(s)$ i $Q(s)$ definišu se kao vektori koeficijenata počevši od najvišeg stepena Laplasovog operatora s.

```
P = [38.9]
Q = [1.0, 5.0, 6.0]
G = tf(P, Q)
```

Takođe, isti tip modela se može opisati i upotrebom promenljive s koja se uvodi funkcijom *tf* i predstavlja Laplasov operator. Prilikom formiranja funkcije prenosa, uz operator s moguće je koristiti i operacije $+$, $-$, $*$, $/$, .

```
s = tf("s")
G = 38.9 / (s^2 + 5 * s + 6)
```

Rezultat izvršavanja prethodnih kodova je funkcija prenosa $G(s) = \dfrac{38.9}{s^2 + 5s + 6}$ prikazana na sledećoj slici.

```
TransferFunction{Continuous, ControlSystems.SisoRational{Float64}}
      38.9
-------------------
1.0s^2 + 5.0s + 6.0

Continuous-time transfer function model
```

Funkciju prenosa je moguće zapisati i preko nula, polova i pojačanja, upotrebom funkcije *zpk*. Nule funkcije prenosa $G(s) = \dfrac{P(s)}{Q(s)}$ predstavljaju korene polinoma $P(s)$, a polovi predstavljaju korene polinoma $Q(s)$, dok je pojačanje konstanta sa kojom množimo taj količnik. Prema tome, ukoliko želimo da definišemo istu funkciju prenosa kao u prethodnim primerima koristeći *zpk*, potrebno je da odredimo nule (nema nula), polove (-2 i -3) i pojačanje (38.9).

```
nule = []
polovi = [-2, -3]
pojacanje = 38.9
G = zpk(nule, polovi, pojacanje)
```

Rezultat izvršavanja datog koda prikazan je na sledećoj slici.

```
TransferFunction{Continuous, ControlSystems.SisoZpk{Float64, Float64}}
             1.0
38.9-----------------------
    (1.0s + 2.0)(1.0s + 3.0)

Continuous-time transfer function model
```

Ukoliko imamo već definisanu promenljivu koja predstavlja funkciju prenosa, vrednosti nula, polova i pojačanja moguće je dobiti pozivom funkcije *zpkdata*. Na primer, za funkciju prenosa $G(s) = \dfrac{s+1}{s^2+3s+2}$, dobijamo da je vrednost nule -1, vrednosti polova su -1 i -2, a pojačanje je 1.

```
G = tf([1, 1], [1, 3, 2])
z, p, k = zpkdata(G)
```

U dosadašnjim primerima funkcije prenosa radilo se isključivo sa modelima sa jednim ulazom i jednim izlazom. Paket *ControlSystems* podržava i rad sa multivarijabilnim modelima, odnosno modelima sa više ulaza i/ili izlaza. Opis ovakvog modela upotrebom funkcije *ss* svodi se na definisanje ulaznih matrica odgovarajućih dimenzija. Međutim, za definisanje multivarijabilnih modela upotrebom funkcija *tf* ili *zpk*, neophodno je kreirati matricu funkcija prenosa. Na primer, sledeći kod ilustruje kreiranje matrice funkcija prenosa sistema sa dva ulaza i dva izlaza.

```
W = [tf(1, [2, 3]) tf(1, [1, 0]); tf(10, [3, 1]) tf(1, [1, 2, 3])]
```

Izvršavanjem datog koda dobijaju se četiri funkcije prenosa, uz odgovarajući ispis iz kog se jasno vidi koja funkcija prenosa se odnosi na koji par ulaz-izlaz.

```
TransferFunction{Continuous, ControlSystems.SisoRational{Int64}}
Input 1 to output 1
  1
------
2s + 3

Input 1 to output 2
  10
------
3s + 1

Input 2 to output 1
1
-
s

Input 2 to output 2
     1
------------
s^2 + 2s + 3

Continuous-time transfer function model
```

2.2 Konverzije modela

Funkcije (konstruktori modela) *ss*, *tf* i *zpk* mogu se upotrebiti za konverziju ranije unetog modela iz jednog tipa u drugi tip modela. U nastavku je dato nekoliko primera koji ilustruju navedene konverzije između različitih tipova modela.

```
G1 = tf(1, [1, 2, 1])
```

```
TransferFunction{Continuous, ControlSystems.SisoRational{Int64}}
     1
------------
s^2 + 2s + 1

Continuous-time transfer function model
```

Ukoliko imamo funkciju prenosa $G1$ definisanu pomoću funkcije *tf*, konverziju u funkciju prenosa opisanu pomoću nula, polova i pojačanja moguće je izvršiti prosleđivanjem deklarisane funkcije prenosa $G1$ funkciji *zpk*.

```
G2 = zpk(G1)
```

```
TransferFunction{Continuous, ControlSystems.SisoZpk{Float64, ComplexF64}}
             1.0
1.0-----------------------
   (1.0s + 1.0)(1.0s + 1.0)

Continuous-time transfer function model
```

Na sličan način, pozivom funkcije *ss* moguće je izvršiti konverziju iz funkcije prenosa (opisane preko *tf* ili *zpk*) u model u prostoru stanja.

```
sys = ss(G1)
```

```
StateSpace{Continuous, Float64}
A =
  0.0   1.0
 -1.0  -2.0
B =
 0.0
 1.0
C =
 1.0  0.0
D =
 0.0

Continuous-time state-space model
```

Sve navedene funkcije mogu se koristiti za transformacije modela između svih mogućih kombinacija tipova u okviru paketa *ControlSystems*.

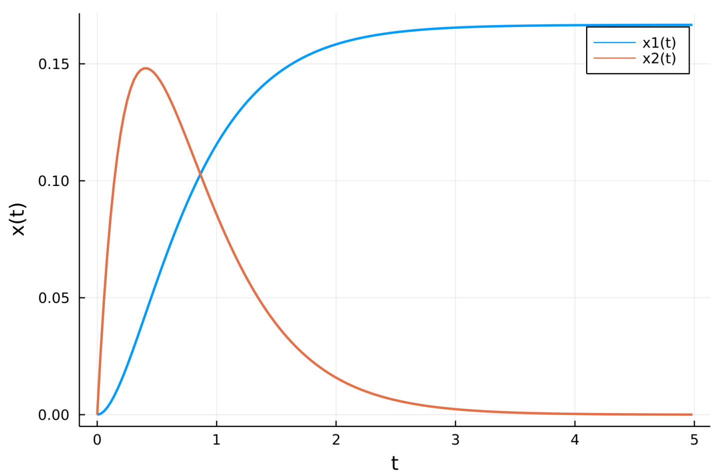
2.3 Analiza ponašanja modela

Analiza ponašanja modela može se sprovesti u vremenskom i u kompleksnom domenu. Analiza u vremenskom domenu podrazumeva simulacije, odnosno izračunavanje izlaza modela kada se na ulaz dovede željena pobuda. Funkcije za izračunavanje izlaza su:

- ***step(sys, t)*** - funkcija izračunava jedinični odziv datog modela (odziv ako se na ulaz dovede Hevisajdov signal $h(t)$). Drugi parametar funkcije predstavlja skalar koji označava završni trenutak simulacije ili ceo vektor vremenskih trenutaka.
- ***impulse(sys, t)*** - funkcija izračunava impulsni odziv datog modela (odziv ako se na ulaz dovede Dirakov impuls $\delta(t)$). Drugi parametar funkcije predstavlja skalar koji označava završni trenutak simulacije ili ceo vektor vremenskih trenutaka.
- ***lsim(sys, u, t, x0)*** - funkcija izračunava odziv sistema na prosleđenu pobudu $u(t)$. Ukoliko postoji više ulaza u sistem, onda se u definiše kao vektor ulaza. Moguće je proslediti i početne uslove, koji se smatraju nultim ukoliko se ne prosledi nikakva vrednost.

Sledeći primer izračunava jedinični odziv u trajanju od 5 sekundi modela opisanog pomoću funkcije prenosa.

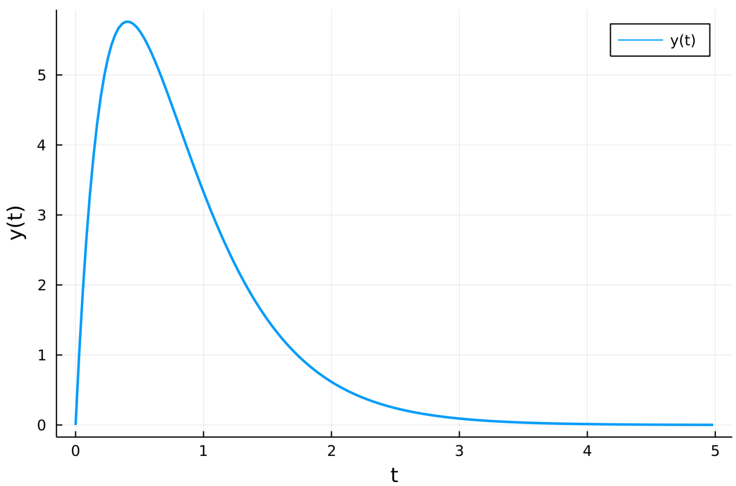
```
G = tf(38, [1, 5, 6])
y, t, x = step(G, 5)
plot(t, x', label=["x1(t)" "x2(t)"], lw=2, xlabel="t", ylabel="x(t)")
```



Dobijeni rezultat simulacije čine vrednosti izlaza y i promenljivih stanja x izračunate u vremenskim trenucima t. Promenljiva t sadrži niz monotono rastućih vrednosti simulacionih trenutaka, počevši od vrednosti 0 do zadatog trajanja simulacije (5 u ovom slučaju). Za svaki vremenski trenutak postoji vrsta vrednosti izlaza y i promenljivih stanja x, tako da sva tri navedena vektora imaju isti broj vrsta. Ukoliko model ima više izlaza, tada broj kolona u y odgovara broju izlaza. Na isti način predstavljaju se vredosti promenljivih stanja u vektoru x. Sledeći grafik prikazuje promene vrednosti za obe promenljive stanja sistema opisanog funkcijom prenosa, kada je na ulaz doveden jedinični odskočni signal $h(t)$.

Ukoliko želimo da odredimo impulsni odziv za isti model, sve što je potrebno uraditi jeste iskoristiti funkciju *impulse*, što je prikazano u narednom primeru.

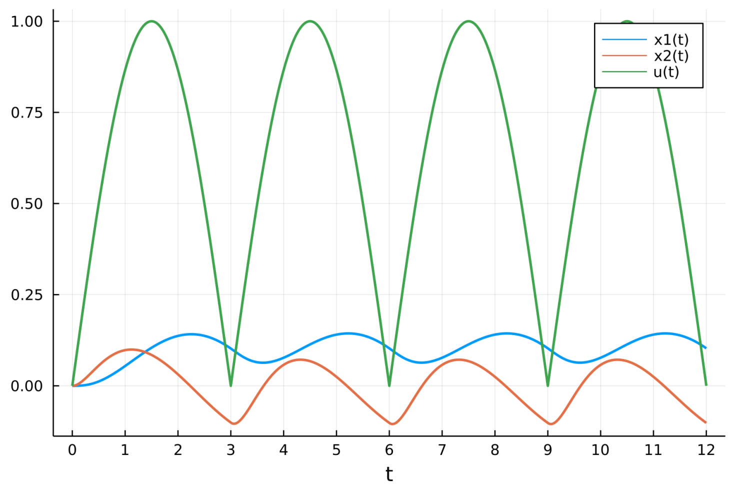
```
G = tf(38, [1, 5, 6])
y, t, x = impulse(G, 5)
plot(t, y', label="y(t)", lw=2, xlabel="t", ylabel="y(t)")
```



U situacijama kada je pobuda definisana kao složen signal, za određivanje odziva modela koristi se funkcija *lsim*. Sledeći primer ilustruje nalaženje odziva modela opisanog funkcijom prenosa $G(s) = \dfrac{38.9}{s^2 + 5s + 6}$, kada je pobuda definisana kao $u(t) = |sin(\dfrac{\pi}{3}t)|$, u trajanju od $12\,s$.

```
function signal(t)
    return abs.(sin.(pi / 3 * t))
end

G = tf(38.9, [1, 5, 6])

t = 0:0.01:12
u = signal(t)

y, t, x = lsim(G, u', t)

plot(t, x', label=["x1(t)" "x2(t)"], lw=2, xlabel="t", xticks=0:12)
plot!(t, u, label="u(t)", lw=2, xticks=0:12)
```

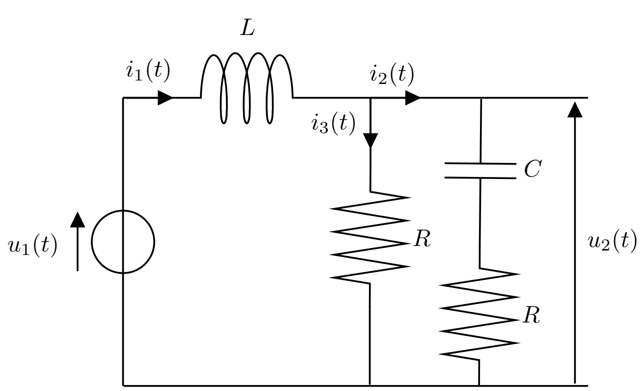
U priloženom kodu, na samom početku implementirana je funkcija koja definiše ulaz u sistem $u(t)$, nakon čega je kreiran model pomoću funkcije tf. U ovom primeru, funkciji za izračunavanje izlaza modela (*lsim*) prosleđen je model, vektor koji predstavlja ulazni signal, kao i vektor vremenskih trenutaka (za razliku od prethodnih primera, gde je vreme prosleđivano kao skalar). Izvršavanjem simulacije dobijeni su vektori vrednosti izlaza i promenljivih stanja modela, a grafički su prikazani ulazni signal $u(t)$, kao i promene vrednosti promenljivih stanja u vremenu $x_1(t)$ i $x_2(t)$.



U slučaju da je sistem opisan kao model u prostoru stanja, funkciji *lsim* moguće je proslediti i četvrti parametar $x0$, koji predstavlja vektor početnih uslova.

3. Primeri sa rešenjima

Primer 1. Odrediti funkciju prenosa $G(s) = \dfrac{U_2(s)}{U_1(s)}$ za električno kolo sa slike.



Upotrebom paketa *ControlSystems* odrediti odziv sistema tokom prvih $10\,s$ za sledeće vrednosti parametara $R = 0.1k\Omega$, $L = 22mH$ i $C = 470nF$, ako je ulazni napon:

a) jedinična odskočna pobuda $u_1(t) = h(t)$,

b) $u_1(t) = 10|sin(t)|$.

Rešenje:

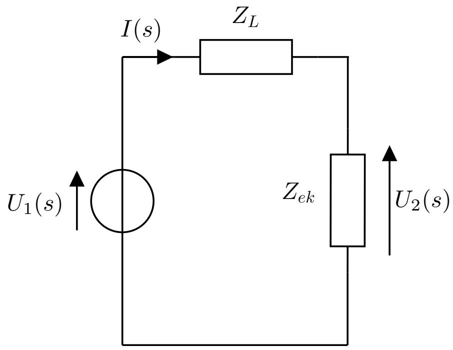
Verovatno najjednostavniji način za rešavanje ovakvih električnih kola jeste preko impedansi. Da bismo rešili kolo preko impedansi, neophodno je da znamo izraz za impedansu svakog od elemenata električnog kola:

- impedansa otpornika $Z_R = R$,
- impedansa kondenzatora $Z_C = \dfrac{1}{j\omega C} = \dfrac{1}{sC}$
- impedansa kalema $Z_L = j\omega L = sL$

Uvođenjem impedansi, moguće je pronaći ekvivalentnu impedansu za kondenzator i dva otpornika:

$$Z_{ek} = (Z_C + Z_R)||Z_R = \frac{(Z_C + Z_R)Z_R}{Z_C + 2Z_R}$$

Na ovaj način, početno kolo svedeno je na električno kolo prikazano na sledećoj slici, koje se dalje veoma jednostavno rešava.



$$U_1 - I(Z_L + Z_{ek}) = 0 \quad \Rightarrow \quad I = \frac{U_1}{Z_L + Z_{ek}}$$

$$U_1 - Z_L I = U_2$$

$$U_1 - Z_L \frac{U_1}{Z_L + Z_{ek}} = U_2$$

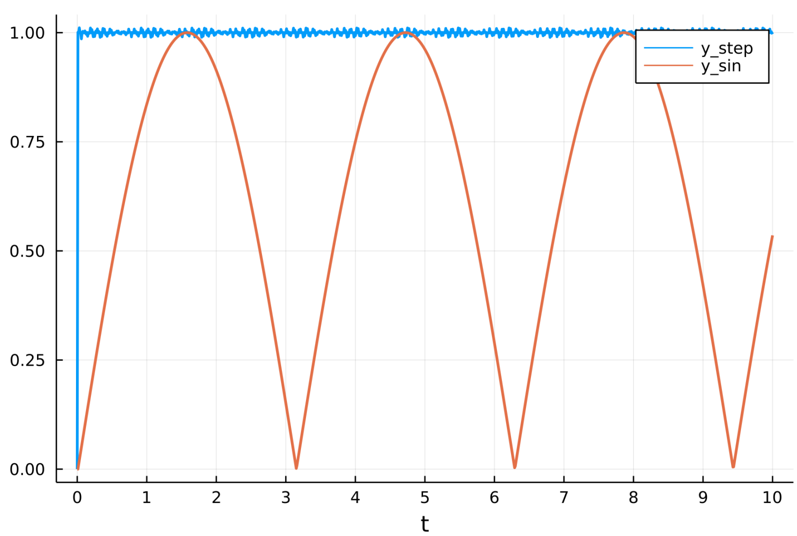
$$(1 - \frac{Z_L}{Z_L + Z_{ek}})U_1 = U_2$$

$$\text{‘}1tr. \boxed{G(s) = \frac{U_2(s)}{U_1(s)} = \frac{Z_{ek}}{Z_L + Z_{ek}}}$$

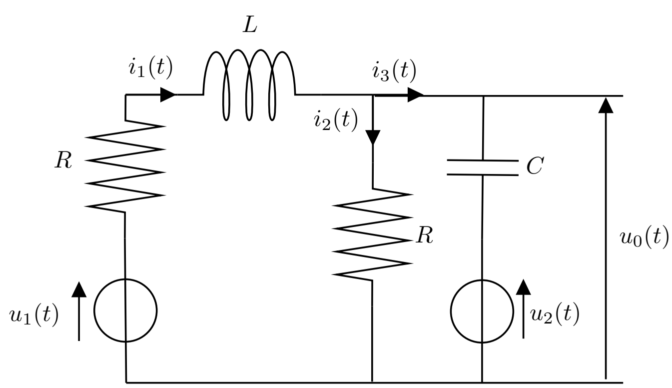
U nastavku rešenja dat je kod kojim je izvršena simulacija upotrebom paketa *ControlSystems*. Funkcija *model_el_kola* kreira funkciju prenosa datog električnog kola, za prosleđene vrednosti konstanti R, L i C. Treba primetiti upotrebu funkcije *minreal* iz paketa *ControlSystems*, koja vrši skraćivanje istih činioca polinoma brojioca i imenioca (skraćivanje istih nula i stabilnih polova) [1]. Dalje je dobijena funkcija prenosa prosleđena funkcijama *step* i *lsim*, čime su dobijeni odzivi modela za različite pobudne signale.

[1] Na primer, ukoliko dobijemo funkciju prenosa $G(s) = \dfrac{P(s)}{Q(s)}$, gde je $P(s)$ polinom drugog reda, a $Q(s)$ polinom četvrtog reda, funkcija *minreal* će skratiti iste nule i polove. Prema tome, ukoliko je postojao jedan isti par nula-pol, nakon izvršavanja funkcije *minreal*, $P(s)$ će biti polinom prvog, a $Q(s)$ trećeg reda. U slučaju da su postojala dva ista para nula i polova, $P(s)$ će sadržati samo slobodan član, a $Q(s)$ će biti polinom drugog reda.

```
function model_el_kola(R, L, C)
    s = tf("s")

    Zr = R
    Zl = s * L
    Zc = 1 / (s * C)
    Zek = ((Zc + Zr) * Zr) / (Zc + 2 * Zr)

    return minreal(Zek / (Zek + Zl))
end

R = 100                 # 0.1 kΩ
L = 0.022               # 22 mH
C = 4.7 * 10^-7         # 470 nF

G = model_el_kola(R, L, C)

t = 0:0.01:10
u = abs.(sin.(t))

y_step, t, x = step(G, t)
y_sin, t, x = lsim(G, u', t)

plot(t, y_step', lw=2, xticks=0:10, label="y_step")
plot!(t, y_sin', lw=2, label="y_sin")
```



Primer 2. Definisati matematički model u prostoru stanja za električno kolo sa slike, tako da promenljive stanja budu struja $i_1(t)$ i napon $u_c(t)$, a naponi u_1 i u_2 ulazi u sistem, dok su izlazi iz sistema $i_2(t)$ i $u_0(t)$. Upotrebom paketa *ControlSystems* odrediti odziv sistema i grafički ga prikazati, za sledeće vrednosti parametara: $R = 1k\Omega$, $L = 22mH$ i $C = 470nF$, dok su ulazni naponi $u_1(t) = u_2(t) = sin(t)$.



$$i_1 = i_2 + i_3 \tag{1}$$
$$u_1 - i_1R - u_L - i_2R = 0 \tag{2}$$
$$i_2R - u_C - u_2 = 0 \tag{3}$$

S obzirom da u prethodnim jednačinama figurišu napon na kalemu u_L i struja kroz kondenzator i_3, potrebno je zameniti ih sa:

$$u_L = L\frac{di_1}{dt}$$
$$i_3 = C\frac{du_c}{dt}$$

Uvođenjem datih smena i sabiranjem druge i treće jednačine ((2) + (3)) dobija se sledeća jednačina, iz koje jednostavno možemo izraziti izvod promenljive stanja i_1.

$$u_1 - i_1R - L\dot{i_1} - u_c - u_2 = 0$$
$$\boxed{\dot{i_1} = \frac{1}{L}(u_1 - i_1R - u_c - u_2)}$$

Ukoliko iz jednačine (1) izrazimo $i_2 = i_1 - i_3 = i_1 - C\dot{u}_c$ i uvrstimo u jednačinu (3), dobijamo:

$$(i_1 - C\dot{u}_c)R - u_c - u_2 = 0$$

S obzirom da u prethodnoj jednačini takođe figurišu samo promenljive stanja, ulazi i konstante, moguće je izraziti $\dot{u}_c$:

$$\boxed{\dot{u}_c = \frac{1}{RC}(i_1R - u_c - u_2)}$$

Ovim su dobijeni izvodi obe promenljive stanja, što znači da je potrebno još napisati jednačine za izlaze, i zapisati sve u matričnoj formi modela u prostoru stanja.

Napon u_0 predstavlja jedan od dva izlaza, što znači da je potrebno izraziti ga kao zavisnost promenljivih stanja i ulaza. Sa slike se jasno vidi da je taj napon jednak zbiru napona na kondenzatoru u_c koji predstavlja promenljivu stanja i napona u_2 koji predstavlja ulaz:

$$\boxed{u_0 = u_c + u_2}$$

S obzirom da je napon na otporniku u grani kroz koju teče struja i_2 jednak naponu $u_0 = u_2 + u_c$, jasno je da je struja i_2 jednaka količniku tog napona i otpornosti u toj grani:

$$\boxed{i_2 = \frac{u_2 + u_c}{R}}$$

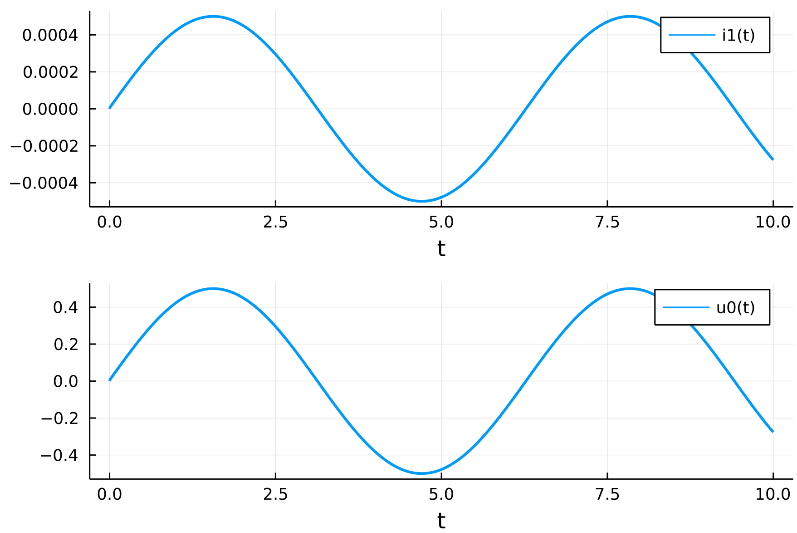
Sada su poznate jednačine izvoda svih promenljivih stanja i izlaza, tako da je preostalo zapisati matematički model u prostoru stanja u matričnoj formi. Ukoliko vektore ulaza, promenljivih stanja i izlaza definišemo na sledeći način:

$$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}, x(t) = \begin{bmatrix} i_1(t) \\ u_c(t) \end{bmatrix}, y(t) = \begin{bmatrix} i_2(t) \\ u_0(t) \end{bmatrix},$$

model u prostoru stanja datog električnog kola je:

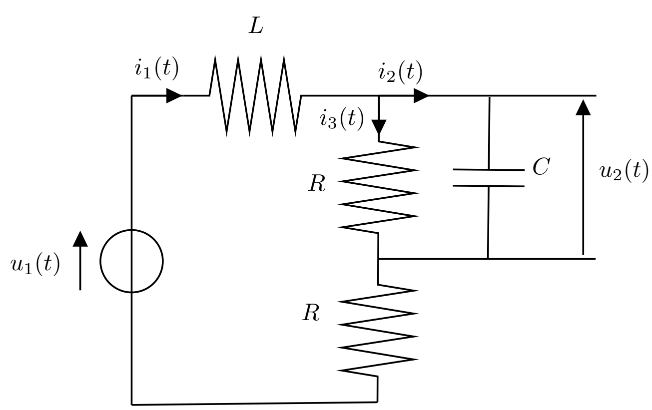
$$\boxed{\begin{aligned} \dot{x}(t) &= \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} x(t) + \begin{bmatrix} \frac{1}{L} & -\frac{1}{L} \\ 0 & -\frac{1}{RC} \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 0 & \frac{1}{R} \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 & \frac{1}{R} \\ 0 & 1 \end{bmatrix} u(t) \end{aligned}}$$

```julia
function model_el_kola(R, L, C)
    ssA = [-R/L -1/L; 1/C -1/(R * C)]
    ssB = [1/L -1/L; 0 -1/(R * C)]
    ssC = [0 1/R; 0 1]
    ssD = [0 1/R; 0 1]
    return ss(ssA, ssB, ssC, ssD)
end

R = 1000                # 1 kΩ
L = 0.022               # 22 mH
C = 4.7 * 10^-7         # 470 nF

model = model_el_kola(R, L, C)

t = 0:0.01:10
u = sin.(t)
y, t, x = lsim(model, [u u]', t)

p1 = plot(t, y[1, :], lw=2, label="i1(t)", xlabel="t")
p2 = plot(t, y[2, :], lw=2, label="u0(t)", xlabel="t")
plot(p1, p2, layout=(2, 1))
```

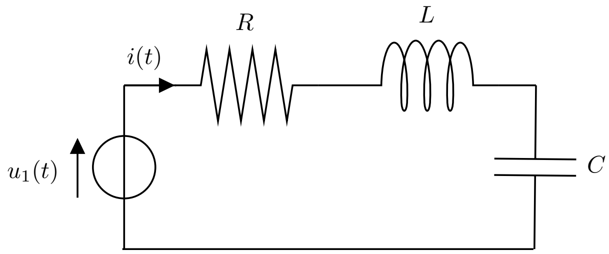


4. Zadaci za vežbu

Zadatak 1. Odrediti funkciju prenosa $G(s) = \dfrac{U_2(s)}{U_1(s)}$ električnog kola prikazanog na slici, ako je $RC = 1$. Odrediti i grafički prikazati odziv sistema, ako je ulazni napon $u_1(t) = 2cos(t)$, a vreme simulacije je $20\,s$.



Zadatak 2. Za električno kolo prikazano na slici odrediti funkciju prenosa i model u prostoru stanja, ako je ulaz napon $u_1(t)$, a izlaz zbir napona na kalemu i kondenzaroru $y(t) = u_L(t) + u_c(t)$. Upotrebom paketa *ControlSystems* odrediti i grafički prikazati izlazni napon, za sledeće vrednosti parametara: $R = 1k\Omega$, $L = 22mH$, $C = 470nF$, $u_1(t) = sin(\dfrac{pi}{3}t)$, a vreme simulacije $10\,s$.



Literatura

- Aleksandar Erdeljan, Darko Čapko: Modelovanje i simulacija sistema - sa primerima; FTN, Novi Sad, 2015.
- *Julia* programski jezik (sajt) https://julialang.org/
- *Think Julia* (*online* knjiga) https://benlauwens.github.io/ThinkJulia.jl/latest/book.html.
- *ControlSystems* dokumentacija https://juliacontrol.github.io/ControlSystems.jl/latest/