Digitalno upravljanje

Žarko Zečević
Elektrotehnički fakultet
Univerzitet Crne Gore

Predavanje 1

Uvod u digitalne sisteme upravljanja

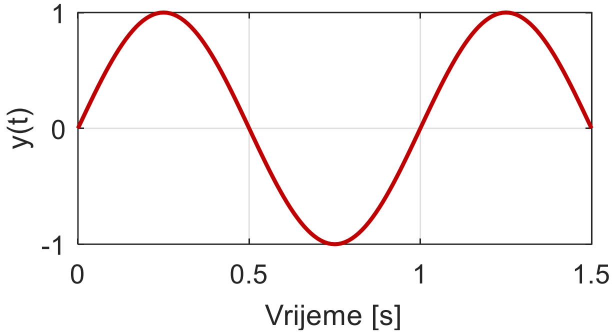
Ishodi učenja:

Nakon savladavanja gradiva sa ovog predavanja studenti će moći da:

- Razumiju razliku između analognog i digitalnog sistema automatskog upravljanja.
- Skiciraju strukturu osnovne regulacione petlje digitalnog SAU-a i prepoznaju ulogu njegovih osnovnih komponenti.
- Diskretizuju i simuluraju kontinualni sistem primjenom nekog od obrađenih diskretizacionih postupaka.

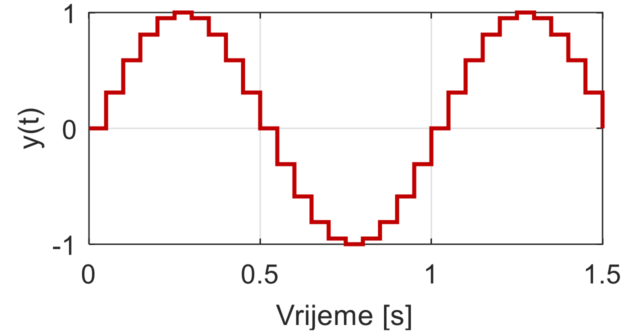
Klasifikacija signala

Kontinualni



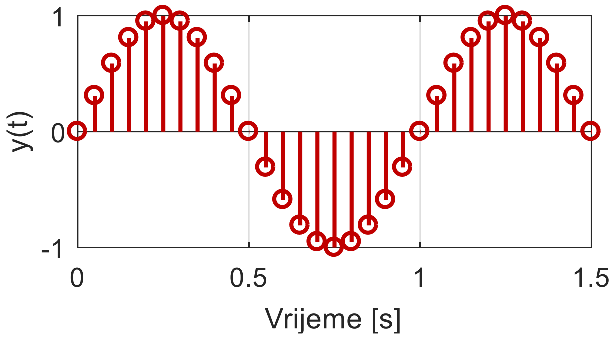
Definisani su u svakom trenutku vremena i mogu imati bilo koju vrijednost amplitude.

Diskretan po amplitudi



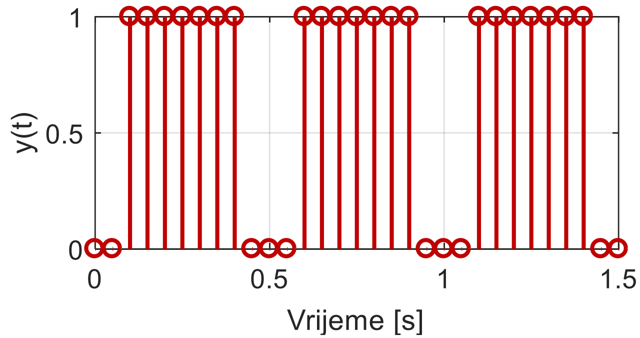
Definisani su u svakom trenutku vremena i mogu imati određene vrijednosti amplitude.

Diskretan po vremenu



Definisani u određenim trenucima vremena i mogu imati bilo koju vrijednost amlitude.

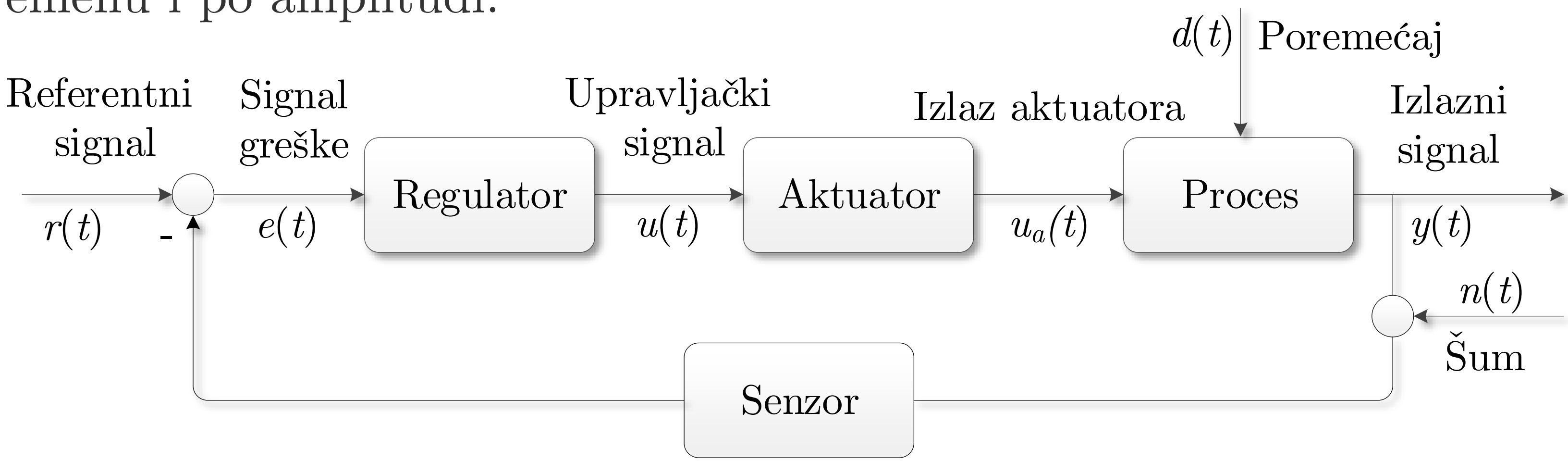
Diskretan po amplitudi i vremenu



Definisani u određenim trenucima vremena i mogu imati određene vrijednosti amlitude.

Kontinualni sistemi upravljanja

Tipična struktura kontinualnog sistema upravljanja je prikazana na slici ispod. Svi signali prikazani na šemi su analogni, odnosno kontinualni i po vremenu i po amplitudi.



Takođe, sve komponente sistema su analogne. Proces, odnosno objekat upravljanja i aktuator su po svojoj prirodi kontinualni sistemi, koji se modeluju pomoću funkcije prenosa ili u prostoru stanja. Regulator se takođe implementira u analognoj tehnici (na primjer korišćenjem komponenti analogne elektronike). Senzor obično daje mjerenja izlazne veličine u vidu električne veličine, dok se referentni signal zadaje preko ulaznog transdjusera.

Kontinualni sistemi upravljanja

Neki od zadataka, odnosno ciljeva upravljanja su:

- Razlika između referentne i izlazne vrijednosti SAU-a treba bude minimizovana

  Greška treba da bude jednaka nuli u stacionarnom stanju, a pored toga treba minimizovati trajanje prelaznog procesa i oscilacije.

- SAU treba da bude robustan na greške u modelovanju sistema

  Greške u modelovanju komponenti SAU su neizbježne. SAU treba dizajnirati tako da nemodelovana dinamika što manje utiče na njegove performanse.

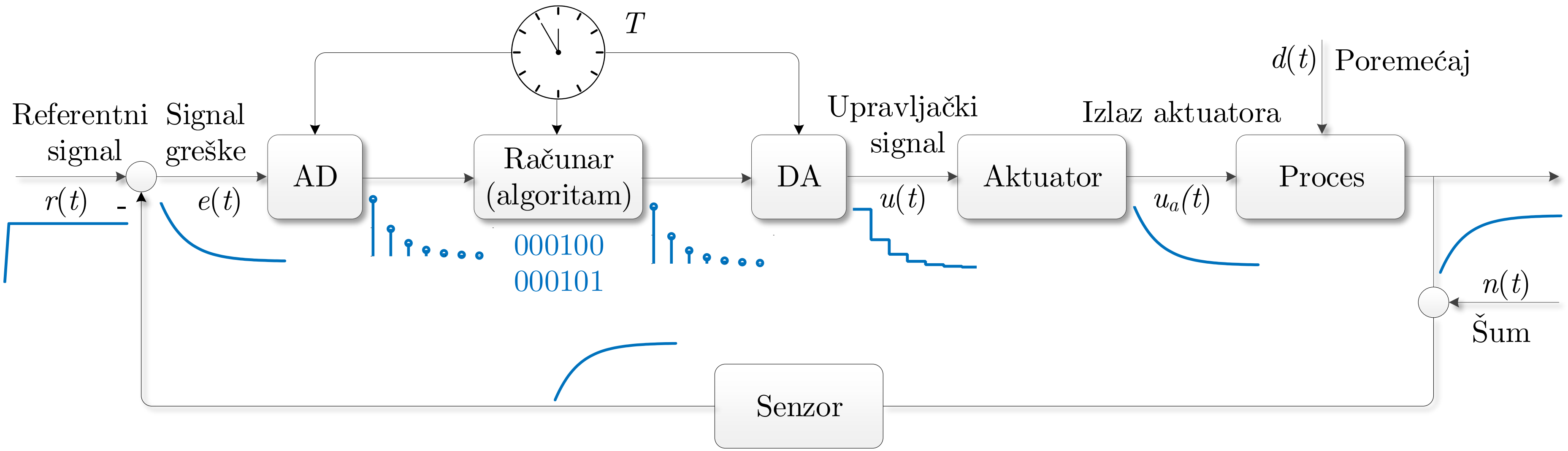
- SAU treba da bude u mogućnosti da suzbija spoljne poremećaje
- Efekti mjernog šuma treba da budu minimizovani

Da bi se našao kompromis između različitih zahtjeva upravljanja, potrebno je dizajnirati odgovarajući zakon/algoritam upravljanja. Dizajn linearnih zakona upravljanja se može vršiti na više načina:

- u frekevencijskom domenu (Bode),
- u s-domenu (root locus metod),
- u prostoru stanja.

Digitalni sistemi upravljanja

Moderni sistemi upravljanja su najčešće digitalni. Kod digitalnih SAU-a upravljački zakon se implementira na računaru/mikrokontroleru.



Kako računar razumije samo digitalne vrijednosti, mjereni signal treba prvo diskretizovati i pretvoriti u digitalni oblik pomoću A/D konvertora. Digitalni signal se obrađuje upravljačkim algoritmom koji kao izlaz daje upravljački signal u digitalnom obliku. Na kraju, upravljački signal se pomoću D/A konvertora pretvara u kontinualni signal i dovodi na ulaz aktuatora.

Digitalni sistemi upravljanja

Dakle, pored standardnih komponenti (aktuator, proces, senzor), digitalni SAU se sastoji iz sklopova koja vrše A/D i D/A konverziju.

Proces konverzije analognog u digitalni signal se sastoji iz dva koraka. U prvom koraku se vrši odabiranje i zadrška analognog signala (sample and hold, S/H), odnosno uzima se odbirak analognog signala u trenutku nT i taj odbirak se drži na izlazu S/H kola jednu periodu, trajanja T. S/H kolo je povezano na A/D konvertor, koji se često naziva i koder. A/D konvertor vrši kvantizaciju odabranog signala i kodira ga tako što ga pretvara u zapis razumljiv računaru (0 i 1). S/H kola mogu biti sastavni dio A/D konvertora, a mogu se dizajnirati kao posebne komponente. Takođe, A/D konvertor može da bude sastavni dio mikrokontrolera na kojem se vrši implementacija upravljačkog algoritma.

Mikrokontroler se obično programira u programskom jeziku višeg nivoa (na primjer C, Python) i njegov zadatak je da u toku periode trajanja T obradi digitalne odbirke signala i da na izlazu generiše kodirani odbirak upravljačkog signala $u(nT)$.

Digitalni sistemi upravljanja

Radi jednostavnijeg modelovanja i analize može se smatrati da mikrokontroler obrađuje *diskretni* signal. Greška koja nastaje uslijed A/D konverzije, odnosno kvantizacije diskretnog signala se može modelovati kao mjerni (kvantizacioni) šum. Najčešće se kvantizacioni šum modeluje normalnom slučajnom raspodjelom, srednje vrijednosti 0 i varijanse $Q^2/12$.

Opseg u kom se kreće signal

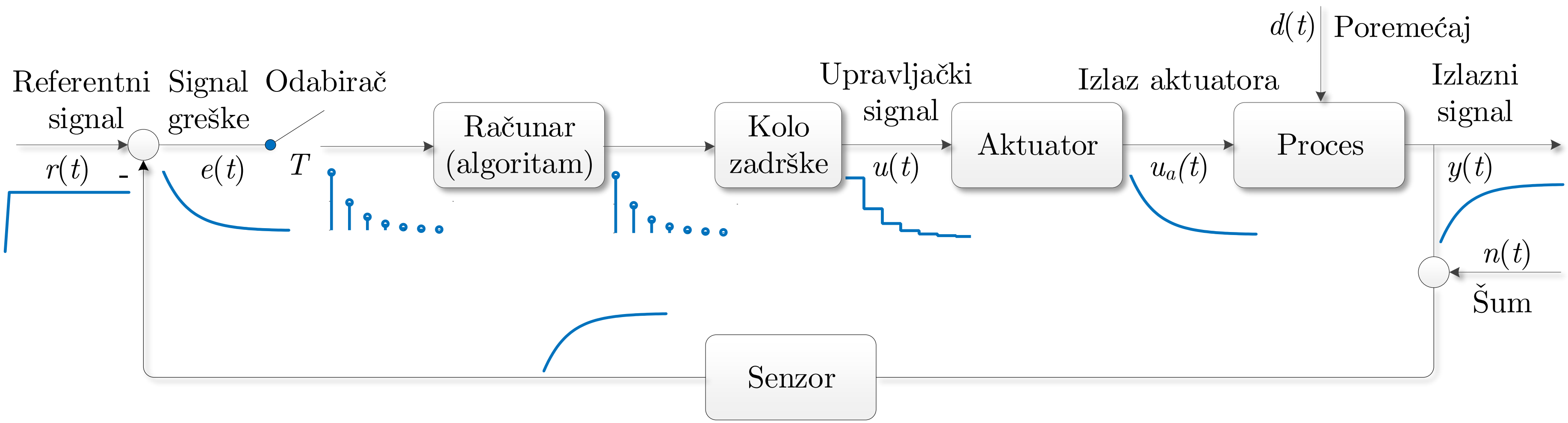
$$n(t) \sim \mathcal{N}(0, \frac{Q^2}{12}). \qquad Q = \frac{y_{max} - y_{\min}}{2^n}$$

Broj bita A/D konvertora

Vrijednost upravljačkog signala, izračunata na mikrokontroleru, se iz digitalnog oblika dekodira u diskretni odbirak pomoću A/D konvertora. Diskretni odbirci upravljačkog signala se pomoću kola zadrške pretvaraju u analogni signal i dalje vode na ulaz aktuatora, odnosno objekta upravljanja. D/A konvertor je aktivna komponenta, tj. ima svoje napajanje. Treba voditi računa o tome da D/A konvertori na svom izlazu ne mogu dati veću vrijednost napona od one kojom se napajaju. Obično su D/A konvertor i kolo zadrške sastavni dio mikrokontrolera.

Digitalni sistemi upravljanja

Na slici ispod je prikazana pojednostavljena šema digitalnog sistema automatskog upravljanja.



U ovoj šemi A/D konvertor je zamijenjen idealnim odabiračem, koji periodično vrši odabiranje analognog signala. Odabrani (diskretni) signal se obrađuje na računaru. D/A konvertor je predstavljen samo kolom za zadršku, koje na svom izlazu zadržava vrijednost zatečenog odbirka jednu periodu odabiranja, sve dok na njegov ulaz ne dođe novi odbirak upravljačkog signala.

Digitalni sistemi upravljanja

Postoje dva načina za dizajniranje diskretnih regulatora:

- Dizajn se može izvršiti u kontinualnom domenu, nakon čega se vrši diskretizacija analognog regulatora odgovarajućim diskretizacionim postupkom. Na ovaj način se ne garantuje stabilnost SAU-a, pa je potrebno simulacijima verifikovati dizajn, odnosno performanse rezultujućeg digitalnog sistema.
- Dizajn se može vršiti direktno u digitalnom domenu. U ovom slučaju potrebno je diskretizovati analogne komponente SAU-a (aktuator i objekat upravljanja) i na taj način čitavu šemu svesti na digitalni domen. Nakon toga, dizajn digitalnog kontrolera se vrši metodama u frekvencijskom, z ili diskretnom vremenskom domenu (može se uspostaviti analogija sa metodama u kontinualnom domenu). Na ovaj način se može postići finije podešavanje performansi SAU-a.

Prilikom sinteze digitalnih regulatora veoma je važno imati razumjevanje o tome kako proces odabiranja signala utuče na performanse SAU-a.

Digitalni sistemi upravljanja

Neke prednosti digitalnih sistema upravljanja su:

- Fleksibilniji su. Veoma je jednostavno modifikovati parametre regulatora ili implementirati novi algoritam.
- Moguće je implementirati nelinearne i sofisticirane zakone upravljanja (implementacija upravljačkog algoritma se svodi na programiranje).
- Spoljašnji faktori ne utiču na parametre regulatora.
- Često su jeftiniji, a i fizički manjih dimenzija (na primjer DSP čip).
- Manje su osjetljivi na mjerne šumove.

Neke loše strane digitalnih SAU-a:

- Sporiji su. Kontinualni regulator gotovo trenutno obrađuje ulazni signal, dok se kod digitalnih SAU-a javlja kašnjenje od jedne periode odabiranja. Vremensko kašnjenje negativno utiče na stabilnost.
- Efekti kvantizacije negativno utiču ne performanse sistema.
- Zahtijevaju konstantno napajanje.

Modelovanje diskretnih sistema

Kao što je već dobro poznato, kontinualni, linearni, vremensko-invarijantni sistemi se u vremenskom domenu modeluju pomoću linearnih diferencijalnih jednačina sa konstantnim koeficijentima. U opštem slučaju dinamika LTI sistema se opisuje jednačinom:

$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1}y}{dt^{n-1}} + \ldots\ldots + a_1\frac{dy}{dt} + a_0 y = b_m\frac{d^m u}{dt^m} + b_{m-1}\frac{d^{m-1}u}{dt^{m-1}} + \ldots\ldots + b_1\frac{du}{dt} + b_0 u.$$

U Laplasovom domenu, LTI sistemi se modeluju pomoću funkcije prenosa, koja predstavlja odnos Laplasovih likova izlaznog i ulaznog signala:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{s^m + b_{m-1}s^{m-1} + \ldots\ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots\ldots + a_1 s + a_0}, m \leq n.$$

Napomenimo da se opšti oblik funkcije prenosa sistema dobija primjenom osobine diferenciranja za nulte početne uslove:

$$\frac{dy^n}{dt^n} \xleftrightarrow{L} s^n Y(s).$$

Modelovanje diskretnih sistema

Na sličan način, diskretni LTI sistemi se modeluju pomoću linearnih diferencnih jednačina sa konstantnim koeficijentima, koje u opštem slučaju imaju sljedeći oblik:

$$
\begin{aligned}
&y(n) + \ldots + a_1 y(n-1) + \ldots + a_{N-1} y(n-N+1) + a_N y(n-N) \\
&= b_0 u(n) + b_1 u(n-1) + \ldots.. + b_{M-1} u(n-M+1) + b_M u(n-M).
\end{aligned}
$$

Posmatrajući prethodnu jednačinu može se uočiti da vrijednost izlaza u trenutku n zavisi od vrijednosti izlaza u prethodnim trenucima, od trenutne vrijednosti ulaza $u(n)$, kao i prethodnih vrijednosti ulaznog signala. Ovo zapravo znači da sistem posjeduje neku dinamiku, odnosno da njegov odziv evoluira tokom vremena. Za razliku od kontinualnih sistema koji se modeluju i u s domenu, alternativni domen za opisivanje diskretnih sistema je z-domen, odnosno funkcija diskretnog prenosa koja se definiše kao odnos Z-transformacija izlaznog i ulaznog signala:

$$G(z) = \frac{Y(z)}{X(z)}.$$

Modelovanje diskretnih sistema

Da bi odredili funkciju diskretnog prenosa, koristićemo sljedeću osobinu Z-transformacije:

$$y(n-M) \xleftrightarrow{L} z^{-M}Y(z).$$

Polazeći od opšteg oblika diferencne jednačine i koristeći prethodnu osobinu, dobija se opšta forma funkcije diskretnog prenosa :

$$G(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \ldots + b_{M-1} z^{-M+1} + b_M z^{-M}}{1 + a_1 z^{-1} \ldots + a_{N-1} z^{-N+1} + z^{-N}}$$
$$= z^{N-M} \frac{b_0 z^M + b_1 z^{M-1} + \ldots + b_{M-1} z + b_M}{z^N + a_1 z^{N-1} \ldots + a_{N-1} z + 1}.$$

Primijetimo da je funkcija diskretnog prenosa racionalna funkcija. Na nekom od narednih predavanja biće više rečeno o Z-transformaciji, njenim najvažnijim osobinama, o potrebi njenog uvođenja i njenoj vezi sa Laplasovom transformacijom.

Modelovanje diskretnih sistema

Neki procesi su diskretni po svojoj prirodi. Primjer takvog procesa bi bio broj studenata koji su prijavili neki ispit u tekućoj školskoj godini, koji bi se mogao opisati sljedećom diferencnom jednačinom:

$$y(n) = ay(n-1) + u(n).$$

U prethodnoj jednačini signal $u(n)$ označava broj novoupisanih studenata, $ay(n\text{-}1)$ predstavlja broj studenta iz prethodne generacije, dok $y(n)$ predstavlja ukupan broj studenata u tekućoj godini.

Međutim, digitalni sistemi upravljanja su najčešće hibridni. Objektat kojim se upravlja je kontinualan, dok se regulator implementira u digitalnom domenu. Da bi vršili analizu i sintezu hibridnog SAU-a, čitav sistem moramo posmatrati u jednom domenu. Regulatori se mogu dizajnirati u kontinualnom domenu, a zatim ih je potrebno diskretizovati, odnosno transformisati u digitalni domen. Slično, dizajn regulatora je moguće izvršiti u digitalnom domenu, ali je prethodno potrebno odrediti diskretni ekvivalent objekta upravljanja.

Modelovanje diskretnih sistema

Takođe, da bi vršili simulaciju sistema upravljanja na računaru (kontinualnih ili diskretnih), potrebno je izvršiti diskretizaciju svih komponenti sistema.

Postoji više pristupa diskretizaciji kontinualnog sistema, a ovdje će biti opisano nekoliko njih. U suštini, za dovoljno malu vrijednost perode odabiranja svaka od metoda će dati slične rezultate. Ukoliko je perioda odabiranja prevelika, može se desiti da diskretni ekvivalent čak bude i nestabilan. Uticaj periode odabiranja na performanse SAU-a biće analiziran na nekom od narednih predavanja. Periodu odabiranja ne treba nepotrebno smanjivati, jer to zahtijeva brži takt mikroprocesora, a samim tim i skuplju implementaciju SAU-a. Periodu odabiranja treba odabrati tako da diskretni ekvivalent dobro „uhvati“ dinamiku kontinualnog sistema, a pored toga da u stacionarnom stanju ima isti odziv kao kontinualni sistem.

Treba napomenuti da metode koje će biti prezentovane dobro oponašaju LTI sisteme, dok je za simulaciju nelinearnih sistema često potrebno koristiti sofisticiranije numeričke metode diferenciranja i integraljenja.

Numerička diferencijacija

Polazeći od definicije prvog izvoda u trenutku $t{=}nT$:

$$\frac{dy(nT)}{dnT} = \lim_{\Delta \to 0} \frac{y(nT+\Delta) - y(nT)}{\Delta},$$

pravila numeričke diferencijacije se intuitivno izvode. Kod **diferenciranja unazad** (eng. backward difference), izvod se aproksimira razlikom između odbirka u trenutku nT i odbirka u prethodnom trenutku $(n\text{-}1)T$:

$$\frac{dy(nT)}{dnT} \approx \frac{y(nT) - y(nT-T)}{T}.$$

Sa druge strane, kod **diferenciranja unaprijed** (eng. forward difference), izvod se aproksimira razlikom između odbiraka u trenucima $(n{+}1)T$ i nT:

$$\frac{dy(nT)}{dnT} \approx \frac{y(nT+nT) - y(nT)}{T}.$$

Numerička diferencijacija

Konačno, kod metoda **centralne razlike** prvi izvod se aproksimira pomoću razlike između odbiraka u trenucima $(n+1)T$ i $(n\text{-}1)T$:

$$\frac{dy(nT)}{dnT} \approx \frac{y(nT+nT)-y(nT-nT)}{2T}.$$

Jasno da tačnost svih izloženih postupka zavisi od vrijednosti periode odabiranja T. Što je perioda odabiranja manja, to će aproksimacija izvoda biti bolja.

Ukoliko je sistem zadat u obliku funkcije prenosa, tada možemo direktno uvesti smjenu promjenljive s na neki od sljedećih načina:

$$s \leftarrow \frac{1-z^{-1}}{T} \text{(diferenciranje unazad)},$$

$$s \leftarrow \frac{z-1}{T} \text{ (diferenciranje unaprijed)},$$

$$s \leftarrow \frac{z-z^{-1}}{2T} \text{ (centralna razlika)}.$$

Numerička diferencijacija

Postupak aproksimacije viših izvoda je analogan aproksimaciji prvog izvoda. Drugi izvod se može zapisati u funkciji od prvog izvoda:

$$\frac{d^2y(nT)}{dnT^2} \simeq \frac{y'(nT) - y'(nT-\Delta)}{T} = \frac{y(nT) - 2y(nT-\Delta) + y(nT-2\Delta)}{T^2}.$$

Tačnost diferenciranja unazad se može analizirati razvojem funkcije $y(nT\text{-}T)$ u Tejlorov red u okolini tačke nT:

$$y(nT-T) = y(nT) - y'(nT)T + \frac{y''(nT)}{2}T^2 + \cdots,$$

odakle se dobija:

$$y'(nT) \approx \frac{y(nT) - y(nT-T)}{T} + T\frac{y''(nT)}{2} = \frac{y(nT) - y(nT-T)}{T} + O(T).$$

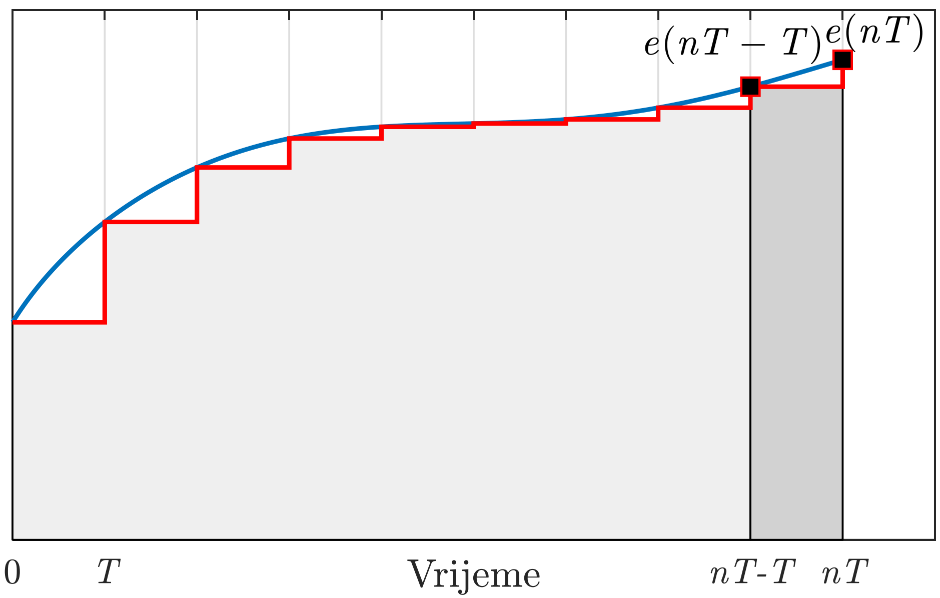
Član $O(T)$ znači da je greška u aproksimaciji reda T, odnosno da greška linearno raste sa povećanjem periode odabiranja. Na sličan način se može pokazati da je greška kod diferenciranja unaprijed takođe reda $O(T)$, dok je metod centralne razlike najtačniji i ima grešku reda $O(T^2)$.

Numerička integracija

Pravila za numeričku integraciju se izvode na sličan način:

$$i_n(t) = \int_0^t e(\tau)d\tau \longrightarrow i_n(nT) = \int_0^{nT} e(t)dt = \int_0^{nT-T} e(t)dt + \int_{nT-T}^{nT} e(t)dt$$
$$= i_n(nT-T) + Te(nT-T).$$

U prethodnom izrazu površina krive $e(t)$ na intervalu $[nT\text{-}T, nT)$ je aproksimirana pravougaonikom čije su stranice T i $e(nT\text{-}T)$, što je ilustrovano na slici ispod.



Ovo pravilo integraljenja je zapravo analogno pravilu diferenciranja unazad. Kao stranica pravougaonika se mogao posmatrati odbirak vrijednosti $e(nT)$. Ovo pravilo integraljenja se svodi na pravilo diferenc. unaprijed.

Numerička integracija

Površina funkcije $e(t)$ na intervalu $[nT\text{-}T, nT)$ se preciznije može aproksimirati trapezom visina $e(nT\text{-}T)$ i $e(nT)$:

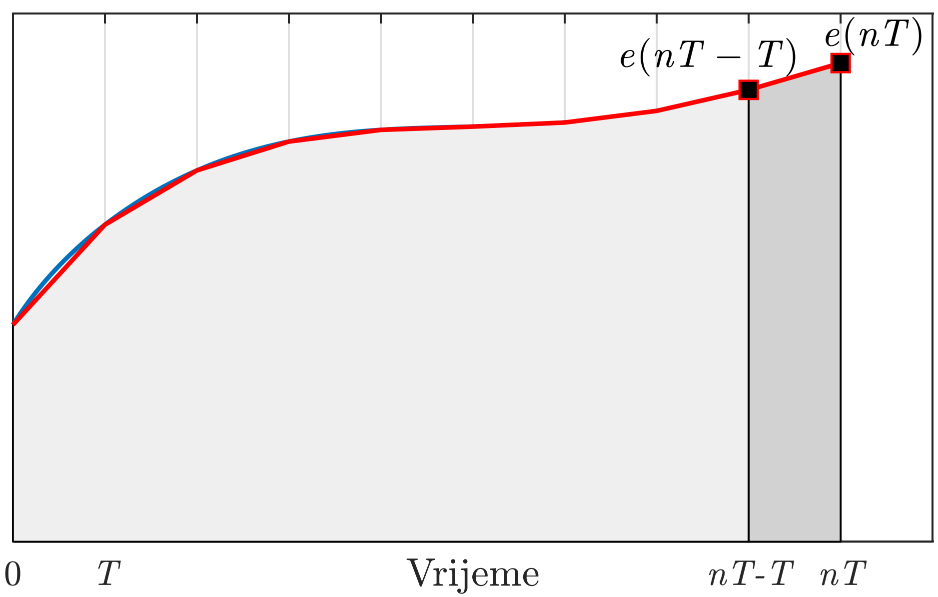
$$i_n(nT) = \int_0^{nT} e(t)dt = \int_0^{nT-T} e(t)dt + \int_{nT-T}^{nT} e(t)dt$$

$$= i_n(nT-T) + T\,\frac{e(nT-T)+e(nT)}{2}.$$

$$I_n(z) = I_n(z)z^{-1} + \frac{T}{2}E(z)\left(1+z^{-1}\right)$$

$$\rightarrow I_n(z) = \frac{T}{2}\,\frac{1+z^{-1}}{1-z^{-1}}\,E(z)$$

Ovo pravilo je u literaturi poznato pod nazivom trapezoidno pravilo, bilinearna transformacija ili Tustinova transformacija.



Na osnovu gornje jednačine dobija se sljedeća smjena za promjenljivu s:

$$s \leftarrow \frac{2}{T}\,\frac{1-z^{-1}}{1+z^{-1}}.$$

Tustinovo pravilo je najtačnije od pomenutih metoda, što se zapravo može i uočiti sa slike.

Diskretizacija PID regulatora

Posmatrajmo upravljački zakon PID regulatora:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}.$$

Prilikom diskretizacije kontinualnih regulatora treba voditi računa o tome da rezultujući diskretni regulator bude kauzalan, kako bi se mogao realizovati u realnom vremenu. Ukoliko primijenimo postupke diferenciranja/integraljenja dobićemo diskretnu verziju PID-a:

$$u(n) = K_p e(n) + K_i \underbrace{\left(i_n(n-1) + Te(n)\right)}_{i_n(n)} + K_d / T \left(e(n) - e(n-1)\right). \qquad \boxed{n \triangleq nT}$$

U prethodnom izrazu se pojavljuje promjenljiva $i_n(n\text{-}1)$, koje se treba osloboditi. Nju možemo odrediti na osnovu izraza za upravljački signal u trenutku $(n\text{-}1)T$:

$$u(n-1) = K_p e(n-1) + K_i i_n(n-1) + K_d / T \left(e(n-1) - e(n-2)\right)$$

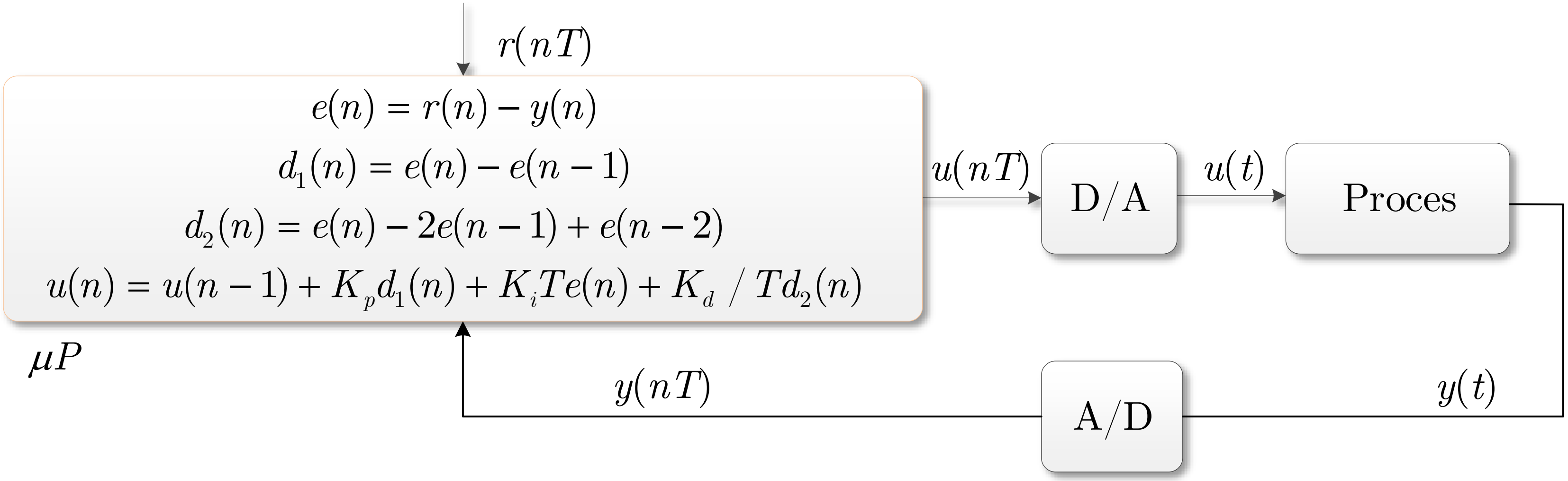
$$\rightarrow i_n(n-1) = \frac{1}{K_i}\left(u(n-1) - K_p e(n-1) - \frac{K_d}{T}\left(e(n-1) - e(n-2)\right)\right).$$

Diskretizacija PID regulatora

Uvrštavanjem $i_n(n\text{-}1)$ u početni izraz dobija se sljedeća rekurzivna jednačina za računanje upravljačkog signala:

$$u(n) = u(n-1) + K_p\big(e(n) - e(n-1)\big) + K_i T e(n) + \frac{K_d}{T}\big(e(n) - 2e(n-1) + e(n-2)\big).$$

Posmatrajući prethodni izraz može se zaključiti da je za računanje upravljačkog signala u tekućem trenutku, u memoriji računara potrebno sačuvati vrijednosti signala greške u prethodna dva trenutka, kao i vrijednost upravljačkog signala u prethodnom trenutku vremena.



Diskretizacija PID regulatora

Do rekurzivne formule za PID regulator se jednostavnije može doći polazeći od funkcije prenosa PID-a:

$$U(s) = \left(K_p + \frac{K_i}{s} + K_d s \right) E(s).$$

Uvodeći smjenu $s=(1\text{-}z^{-1})/T$ dobijaju se sljedeći izrazi:

$$U(z) = \left(K_p + \frac{TK_i}{1 - z^{-1}} + \frac{K_d}{T}\left(1 - z^{-1}\right) \right) E(z)$$

$$U(z)(1 - z^{-1}) = K_p(1 - z^{-1})E(z) + TK_i E(z) + \frac{K_d}{T}(1 - z^{-1})^2 E(z)$$

Konačno, zadnji izraz se može zapisati u formi diferencne jednačine:

$$u(n) = u(n-1) + K_p\big(e(n) - e(n-1)\big) + K_i T e(n) + \frac{K_d}{T}\big(e(n) - 2e(n-1) + e(n-2)\big).$$

Diskretizacija integralnog kompenzatora

Na sličan način se vrši diskretizacija integralnog kompenzatora:

$$G_i(s) = \frac{U(s)}{E(s)} = K\,\frac{\dfrac{s}{b}+1}{\dfrac{s}{a}+1}.$$

Funkcija prenosa integralnog kompenzatora se može zapisati u vidu diferencijalne jednačine, a zatim se primjenjuju pravila diferenciranja. Mada, možemo direktno zamijeniti promjenljivu s odgovarajućim pravilom:

$$G_i(z) = \frac{U(z)}{E(z)} = K\,\frac{\dfrac{1-z^{-1}}{bT}+1}{\dfrac{1-z^{-1}}{aT}+1} = K\,\frac{a}{b}\,\frac{1+bT-z^{-1}}{1+aT-z^{-1}}.$$

Odnosno, upravljački signal se generiše na sljedeći način:

$$u(n) = \frac{1}{1+aT}\,u(n-1) + K\,\frac{a}{b}\,\frac{1+bT}{1+aT}\,e(n) - K\,\frac{a}{b}\,\frac{1}{1+aT}\,e(n-1).$$

Primjer – diskretizacija procesa

Kontinualni sistem je opisan sljedećom diferencijalnom jednačinom:

$$\dot{y}(t) + 2y(t) = u(t).$$

Usvajajući da je perioda odabiranja T=1s, diskretizovati dati sistem. Uporediti step odzive diskretnog i kontinualnog sistema.

U ovom primjeru ćemo koristiti diferenciranje unazad. Kod metoda diferenciranja unazad vremenski trenutak t se zamjenjuje diskretnim vremenom nT, gdje n predstavlja redni broj odbirka, a T vrijeme odabiranja. Radi jednostavnosti zapisa, nT ćemo označiti sa n. Kod diferenciranja unazad prvi izvod se aproksimira na sljedeći način:

$$\dot{y}(nT) \approx \frac{y(nT) - y(nT - T)}{T} \triangleq \frac{y(n) - y(n-1)}{T}.$$

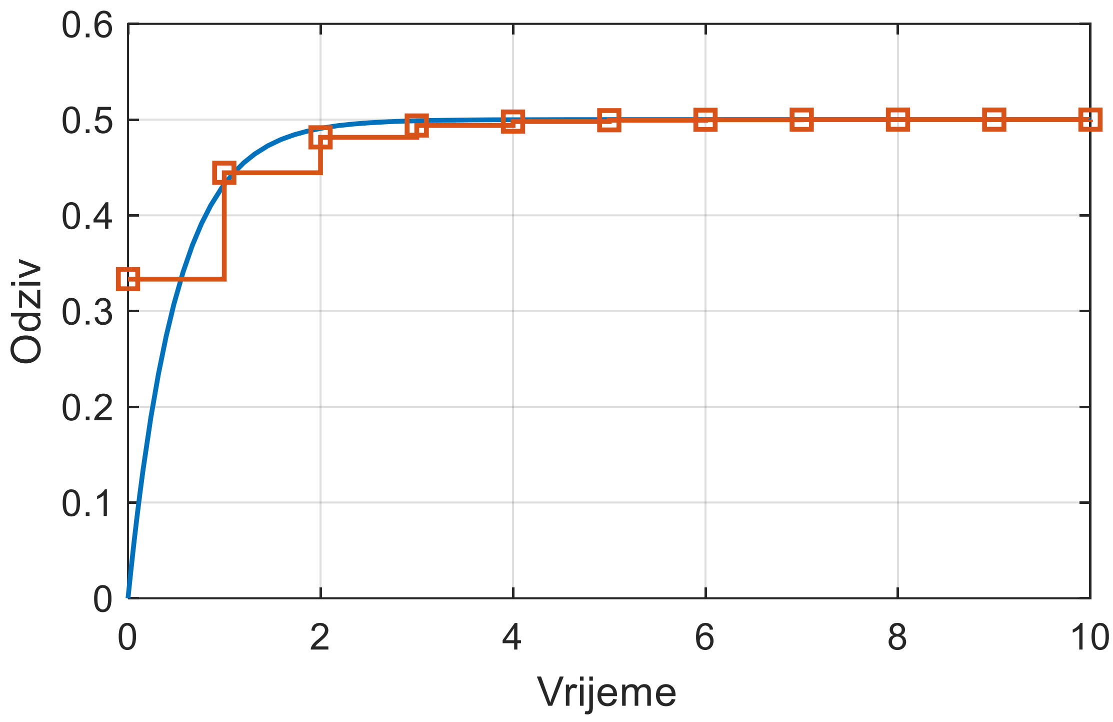
Uvrštavajući prethodnu aproksimaciju u polaznu jednačinu, dobija se sljedeća diferencna jednačina:

$$\frac{y(n) - y(n-1)}{T} + 2y(n) = u(n).$$

$$\rightarrow y(n) = \frac{1}{1+2T} y(n-1) + \frac{T}{1+2T} u(n) = \frac{1}{3} y(n-1) + \frac{1}{3} u(n).$$

Primjer – diskretizacija procesa

Prethodna rekurzivna jednačina se može riješiti u zatvorenom obliku. Međutim, u ovom primjeru ćemo je riješiti numeričkim putem u Matlab-u. Na slici ispod je prikazan step odziv kontinulanog i diskretnog sistema. Može se uočiti da su u stacionarnom stanju odzivi oba sistema jednaka. Sa druge strane, postoji greška u aproksimaciji prelaznog procesa, koja je posljedica velike periode odabiranja.



Primjer – diskretizacija procesa

```
clear y
close all
N=11; % broj trenutaka vremena
T=1; % perioda odabiranja
y(1)=1/3*1; % ovo je zapravo y(0), samo u mat indeksiramo od 1
for n=2:N % pocinjemo od 2, jer se u jedncini pojavljuje y(n-1)
    y(n)=1/(1+2*T)*y(n-1)+T/(1+2*T)*1; % ulazni signal je step funkcija
end
vrijeme = [0 10]; % od nula do 10s
y0 = 0; % pocetnu uslov
[t,yc] = ode45(@(t,y) -2*y+1, vrijeme, y0);
plot(t,yc,'linewidth',1.2)
hold on
stairs([0:N-1]*T,y,'-s','linewidth',1.2); % mnozimo odbirke sa T, da bi
pravilno prikazali vremensku skalu
grid
axis([0 10 0 0.6])
xlabel('Vrijeme')
ylabel('Odziv')
% kont. sistem se mogao simulirati i tako sto se odredi funkcija prenosa i
% nacrta step odziv
% s=tf('s')
% G=1/(s+2)
% step(G)
```

Primjer – diskretizacija procesa

Na slici ispod je prikazana realizacija kontinualnog i diskretnog sistema u Simulink-u. Blokovi koji su korišćeni za realizaciju su: *transfer function*, *discrete transfer function*, *step function*, *scope* i *to workspace* (kucati u search). Odziv *yc* se u radnom prostoru može nacrtati na sljedeći način:

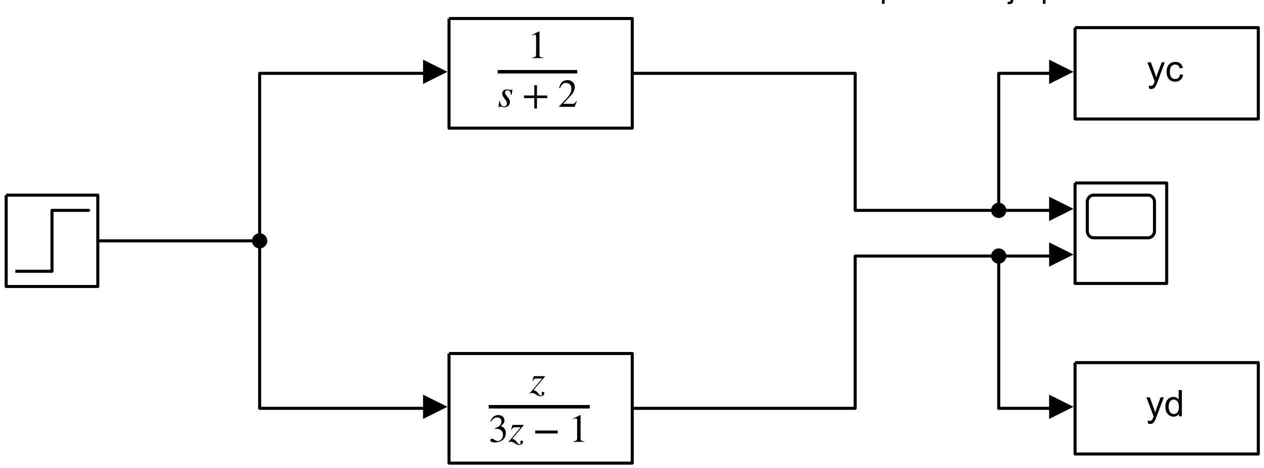
$$y(n) = \frac{1}{3} y(n-1) + \frac{1}{3} u(n)$$

$$Y(z) = \frac{1}{3} z^{-1} Y(z) + \frac{1}{3} U(z)$$

$$\rightarrow \frac{Y(z)}{U(z)} = \frac{1/3}{1 - z^{-1}/3} = \frac{z}{3z-1}$$

```
>> plot(yc.Time,yc.Data)
>> yc
timeseries
  Common Properties:
            Name: ''
            Time: [57x1 double]
        TimeInfo: [1x1 tsdata.timemetadata]
            Data: [57x1 double]
        DataInfo: [1x1 tsdata.datametadata]
```

Export-ovanje podataka u radni prostor (u varijablu yc)



Napomena: obavezno podesiti periodu odabiranja u ovom bloku

Primjer – diskretizacija procesa

```
close all
stairs([0:N-1]*T,y,'-s','markerfacecolor','b','linewidth',1.2); % odziv dobijen preko
koda
hold on
stairs(yd.time,yd.data,'-o','markersize',8,'linewidth',1.2); % odziv iz Simulinka
grid
axis([0 10 0 0.6])
xlabel('Vrijeme')
ylabel('Odziv')
legend('Kod','Simulink')
```

Na slici desno su prikazani odziv diskretnog sistema dobijen pomoću Matlab koda, kao i odziv iz Simulinka. Može se uočiti da su vrijednosti diskretnih odbiraka identične. Simulink i svi ostali softverski alati zapravo i kontinualne sisteme simuliraju primjenom nekog diskretizacionog postupka.