

Sistemi opisani blok dijagramima

Nedeljko Stojaković

Marko Pejić

Nikola Gavranović

Decembar, 2021.

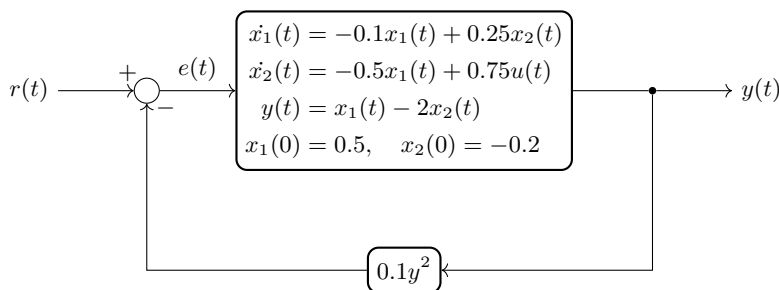
Cilj ovog dokumenta je rešavanje (simulacija) sistema opisanih blok dijagramima, koristeći pakete *DifferentialEquations* i *ControlSystems*.

1. Uvod

U prethodnom materijalu, videli smo da je element blok dijagrama isključivo funkcija prenosa, koja je po definiciji odnos kompleksnih likova izlaza i ulaza pri nultim početnim uslovima i važi samo za linearne sisteme. Ukoliko u sistemu postoje nelinearnosti (npr. ograničenje signala, maksimum, minimum), nenulti početni uslovi, nelinearne diferencijalne jednačine itd. sistem ili deo sistema se ne može predstavljati funkcijom prenosa, već je potrebno svaki specifičan deo predstaviti odgovarajućim blokom. Na ovaj način dobijamo blok dijagrame čiji su elementi raznoliki, pa nije moguće raditi transformacije za različite tipove veza (redna, paralelna, povratna sprega). Kao što je bio slučaj i do sada, ovi sistemi takođe mogu biti multivarijabilni.

2. Primeri sa rešenjima

Primer 1. Odrediti izlaz sistema $y(t)$ sa slike. Na istom grafiku prikazati $r(t)$ i $y(t)$, ako je $r(t) = \sin(2t)$.



U glavnoj grani je blok opisan matematičkim modelom u prostoru stanja, a u povratnoj grani je nelinearna funkcija koja kvadrira ulazni signal i množi ga sa 0.1.

Očigledno je da ovakav sistem ne možemo opisati funkcijom prenosa.

U modelu imamo sistem od dve diferencijalne jednačine što se može rešiti upotrebom funkcija iz paketa *DifferentialEquations*.

$$\dot{x}_1(t) = -0.1x_1(t) + 0.25x_2(t)$$

$$\dot{x}_2(t) = -0.5x_1(t) + 0.75u(t)$$

Problem je nepoznati ulazni signal u .

Sa slike modela se može zaključiti da je ulazni signal u blok (u), u stvari vrednost e koju možemo izraziti kao:

$$e = r - 0.1y^2$$

A izlazni signal y iz postavke matematičkog modela se računa kao:

$$y = x_1 - 2x_2$$

Uvrštavanjem smena se dobija sistem diferencijalnih jednačina u formi $\dot{x}_i = f_i(\mathbf{x}, \mathbf{u})$, što se dalje rešava funkcijama iz *DifferentialEquations* paketa.

```
using DifferentialEquations, Plots, LinearAlgebra

function sistem(dx, x, p, t)

    y = x[1] - 2 * x[2]
    r = sin(2 * t)
    u = r - 0.1 * y^2

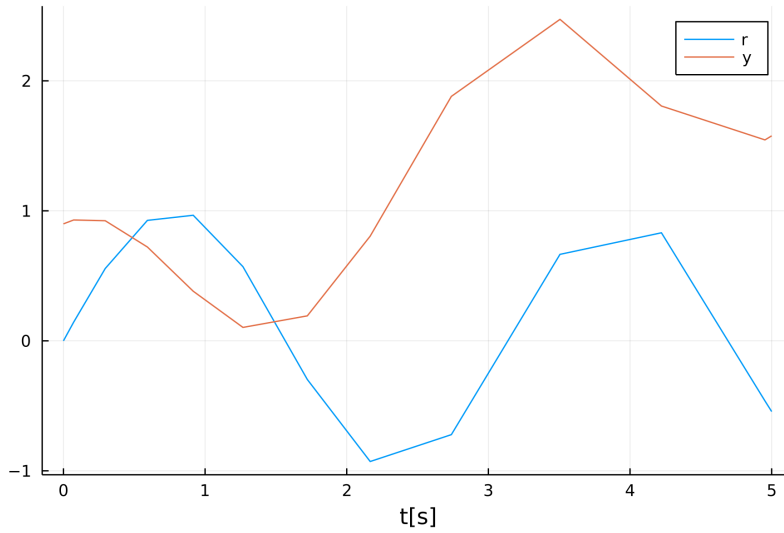
    dx[1] = -0.1 * x[1] + 0.25 * x[2]
    dx[2] = -0.5 * x[1] + 0.75 * u

end

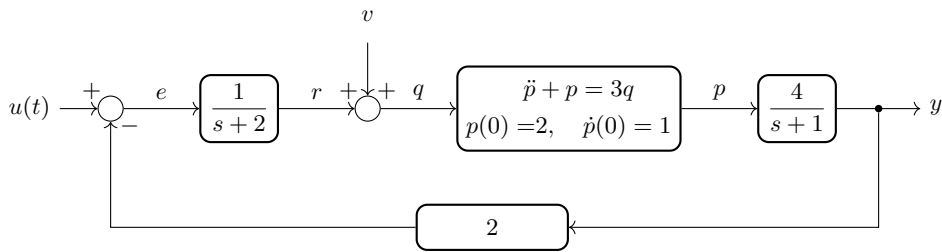
tspan = (0.0, 5.0)
x0 = [0.5, -0.2]

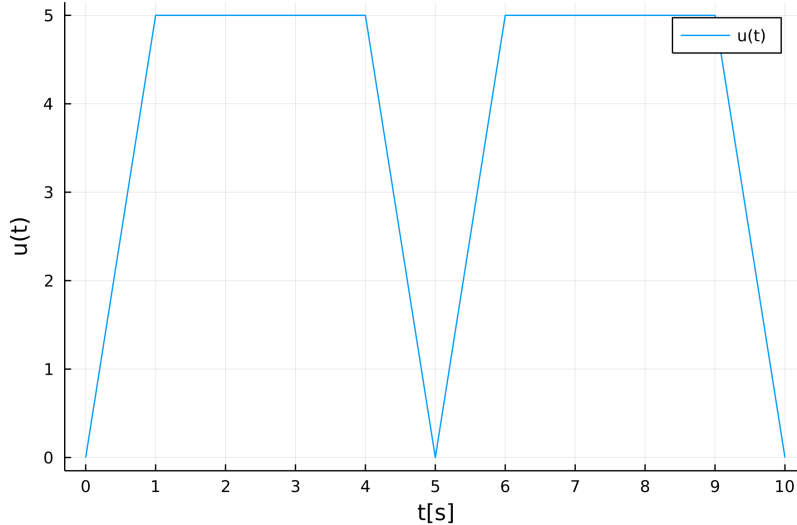
prob = ODEProblem(sistem, x0, tspan)
sol = solve(prob)
x1 = [x[1] for x in sol.u]
x2 = [x[2] for x in sol.u]

y = x1 - 2 * x2
r = sin.(2 * sol.t)
plot(sol.t, r, label = "r")
plot!(sol.t, y, label = "y", xlabel = "t[s]")
```



Primer 2. Odrediti izlaz sistema $y(t)$ opisanog blok dijagramom sa slike, ukoliko je pobuda $u(t)$ signal predstavljen grafikom, a pobuda $v(t) = h(t - 3)$.





Rešenje:

Pošto se u modelu nalazi diferencijalna jednačina koja ima nenulte početne uslove, izlaz $y(t)$ ćemo odrediti paketom *DifferentialEquations*. To znači da nam je potreban sistem diferencijalnih jednačina prvog reda koji opisuje ovaj model. Posmatraćemo svaki blok pojedinačno i određivati diferencijalnu(e) jednačinu(e) prvog reda.¹

¹ Redosled posmatranja blokova je proizvoljan.

Posmatramo prvi blok (sa leva na desno). Po definiciji funkcije prenosa zapisujemo:

$$\frac{R(s)}{E(s)} = \frac{1}{s+2}$$

Određujemo diferencijalnu jednačinu:

$$sR(s) + 2R(s) = E(s)/\mathcal{L}^{-1}$$

$$\dot{r} + 2r = e$$

Uvodimo potrebne smene:

$$x_1 = r \quad \Rightarrow \quad \dot{x}_1 = e - 2x_1$$

Potrebno je sve vrednosti u jednačinama definisati preko uvedenih smena i vrednosti ulaznih signala u i v . U prethodnoj jednačini imamo vrednost e koju određujemo sa modela. U modelu vidimo da je vrednost e izlaz iz prvog sabirača, tako da dobijamo:

$$e = u - 2y$$

U ovom izrazu, potrebno je vrednost y zameniti sa odgovarajućom smenom. Za sada nismo definisali smenu za y , ali ćemo u nastavku i to uraditi.

Posmatramo sledeći blok u kome je postavljena diferencijalna jednačina drugog reda. Već poznatim postupkom, odredićemo sistem diferencijalnih jednačina prvog reda za ovaj blok.

$$\begin{aligned} \ddot{p} + p &= 3q \\ x_2 = p &\Rightarrow \dot{x}_2 = x_3 \\ x_3 = \dot{p} &\Rightarrow \dot{x}_3 = 3q - x_2 \end{aligned}$$

U jednačini \dot{x}_3 pojavila se vrednost q koja predstavlja izlaz iz drugog sabirača u modelu, tako da dobijamo:

$$q = v + r \implies q = v + x_1$$

Ostao je još jedan blok u kome se nalazi funkcija prenosa, tako da ćemo ponoviti isti postupak kao za prvi blok.

$$\begin{aligned} \frac{Y(s)}{P(s)} &= \frac{4}{s+1} \\ sY(s) + Y(s) &= 4P(s)/\mathcal{L}^{-1} \\ \dot{y} + y &= 4p \\ x_4 = y &\Rightarrow \dot{x}_4 = 4x_2 - x_4 \end{aligned}$$

Ovde smo dobili smenu za y i to ćemo uvrstiti u izraz za grešku e , tako da dobijamo $e = u - 2x_4$.

Sada imamo sve potrebne jednačine. Ostaje da se napiše *Julia* kod:

```
function sistem(dx, x, p, t)
    v = p;
    e = u(t) - 2*x[4];
    q = v(t) + x[1];
    dx[1] = e - 2*x[1];
    dx[2] = x[3];
    dx[3] = 3*q - x[2];
    dx[4] = 4*x[2] - x[4];
end

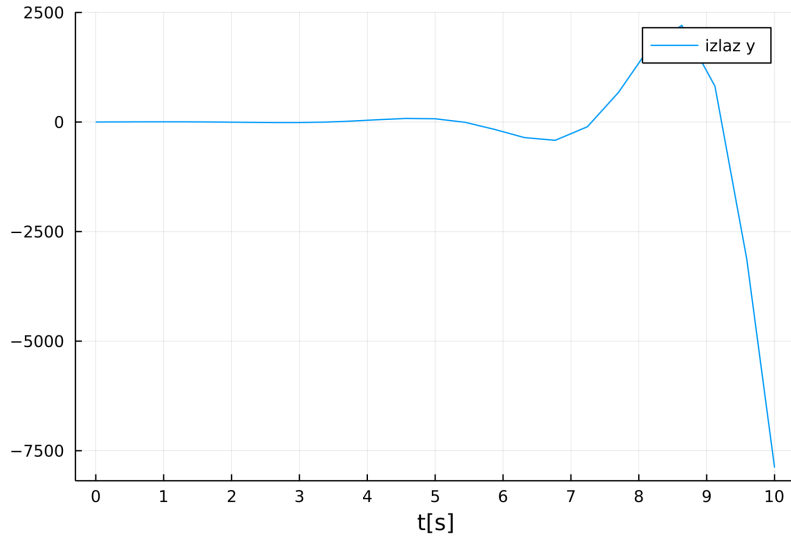
function u(t)
    tp = rem(t, 5);
    y = 5*tp * (tp < 1) + 5 * ((tp >=1) & (tp < 4)) +
        (-5*tp + 25) * (tp >=4);
end

interval = (0.0, 10.0);
x0 = [0.0, 2.0, 1.0, 0.0];
param = (t -> (t >= 3.0) ? 1.0 : 0.0);

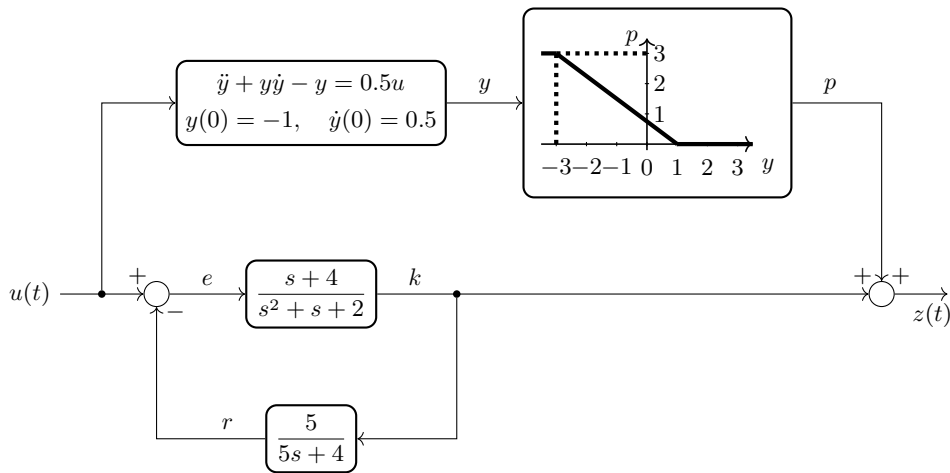
prob = ODEProblem(sistem, x0, interval, param);
r = solve(prob);
y = [x[4] for x in r.u];

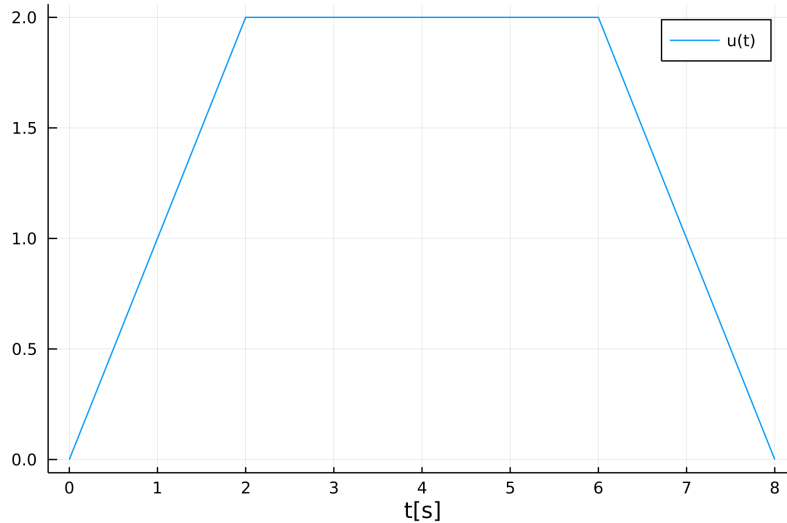
plot(r.t, y, label = "izlaz y", xlabel = "t[s]")
```

Pošto smo za prvi blok iskoristili smenu x_1 , onda za ovaj blok nastavljamo od x_2 . Ne možemo koristiti više puta istu oznaku za smenu u istom modelu.



Primer 3. Simulirati sistem sa slike u toku prvih 8s, ako je ulaz $u(t)$ prikazan na grafiku. Dobijeni izlaz prikazati na grafiku.





S obzirom da dati sistem sadrži blok predstavljen preko diferencijalne jednačine sa nenultim početnim uslovima, jasno je da je nemoguće odrediti izlaz isključivo upotrebom paketa *ControlSystems*. Međutim, pošto se izlaz formira kao zbir izlaza dve odvojene grane, moguće je izlaz donje grane predstaviti preko paketa *ControlSystems* (oba bloka su predstavljeni preko funkcija prenosa), a izlaz gornje grane predstaviti upotrebom paketa *DifferentialEquations*. Druga mogućnost za rešavanje zadatka jeste upotreba isključivo paketa *DifferentialEquations*. U nastavku su prikazana rešenja zadatka na oba navedena načina.

Nezavisno od načina na koji rešavamo zadatak, neophodno je definisati pobudnu silu, što je urađeno implementacijom sledeće funkcije.

```
function signal(t)
    y = t .* (t < 2) .+ 2 .* ((t .>= 2) .& (t < 6)) .+ (-t .+ 8) .* (t .>= 6)
end
```

I način: kombinacija paketa *DifferentialEquations* i *ControlSystems*

Paketom *DifferentialEquations* predstavimo gornju granu, odnosno datu diferencijalnu jednačinu drugog reda. Prvi korak je transformacija date jednačine u sistem diferencijalnih jednačina prvog reda, odnosno određivanje izvoda promenljivih stanja.

$$\ddot{y} + y\dot{y} - y = 0.5u$$

$$x_1 = y \quad \Rightarrow \quad \dot{x}_1 = \dot{y}$$

$$x_2 = \dot{y} \quad \Rightarrow \quad \dot{x}_2 = 0.5u + x_1 - x_1x_2$$

Izlaz bloka predstavljenog diferencijalnom jednačinom dovodi se na blok koji vrši

transformaciju definisanu sledećom funkcijom:

$$p = \begin{cases} 3, & y < -3 \\ -\frac{3}{4}x + \frac{3}{4}, & y \in [-3, 1) \\ 0, & y > 1 \end{cases}$$

Kod kojim je izvršeno numeričko određivanje kretanja promenljivih stanja dat je u nastavku.

```
function gornja_grana!(dx, x, p, t)
    u = signal(t)
    dx[1] = x[2]
    dx[2] = 0.5 * u + x[1] - x[1] * x[2]
end

tspan = (0.0, 8.0)
x0 = [-1, 0.5]

prob = ODEProblem(gornja_grana!, x0, tspan)
sol = solve(prob)
```

Donju granu možemo predstaviti pomoću algebre funkcije prenosa, odnosno preko povratne sprege između $G(s) = \frac{s+4}{s^2+s+2}$ u glavnoj grani i $H(s) = \frac{5}{5s+4}$ u povratnoj grani. Sledeći kod sadrži određivanje izlaza donje grane $k(t)$, kao i izračunavanje izlaza iz sistema $z(t)$. Bitno je napomenuti da je neophodno izvršiti interpolaciju rešenja dobijenih upotrebom paketa *DifferentialEquations*, kako bi vektori p i k imali isti broj elemenata.

```
function donja_grana()
    G = tf([1, 4], [1, 1, 2])
    H = tf(5, [5, 4])
    return minreal(feedback(G, H))
end

t = 0:0.01:8
u = signal(t)

sol_interpolirano = sol(t)

y = [x[1] for x in sol_interpolirano.u]
p = 3 .* (y .< -3) .+ (-3/4 * y .+ 3/4) .* ((y .>= -3) .& (y .< 1))

G = donja_grana()
k, ~, ~ = lsim(G, u', t)

z = p .+ k'

plot(t, z, lw=2, label="z(t)", xticks=0:8, xlabel="t [s]")
```

II način: upotreba paketa *DifferentialEquations*

U ovom slučaju, neophodno je i blokove predstavljene preko funkcija prenosa konvertovati u model u prostoru stanja, odnosno predstaviti ih preko diferencijalnih jednačina.

S obzirom da funkcija prenosa $\frac{s+4}{s^2+s+2}$ u brojiocu sadrži polinom zavisian od s , potrebno je da upotrebimo drugu kanonski formu kako bismo izvršili konverziju funkcije prenosa u model u prostoru stanja.

$$\frac{K(s)}{E(s)} = \frac{s+4}{s^2+s+2} \cdot \frac{Z(s)}{Z(s)}$$

$$K(s) = (s+4)Z(s) = sZ(s) + 4Z(s)$$

$$E(s) = (s^2+s+2)Z(s) = s^2Z(s) + sZ(s) + 2Z(s)$$

Promenljive stanja x_1 i x_2 smo već iskoristili za transformaciju bloka predstavljenog preko diferencijalne jednačine drugog reda, tako da nastavljamo od promenljive stanja x_3 , odnosno uvodimo sledeće smene:

$$X_3(s) = Z(s)$$

$$X_4(s) = sZ(s) \implies X_4(s) = sX_3(s) \xrightarrow{\mathcal{L}^{-1}} \dot{x}_3(t) = x_4(t)$$

Uvrštavanjem smena dobija se:

$$K(s) = X_4(s) + 4X_3(s)$$

$$sX_4(s) = E(s) - 2X_3(s) - X_4(s)$$

Primenom inverzne Laplasove transformacije dobijaju se sledeći izrazi:

$$k(t) = 4x_3(t) + x_4(t)$$

$$\dot{x}_4(t) = e(t) - 2x_3(t) - x_4(t)$$

Konverzija funkcije prenosa $\frac{5}{5s+4}$ u model u prostoru stanja izvršena je na sledeći način.

$$\frac{R(s)}{K(s)} = \frac{5}{5s+4}$$

$$5sR(s) + 4R(s) = 5K(s)$$

Primenom inverzne Laplasove transformacije dobija se sledeća diferencijalna jednačina prvog reda, za koju je potrebno uvesti novu promenljivu stanja.

$$5\dot{r} + 4r = 5k$$

$$x_5 = r \implies \dot{x}_5 = k - \frac{4}{5}x_5$$

Sad kada je poznata vrednost grane označene sa r , granu e možemo izraziti kao:

$$e = u - r = u - x_5$$

Na ovaj način definisani su izvodi svih promenljivih stanja i određeni su izlazi svih blokova, odnosno određene su sve veličine neophodne za implementaciju simulacije. Simulacijom su određene promene svih promenljivih stanja u vremenu, na osnovu čega se jednostavno određuju izlazi pojedinačnih grana $p(t)$ i $k(t)$, čijim se sabiranjem određuje izlaz celog sistema $z(t)$.

```
function sistem!(dx, x, p, t)
    u = signal(t)
    e = u - x[5]
    k = 4 * x[3] + x[4]

    dx[1] = x[2]
    dx[2] = 0.5 * u + x[1] - x[1] * x[2]
    dx[3] = x[4]
    dx[4] = e - 2 * x[3] - x[4]
    dx[5] = k - 4/5 * x[5]
end

tspan = (0.0, 8.0)
x0 = [-1.0, 0.5, 0.0, 0.0, 0.0]

prob = ODEProblem(sistem!, x0, tspan)
sol = solve(prob)

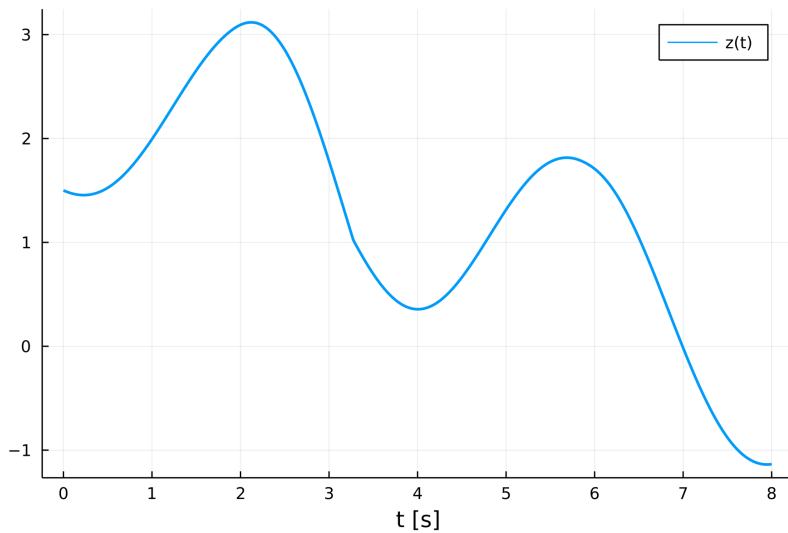
t = 0:0.01:8
sol_interpolirano = sol(t)

y = [x[1] for x in sol_interpolirano.u]
p = 3 .* (y .< -3) .+ (-3/4 * y .+ 3/4) .* ((y .>= -3) .& (y .< 1))

x3 = [x[3] for x in sol_interpolirano.u]
x4 = [x[4] for x in sol_interpolirano.u]
k = 4 * x3 .+ x4

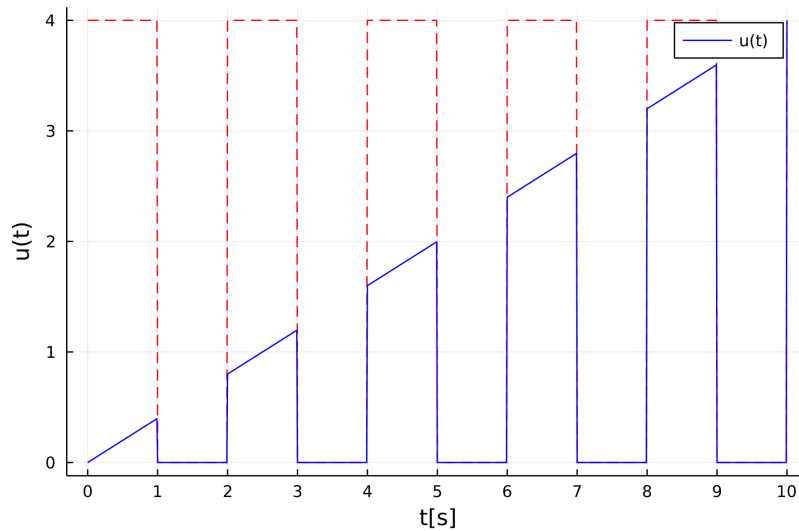
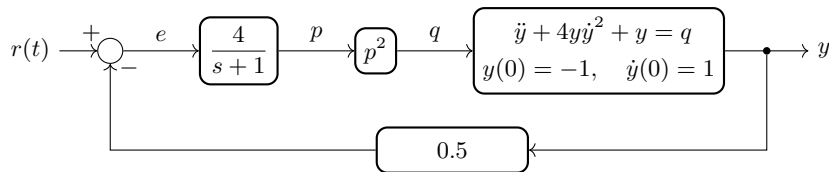
z = p .+ k

plot(t, z, lw=2)
```

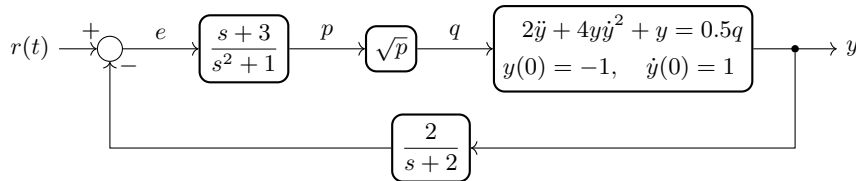


3. Zadaci za vežbu

Zadatak 1. Simulirati sistem sa slike u toku prvih 10s, ako je ulaz $u(t)$ prikazan na grafiku. Prikazati ulaz i izlaz sistema na istom grafiku.



Zadatak 2. Simulirati sistem sa slike u toku prvih 10s, ako je ulaz $u(t) = 2h(t-4)$. Prikazati ulaz i izlaz sistema na istom grafiku.



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>.