

# DIGITALNE TELEKOMUNIKACIJE

PRAKTIKUM ZA LABORATORIJSKE VEŽBE

**Student:**

Ime i prezime: \_\_\_\_\_

Broj indeksa: \_\_\_\_\_

**Profesor:**

dr Nikola Sekulović

**Asistent:**

Nikola Milutinović

# SADRŽAJ

Vežba 1. Uvod u Simulink.....	1
Uvod.....	1
Startovanje Simulinka.....	1
Praktični deo .....	4
Primer 1.....	5
Vežba 2. Odmeravanje signala .....	8
Teorijski deo .....	8
Praktični deo .....	9
Zadatak za samostalni rad.....	13
Vežba 3. Uniformna kvantizacija .....	16
Teorijski deo .....	16
Praktični deo .....	17
Zadatak za samostalni rad.....	21
Vežba 4. Simulacija generatora pseudoslučajnog niza.....	23
Teorijski deo .....	23
Praktični deo .....	24
Vežba 5. Linearni linijski kodovi .....	28
Teorijski deo .....	28
Praktični deo .....	29
Simulacija kodera .....	29
Simulacija dekodera.....	32
Vežba 6. Amplitudska, frekvencijska i fazna modulacija .....	34
Teorijski deo .....	34
6.1. Amplitudska modulacija .....	34
Praktični deo .....	35
Teorijski deo .....	38
6.2. Frekvencijska modulacija .....	38
Praktični deo .....	39
Teorijski deo .....	43
6.3. Fazna modulacija .....	43
Praktični deo .....	44
Vežba 7. Digitalna binarna amplitudska, frekvencijska i fazna modulacija.....	48
Teorijski deo .....	48
Praktični deo .....	49
BASK – Binarna amplitudska digitalna modulacija.....	51

BPSK – Binarna fazna digitalna modulacija.....	51
BFSK – Binarna frekvencijska digitalna modulacija.....	52



## Vežba 1. Uvod u Simulink

### Uvod

Simulink je programski paket za modelovanje i simulaciju sistema. Predstavlja proširenje MATLAB-a i kao takav zadržava sve karakteristike MATLAB-a, jednog od najboljih matematičkih paketa za PC platformu. Osim toga Simulink sadrži i mnoge specifične karakteristike potrebne za brzu i laku simulaciju dinamičkih sistema.

Simulink nudi tri mogućnosti za opisivanje modela koje se mogu kombinovati:

- opis modela korišćenjem viših programskih jezika
- opis modela pomoću Matlab funkcija
- grafičko predstavljanje modela pomoću blokova.

Najpopularniji način za većinu neiskusnih korisnika je korišćenje blokova. Ovaj način je ujedno i najpogodniji i najbrži za editovanje modela do postizanja željenih rezultata. Radi lakšeg snalaženja, blokovi su razvrstani u gotove biblioteke koje se mogu dopuniti blokovima koje formira sam korisnik. Implementacija hijerarhije modela je veoma važna osobina Simulink-a, naročito kod velikih i složenih sistema. U takvim slučajevima se grupa blokova objedinjuje i u šemi predstavlja kao jedan blok što veoma utiče na preglednost modela. Proces simulacije može se pratiti vizuelno, a kada se simulacija završi rezultati su dostupni u radnom okruženju MATLAB-a.

Uz pomoć Simulinka se može vršiti analiza ne samo linearnih modela, već se mogu analizirati realistični nelinearni modeli i razmatrati uticaj trenja, otpor vazduha, brzina klizanja, nagla kočenja, lepršanje krila aviona, efekat monetarnog dohotka u ekonomiji kao i druge stvari koje opisuju realni fenomeni. Simulink personalni računar pretvara u pravu laboratoriju za modelovanje i analizu sistema koje često iz praktičnih razloga nismo u stanju da realizujemo.

Simulink obezbeđuje grafički interfejs (eng. *Graphical user interface - GUI*) za izgradnju modela kao što su blok dijagrami, što nam omogućava da nacrtamo modele kao što bi to uradili olovkom na papiru. Simulink takođe podrazumeva obimnu biblioteku blokova kao što su, izvori, nelinearne i linearne komponente, vizulizatori, konektori... Ako ovi blokovi ne zadovoljavaju naše potrebe, možemo kreirati svoje blokove.

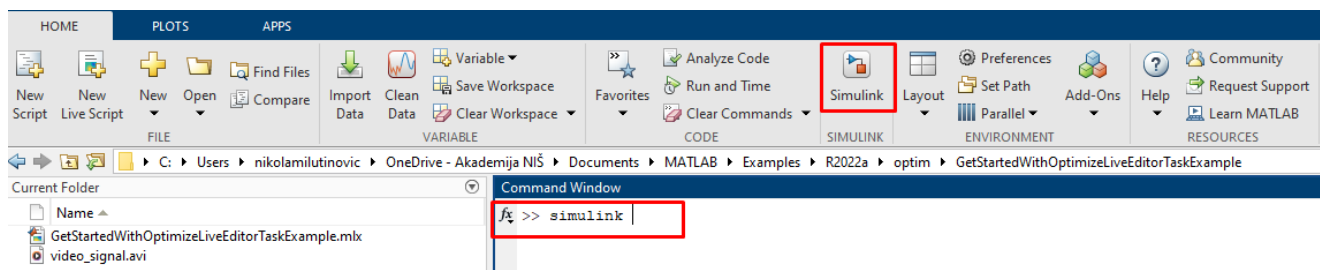
### Startovanje Simulinka

Pokrenuti program MATLAB (Slika 1), a kada se pojavi komandni prozor MATLAB-a, Simulink se može pokrenuti na jedan od sledeća dva načina (Slika 2):

- U MATLAB prozoru otkucajte komandu *simulink* i pritisnite Enter.
- Simulink biblioteka se može otvoriti pritiskom na ikonicu za njeno pokretanje u MATLAB *Toolstrip*-u.

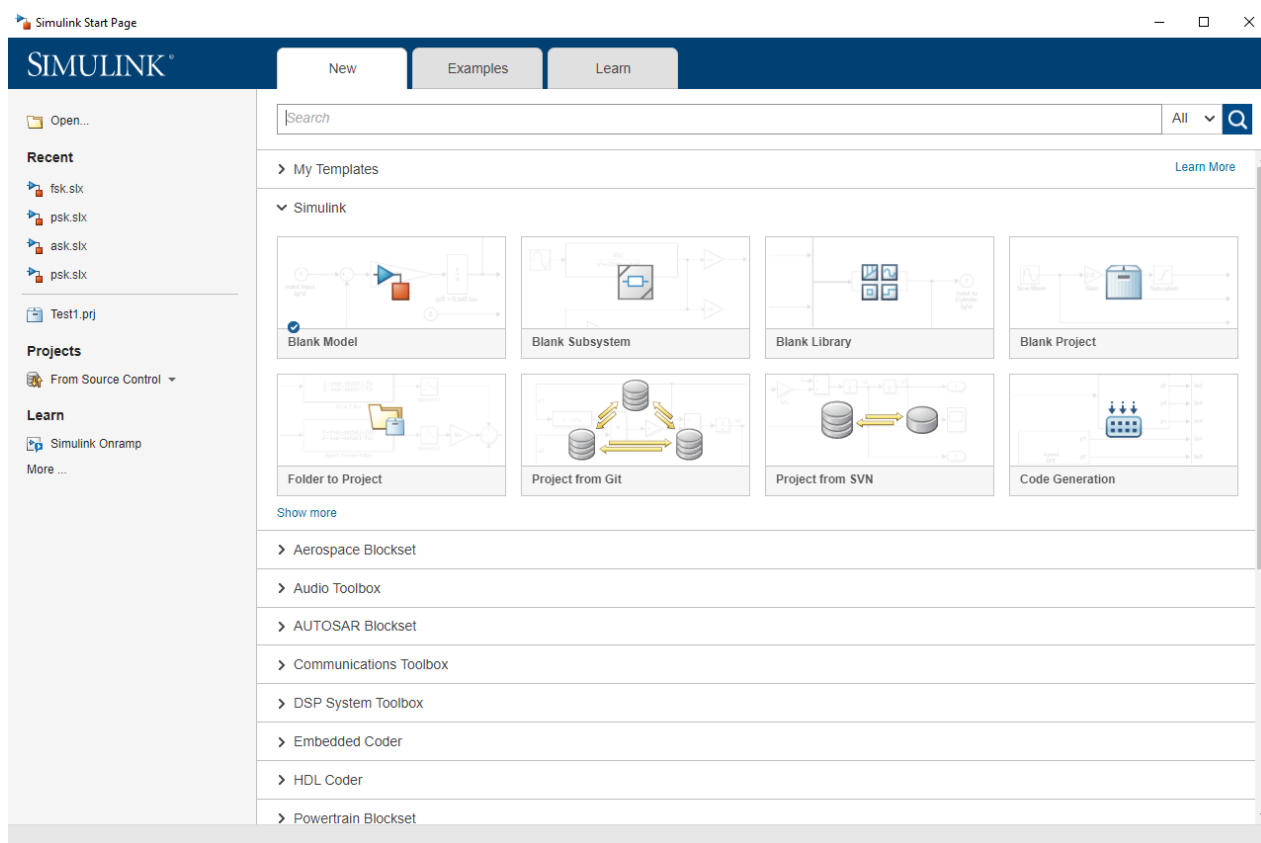


*Slika 1.1. Izgled MATLAB prečice na desktop-u*



*Slika 1.2. Standardni izgled MATLAB radne površine i pokretanje Simulink-a*

Nakon pozivanja otvara se novi prozor u kome je potrebno selektovati opciju *Blank Model*, nakon čega se otvara radna površina za rad.

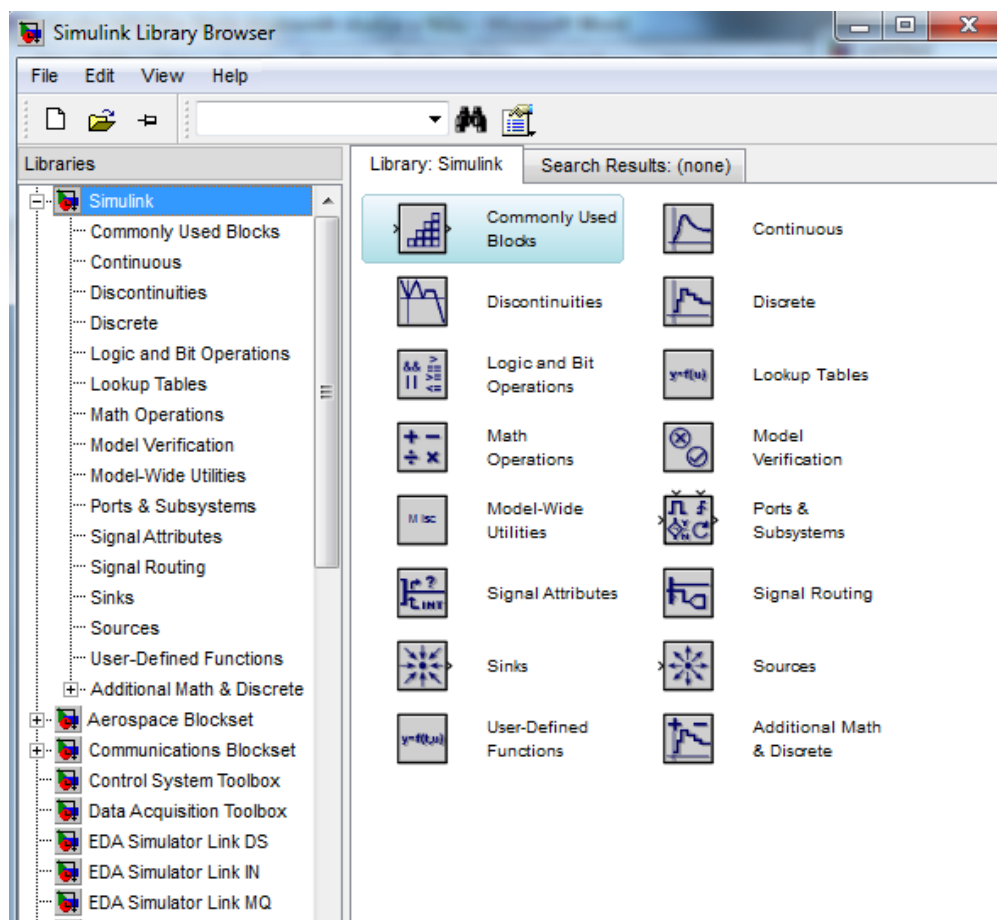


*Slika 1.3. Pokretanje Simulink radne površine*

Ukoliko vam se Simulink biblioteka (slika 1.4) već nije učitala budite strpljivi, doći će do kratkotrajnog kašnjenja. Podešavanje da Simulink biblioteka bude iznad svih ostalih prozora na

vašem PC-u, u biblioteci pod padajućim menijom *View* čekirajte opciju “Ostani na vrhu” (eng. *Stay on top*). Biblioteka sadrži preko dvestotine implementiranih blokova razvrstanih kroz podbibliotečke jedinice kao što su:

- najčešće korišćeni blokovi,
- kontinualni (integrator, diferenciator, idr),
- diskontinualni,
- diskretni,
- logičke i bitske operacije,
- matematičke operacije,
- izvori...



*Slika 1.4. Simulink biblioteka*

U Simulink biblioteci blokova možete učiniti sledeće stvari:

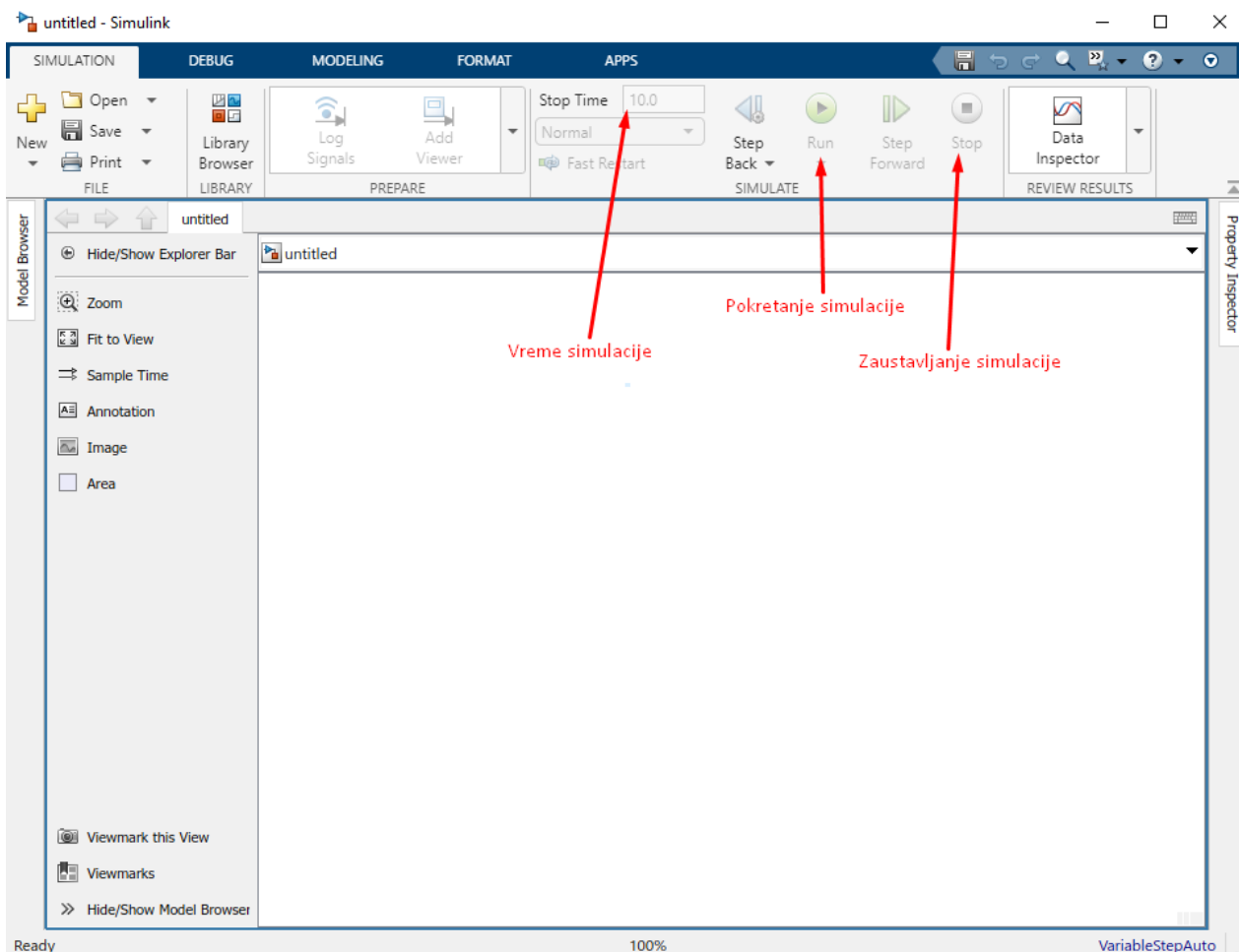
Zadatak	Šta uraditi
<b>Pregled blokova u biblioteci</b>	U levom panelu, selektujte ime biblioteke, dupli klik na ikonu biblioteke
<b>Pretraga blokova</b>	U polju za pretragu blokova ukucajte ime bloka i nakon toga kliknite ikonu.
<b>Informacije o bloku</b>	Selektujte <i>View &gt; Show Block Description</i> i nakon toga selektujte blok
<b>Detaljne informacije o bloku</b>	Selektujte blok, nakon toga selektujte <i>Help &gt; Help for Selected Block</i> . Otvoriće se prozor za pomoć sa stranom koja se odnosi na selektovani blok.
<b>Videti parametre bloka</b>	Desni klik na blok, nakon toga selektujte <i>Block parameters</i>
<b>Kopiranje bloka iz biblioteke na radnu površinu</b>	Jednostavno prevucite blok iz biblioteku na prozor radne površine

## Praktični deo

Kreiranje novog modela iz Simulink biblioteke možete učiniti na jedan od sledeća dva načina:

1. U *File* padajućem meniju kliknite opciju *New*
2. Klikom na ikonu u gornjem levom uglu Simulink biblioteke.

Kada ovo učinite na vaš desktop pojaviće se novootvorena radna površina na kojoj možete kreirati svoj model (Slika 1.5).



*Slika 1.5. Radna površina za kreiranje novog modela*

Prozor za rad sastoji se od blanko površine na koju se blokovi unose prostim prevlačenjem iz biblioteke na radnu površinu. Na radnoj površini možemo uočiti par bitnih funkcija kao što su:

- *Run i Stop* (za pokretanje i zaustavljanje simulacije)
- *Simulation stop time* (vremenski interval koji određuje koliko će simulacija trajati)

### **Primer 1**

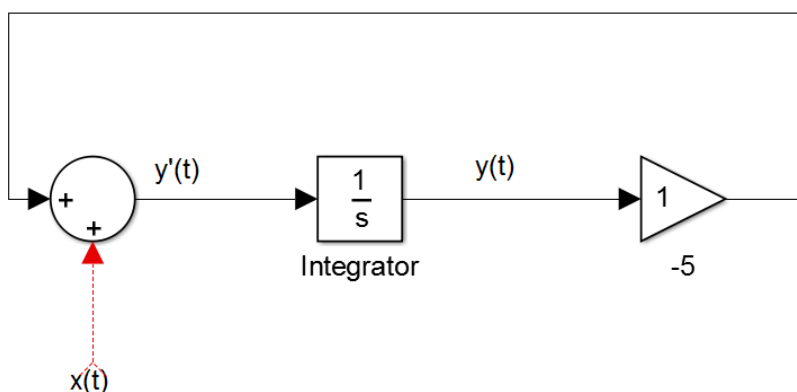
Kao primer rada u Simulink-u uzećemo rešavanje diferencijalne jednačine kojom se modeluje električni sistem

$$\frac{dy(t)}{dt} + 5y(t) = x(t), \quad x(t) = u(t), \quad y(0) = -2$$

U datoj jednačini  $x(t)$  predstavlja ulaz sistema, dok  $y(t)$  predstavlja izlaz iz sistema. Rešavanjem diferencijalne jednačine dolazimo do sledećeg rešenja:

$$y(t) = 0.2(1 - 11\exp(-5t))u(t)$$

Iznad prikazana matematička diferencijalna jednačina može biti reprezentovana kao sistemski model napravljen od blokova u Simulinku. Model takvog sistema prikazan je na slici 1.6.



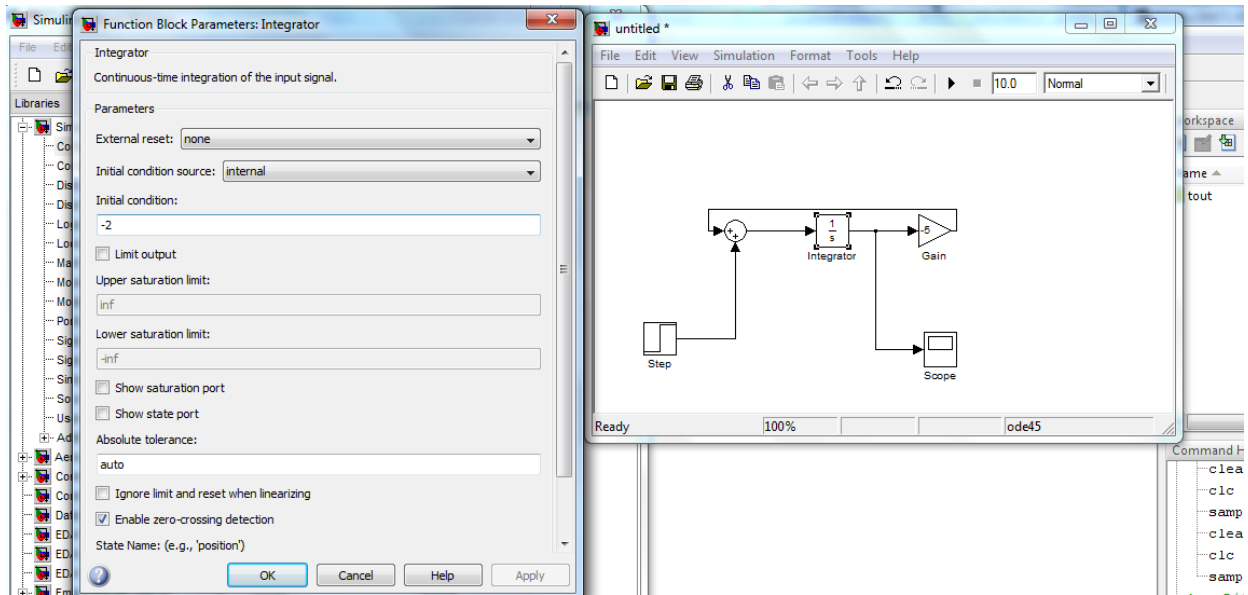
**Slika 1.6.** Blok šema diferencijalne jednačine

Korišćenjem jednostavnog *drag and drop* principa formiraćemo od blokova sistemski model za simulaciju ove matematičke jednačine. Nakon povezivanja blokova dobiće se model prikazan na slici 1.6 nad kojim bi sada trebalo izvršiti određena podešavanja.

Od blokova koristimo sledeće:

- sabirač (*Sum*), integrator (*Integrator*) i pojačavač (*Gain*) koji se nalaze u podbiblioteci *Comonly Used Blocks*
- osciloskop (*Scope*) koji se nalazi u podbiblioteci *Sinks* i
- izvor Hevisajdove funkcije (*Step*) koji se nalazi u podbiblioteci *Sources*

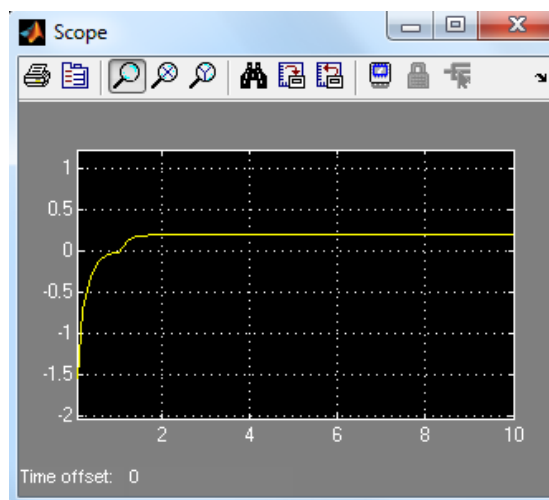
Potrebno je podesiti početno stanje integratora koje je zadato matematičkom jednačinom i iznosi -2. Novo početno stanje integratora unosi se tako što se duplim klikom na sam integrator otvori blok za promenu funkcije (eng. *Function block parameters*). U polje početno stanje (eng. *Initial condition*) unosimo zadato početno stanje. Istim postupkom menjamo pojačanje (eng. *Gain*) koje u ovom zadatku iznosi -5.



*Slika 1.7. Blok šema kola i podešavanje integratora*

Nakon podešavanja svih potrebnih parametara, otvaramo prozor osciloskopa duplim klikom na blok koji isti predstavlja. Simulaciju pokrećemo tasterom *Start*.

Sledeći korak jeste da izvršimo analizu grafika koji dobijamo na izlazu osciloskopa. i pokušamo da izvučemo što više korisnih informacija iz njega. Možemo da primetimo da je odziv sistema podeljen u dve oblasti. Od nule do dve sekunde sistem je nelinearan i nestabilan, što se naziva prolazno stanje. Stanje sistema nakon dve sekunde postaje linearno i stabilno i po tome dobija naziv stabilno stanje sistema.




*Slika 1.8. Rezultat simulacije*

## Vežba 2. Odmeravanje signala

### Teorijski deo

1. Ilustrovati proces odmeravanja signala u vremenskom i frekvencijskom domenu.

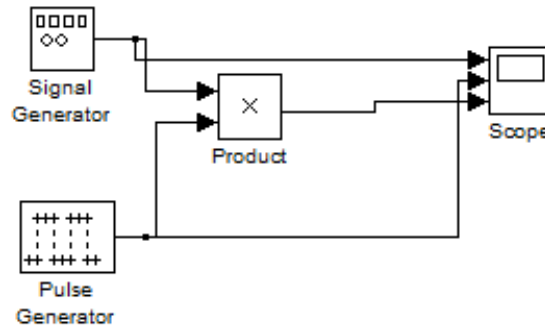


2. Obrazložiti teoremu odmeravanja



## Практични део

U Simulink-u programskog paketa MatLab kreirati model kao na slici 2.1.



*Slika 2.1. Model kojim se vrši odmeravanje signala*

Prilog uz sliku 2.1:

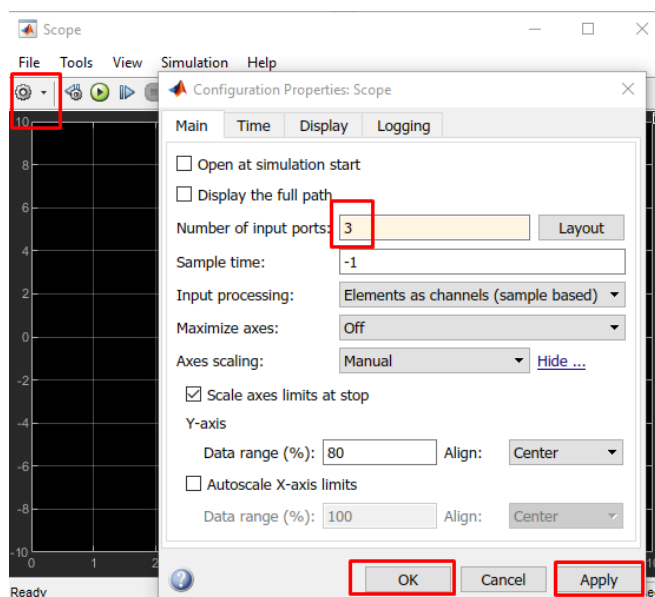
Simulink / Sources / **Signal Generator**

Simulink / Sources / **Pulse Generator**

Simulink / Commonly Used Blocks / **Product**

Simulink/ Sinks/ **Scope**

Potrebno je podesiti da blok *Scope* ima tri ulaza. Ovo podešavanje, nakon duplog klika na blok *Scope*, ilustrovano je na slici 2.2.



*Slika 2.2. Podešavanje bloka Scope*

**Primer 1:** Odmeravanje sinusnog signala  $s(t) = A \sin(2\pi ft)$ ,  $A = 3$ ,  $f = 1\text{Hz}$  periodom odmeravanja  $T_0 = 0.1\text{s}$ .

Пodešavanje bloka Signal Generator:

Waveform: sine

Amplitude: 3

Frequency: 1

Пodešavanje bloka Pulse Generator:

Pulse type: Sample based

Amplitude: 1

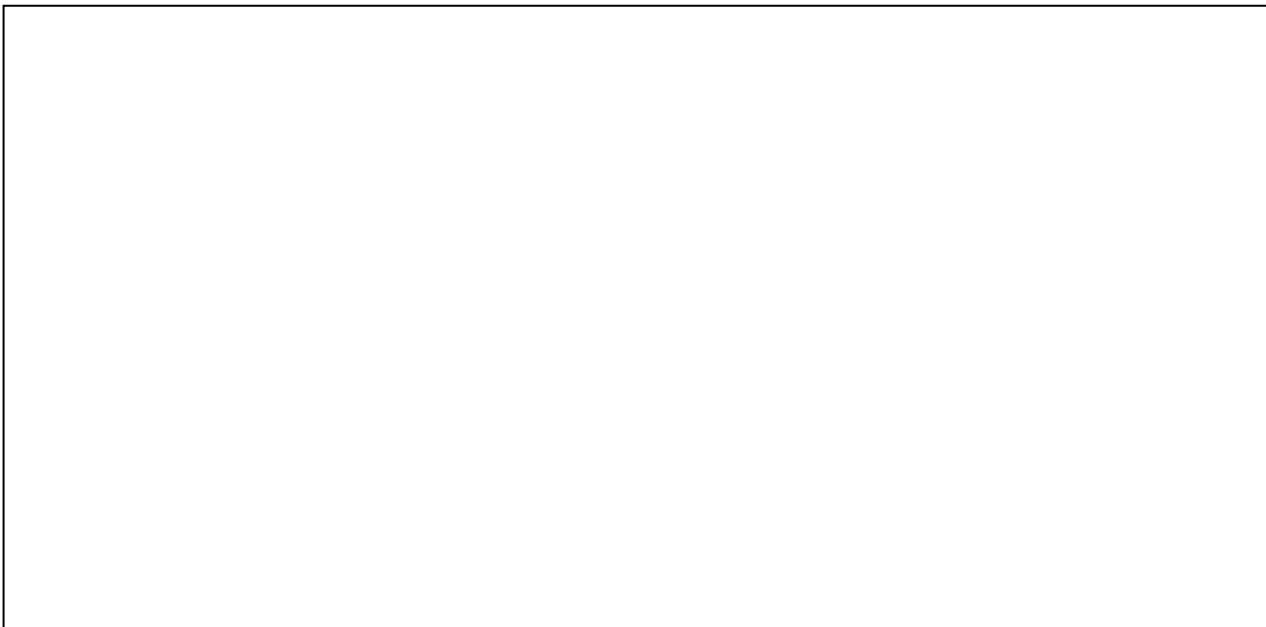
Sample time: 0.01

Period: 10                      Sample time \* Period = 0.1

Pulse width: 1

Vreme izvršenja simulacije je 1s (Simulation stop time = 1)

Predstaviti rezultate dobijene simulacijom.



**Primer 2:** Odmeravanje četvrtastog signala  $s(t) = \begin{cases} -4, & t \in \left[ kT, \frac{T}{2} + kT \right], k \in Z \\ 4, & \text{inače} \end{cases}$

periodom odmeravanja  $T_0 = 0.1s$ .

Podешavanje bloka Signal Generator:

Waveform: square

Amplitude: 4

Frequency: 2

Podешavanje bloka Pulse Generator:

Pulse type: Sample based

Amplitude: 1

Sample time: 0.01

Period: 10                      Sample time \* Period = 0.1

Pulse width: 1

Vreme izvršenja simulacije je 2s (Simulation stop time = 2)

Predstaviti rezultate dobijene simulacijom.



**Primer 3:** Odmeravanje testerastog signala  $s(t) = 10(0.5 - t)$ ,  $t \in [0, T]$  periodom odmeravanja  $T_0 = 0.1s$ .

Пodešavanje bloka Signal Generator:

Waveform: sawtooth

Amplitude: 5

Frequency: 1

Пodešavanje bloka Pulse Generator:

Pulse type: Sample based

Amplitude: 1

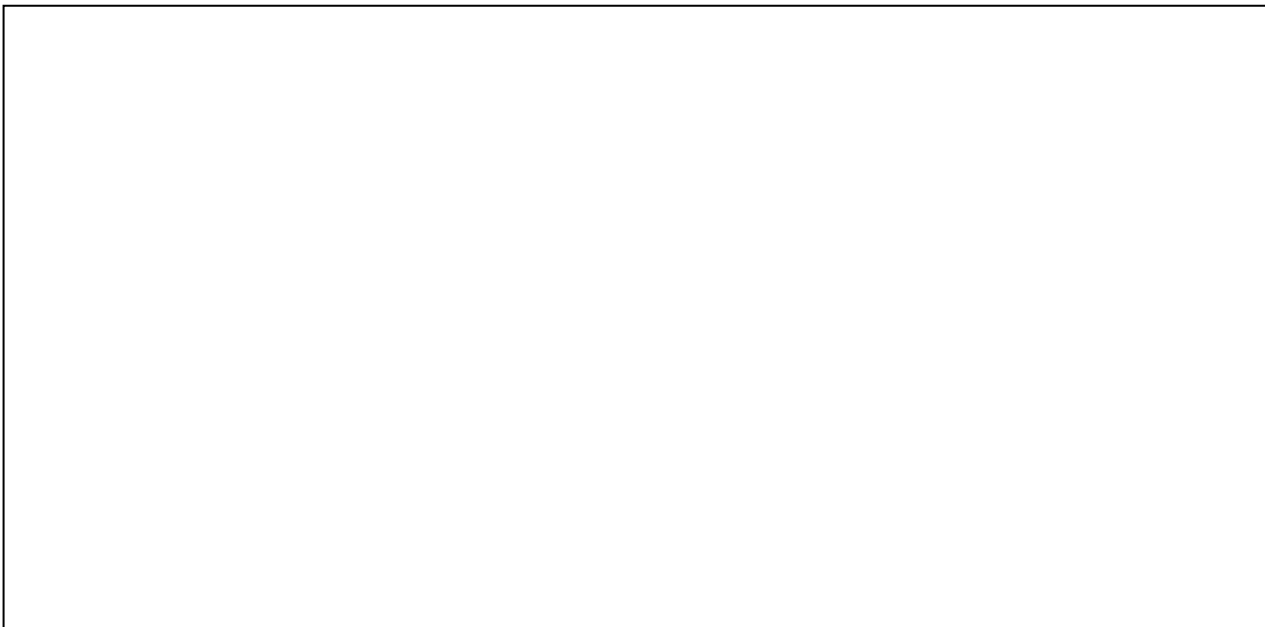
Sample time: 0.01

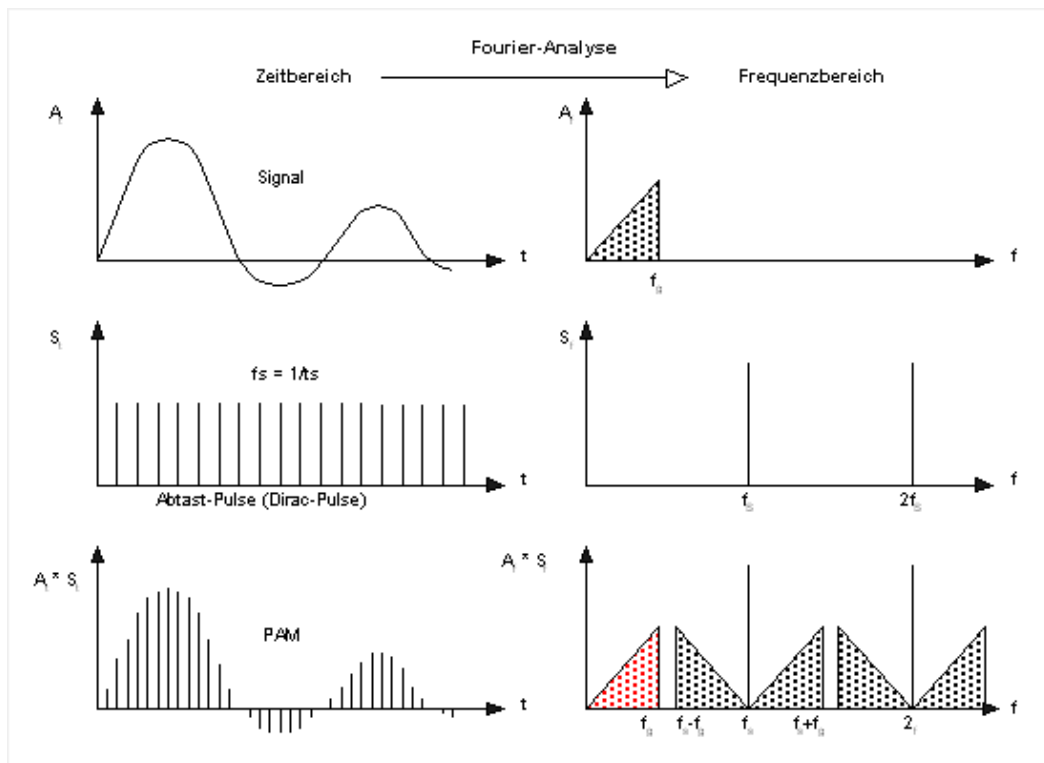
Period: 10                      Sample time \* Period = 0.1

Pulse width: 1

Vreme izvršenja simulacije je 2s (Simulation stop time = 2)

Predstaviti rezultate dobijene simulacijom.





**Prilog 1:** Odmeravanje signala u vremenskom i frekvencijskom domenu

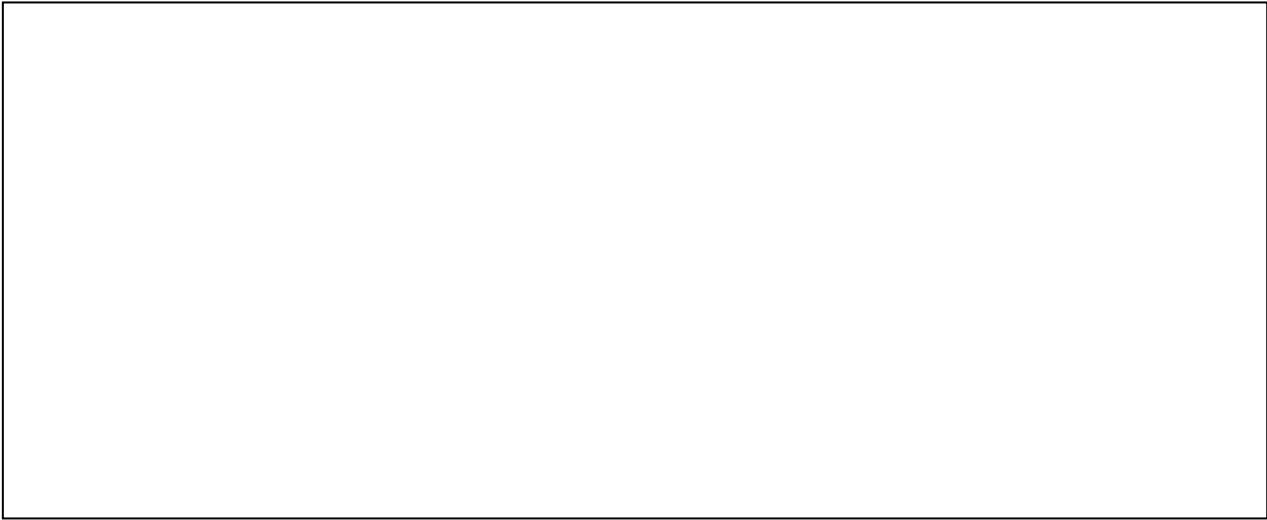
**Zadatak za samostalni rad**

Koristeći instrukcije sa sajta *Signal Sampling: Nyquist-Shannon Theorem* čija je web adresa [http://195.134.76.37/applets/AppletNyquist/App1\\_Nyquist2.html](http://195.134.76.37/applets/AppletNyquist/App1_Nyquist2.html) nacrtati sledeće sinusne signale i izvršiti njihovo odmeravanje.

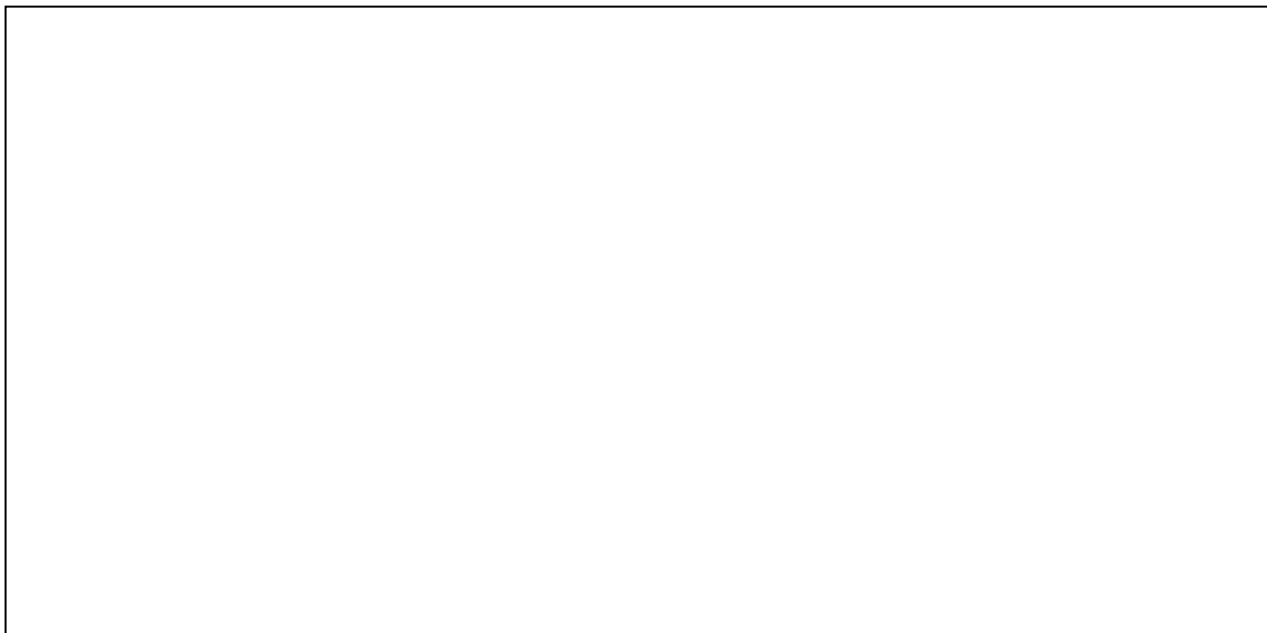
1. Frekvencija signala je 1 Hz, a frekvencija odmeravanja 2 Hz.



2. Frekvencija signala je 1 Hz, a frekvencija odmeravanja 4 Hz.



3. Frekvencija signala je 1 Hz, a frekvencija odmeravanja 10 Hz.



4. Na osnovu dobijenih grafika, dati komentar o samplovanju u zavisnosti od odnosa frekvencija signala i odmeravanja, kao i komentar o pojavi *aliasing* efekta.

5. Definisati pojam aliasing (*aliasing*).

## Vežba 3. Uniformna kvantizacija

### Teorijski deo

1. Definisati proces kvantovanja.

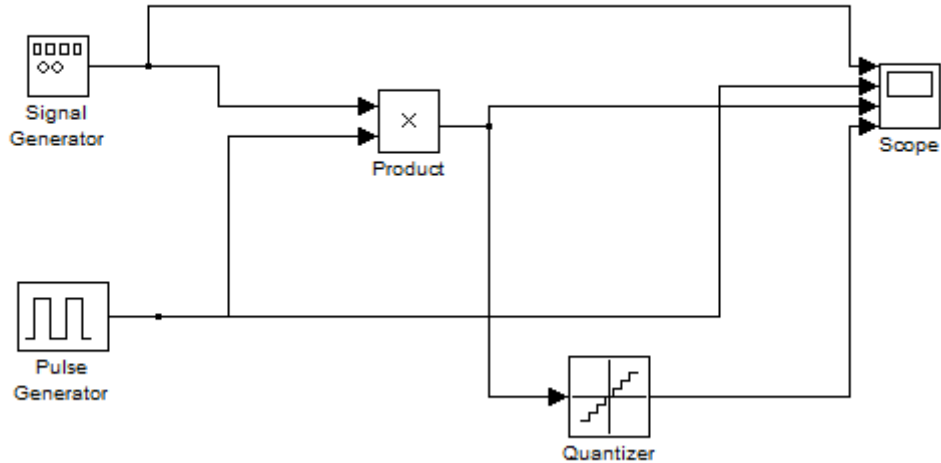
2. Zašto je kvantovanje neophodno u procesu digitalizacije?

3. Podela kvantizera prema veličini koraka kvantovanja.

4. Nacrtati prenosnu karakteristiku kvantizera ako je  $y = q * \text{round}(u/q)$ ,  
gde je  $u$  je ulazni signal,  $q$  kvantizacioni interval, a  $y$  izlazni signal iz kvantizera za slučaj  
da je  $u=3$ , a  $q=0.5$ .

## Практични део

U Simulink-u programskog paketa MatLab kreirati model kao na Slici 3.1.



*Slika 3.1. Model kojim se vrši kvantovanje signala*

Prilog uz sliku 3.1:

Simulink / Sources / **Signal Generator**

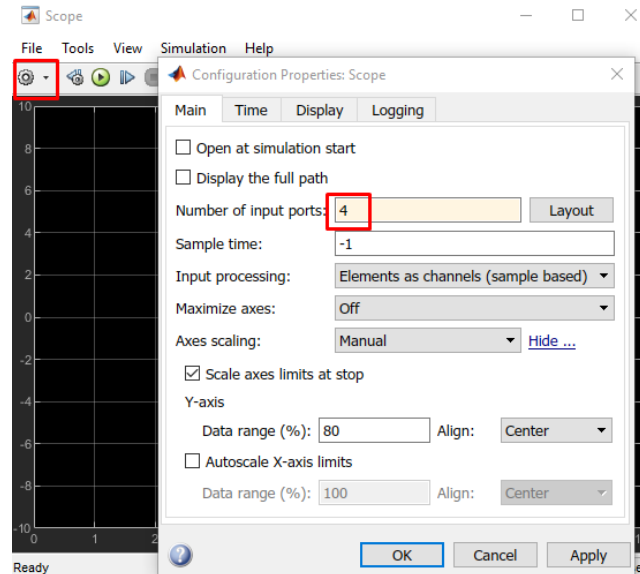
Simulink / Sources / **Pulse Generator**

Simulink / Commonly Used Blocksets / **Product**

Simulink/ Sinks/ **Scope**

Signal Processing Blockset / Quantizers / **Quantizer**

Potrebno je podesiti da blok Scope ima četiri ulaza. Ovo podešavanje, nakon duplog klika na blok Scope, ilustrovano je na slici 3.2.



*Slika 3.2. Podešavanje bloka Scope*

**Primer 1:** Kvantovanje odmeraka sinusnog signala  $s(t) = A \sin(2\pi ft)$ ,  $A = 3.8$ ,  $f = 1\text{Hz}$ , ako je perioda odmeravanja  $T_0 = 0.1\text{s}$ .

Podešavanje bloka Signal Generator:

Waveform: sine

Amplitude: 3.8

Frequency: 1

Podešavanje bloka Pulse Generator:

Pulse type: Sample based

Amplitude: 1

Sample time: 0.01

Period: 10

Sample time \* Period = 0.1

Pulse width: 1

Vreme izvršenja simulacije je 1s (Simulation stop time = 1)

Predstaviti rezultate dobijene simulacijom.



**Primer 2:** Kvantovanje odmeraka četvrtastog signala  $s(t) = \begin{cases} -4, & t \in \left[ kT, \frac{T}{2} + kT \right], k \in Z \\ 4, & \text{inače} \end{cases}$

ako je perioda odmeravanja  $T_0 = 0.1s$ .

Podешavanje bloka Signal Generator:

Waveform: square

Amplitude: 4

Frequency: 2

Podешavanje bloka Pulse Generator:

Pulse type: Sample based

Amplitude: 1

Sample time: 0.01

Period: 10

Sample time \* Period = 0.1

Pulse width: 1

Vreme izvršenja simulacije je 2s (Simulation stop time = 2)

Predstaviti rezultate dobijene simulacijom.



**Primer 3:** Kvantovanje odmeraka testerastog signala  $s(t) = 10(0.5 - t)$ ,  $t \in [0, T]$ , ako je perioda odmeravanja  $T_0 = 0.1s$ .

Podешavanje bloka Signal Generator:

Waveform: sawtooth

Amplitude: 5

Frequency: 1

Podешavanje bloka Pulse Generator:

Pulse type: Sample based

Amplitude: 1

Sample time: 0.01

Period: 10

Sample time \* Period = 0.1

Pulse width: 1

Vreme izvršenja simulacije je 2s (Simulation stop time = 2)

Predstaviti rezultate dobijene simulacijom.



### Zadatak za samostalni rad

Prenos digitalnih signala u osnovnom opsegu učestanosti definise se kao direktan prenos, tj. prenos koji ne podrazumeva konverziju spektra. U osnovi ovakvog postupka prenosa je tehnika oblikovanja signala koji bi trebalo preneti tako da se on učini pogodnim za prenos. U pokušajima da se pronadju takvi oblici signala koji imaju određene osobine, kao sto su, recimo, visoka spektralna i energetska efikasnost ili osobina samosinhronizacije, predložen je veliki broj različitih oblika signala. Najčesće se ovi signali nazivaju linijski kodovi. Linijski kodovi uglavnom su razvijani 60-ih godina za potrebe digitalnog prenosa kroz telefonske kanale ili snimanja digitalnih signala na magnetne medijume. Novija istraživanja uglavnom su usmerena na razvoj linijskih kodove za prenos signala po optickim vlaknima.

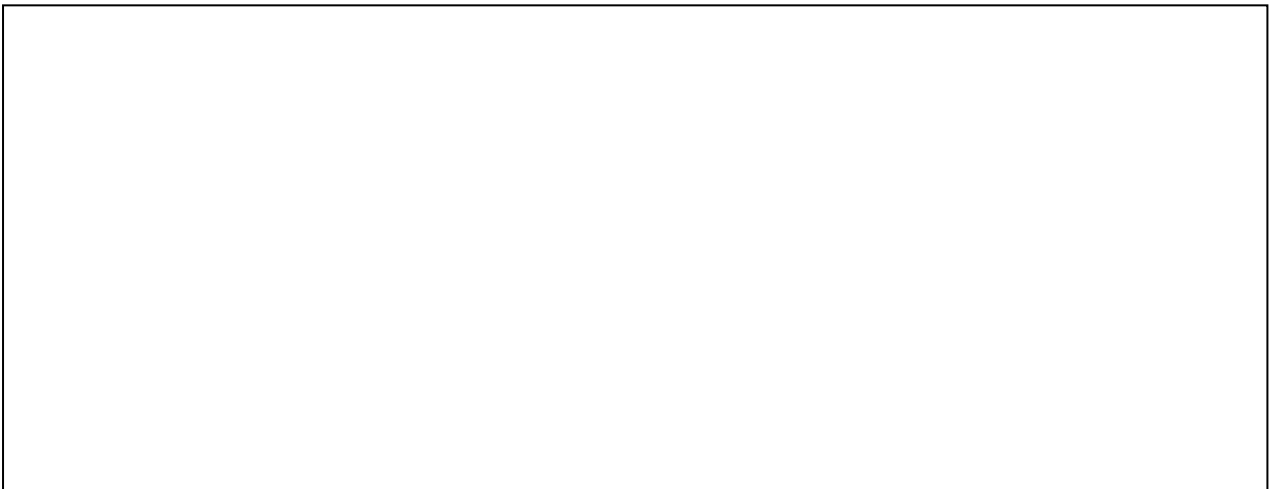
Cilj ove vežbe je upoznavanje studenata sa talasnim oblicima digitalnog signala.

**Zadatak 1.** Koristeći simulator linijskog koda dostupan na Internetu (web adresa: <https://sourceforge.net/projects/linecodingsimulator/files/>), za sekvencu 1 0 0 1 1 1 0 1 0 0 1 nacrtati talasne oblike za slučajeve:

- a) unipolar NRZ,
- b) polar RZ,
- c) bipolar NRZ-L,
- d) bipolar NRZ-I,
- e) bipolar AMI,
- f) Manchester



**Zadatak 2.** Opisati Manchester i bipolar NRZ-I kodovanje.



## Vežba 4. Simulacija generatora pseudoslučajnog niza

### Теоријски део

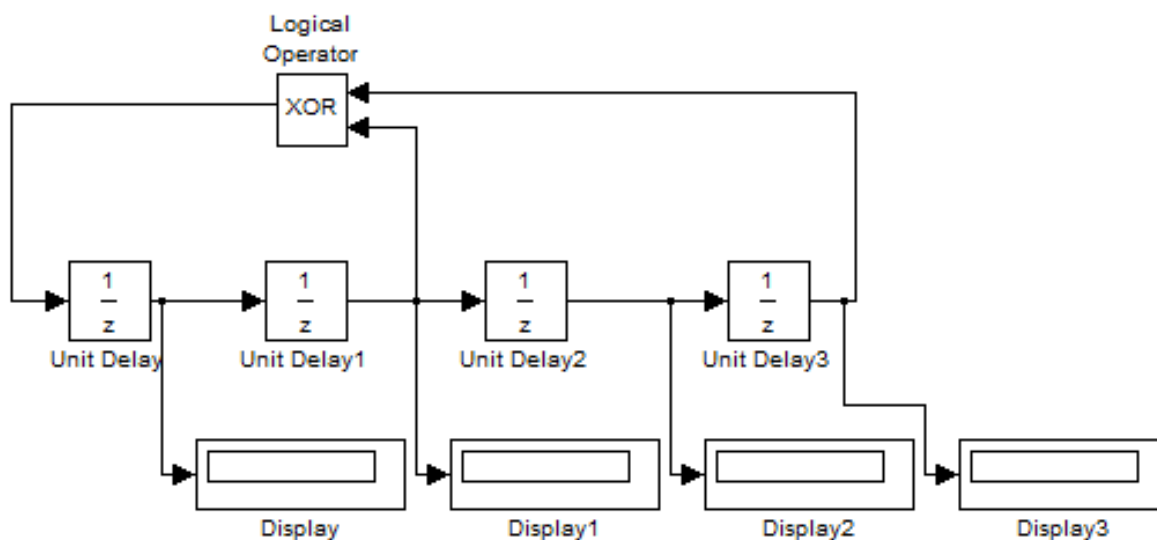
1. Definисати pseudoslučajni niz.

2. Представити блок шему генератора pseudoslučajnog niza реда  $N$  и одредити максималну периоду niza.

3. Од чега зависи периода генерисаног pseudoslučajnog niza?

## Практични део

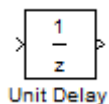
U Simulink-u programskog paketa MatLab kreirati model kao na Slici 1.



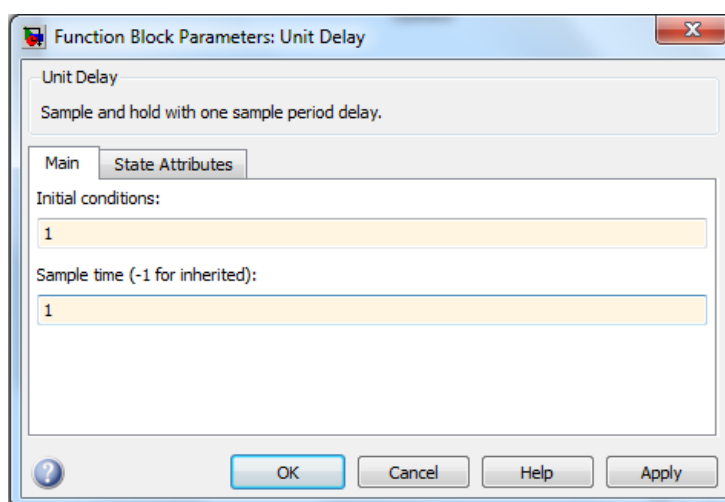
*Slika 4.1. Model generatora pseudoslučajnog niza*

**Prilog uz sliku 4.1:**

### 1. Simulink / Commonly used Blocks / Unit delay

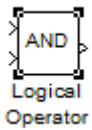


Nakon duplog klika na blok, izvršiti podešavanje kao na slici 4.2.

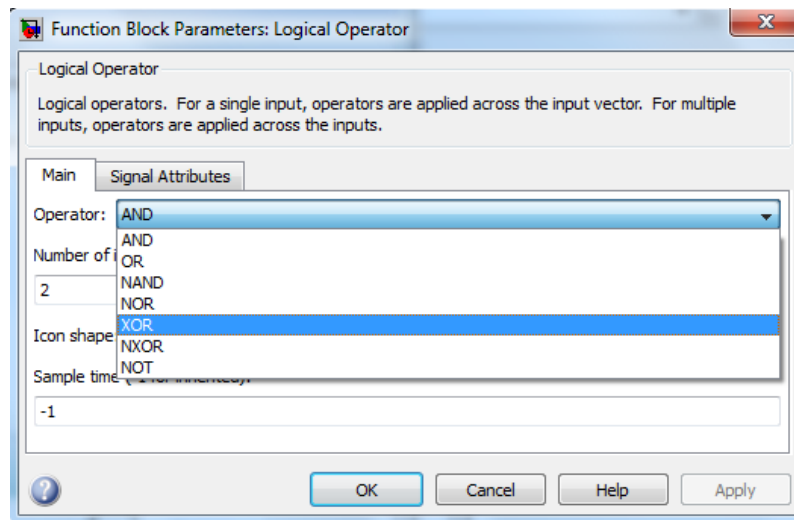


*Slika 4.2. Podešavanje bloka za kašnjenje*

## 2. Simulink / Commonly used Blocks / **Logical operator**

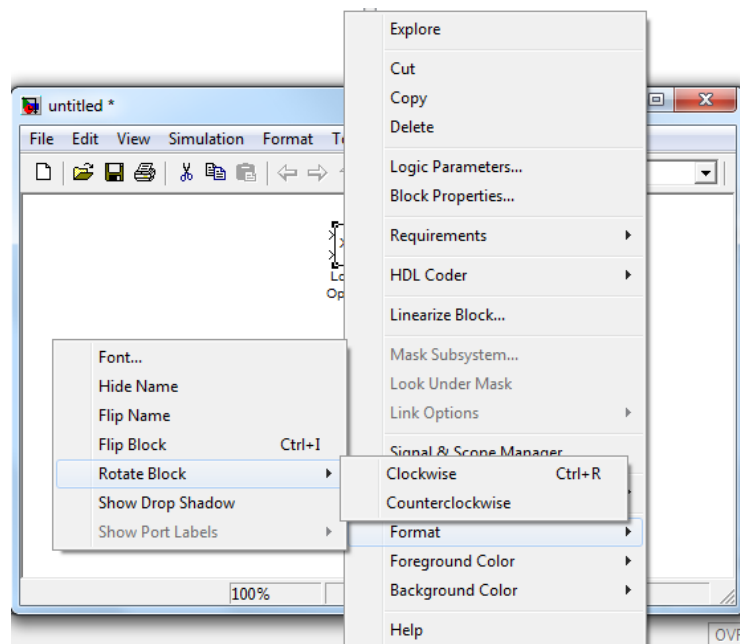


Nakon duplog klika na blok, izabrati XOR logičku funkciju (slika 4.3).



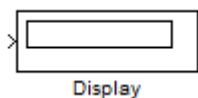
*Slika 4.3. Podešavanje logičkog operatora*

Zatim izvršiti rotaciju bloka XOR (slika 4.4).

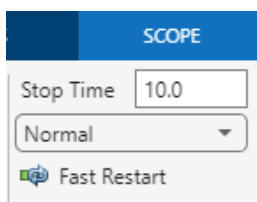


*Slika 4.4. Rotacija bloka*

### 3. Simulink / Sinks / **Display**



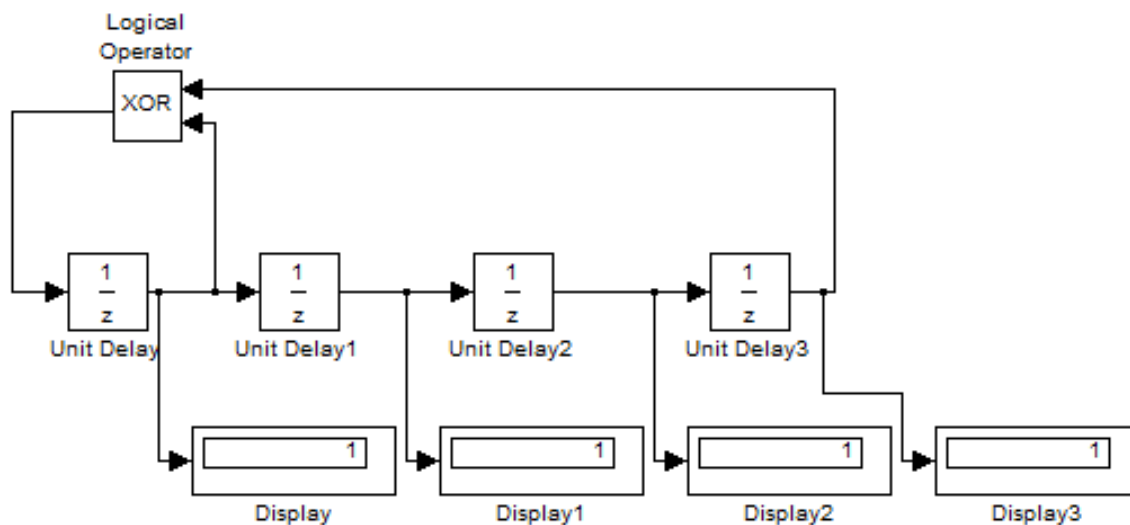
**Zadatak 1.** Menjati *Simulation stop time* počevši od 0 sa korakom 1 (Slika 4.5) i odrediti periodu niza.



*Slika 4.5. Vreme simulacije*

### **Zadatak 2.**

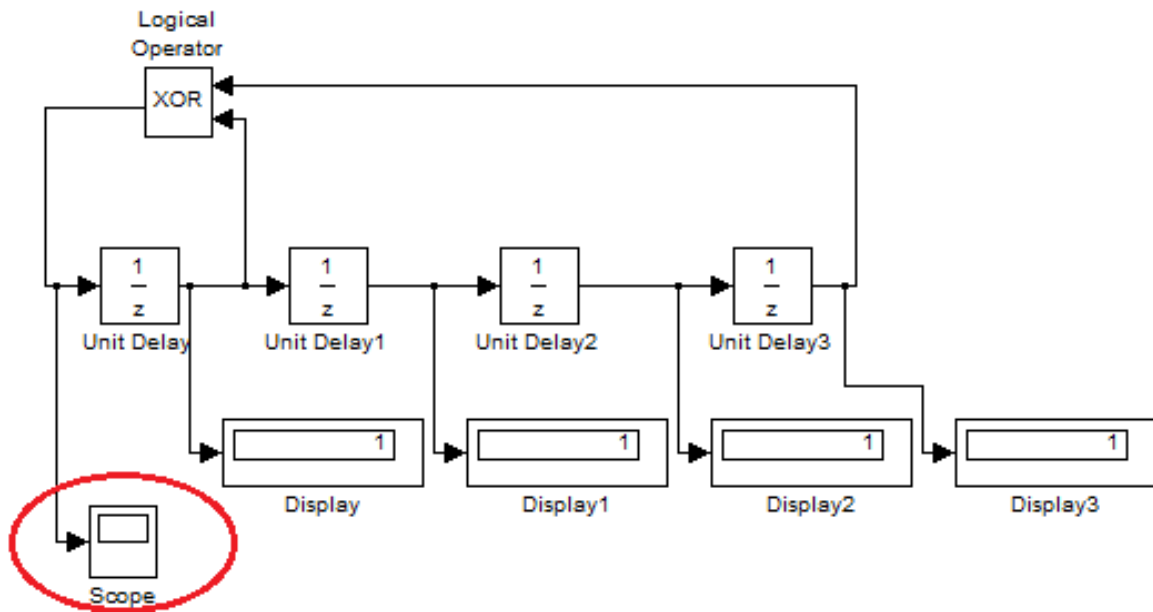
a) Preurediti model sa slike 4.1 u model sa slike 4.6.



*Slika 4.6. Model generatora pseudoslučajnog niza*

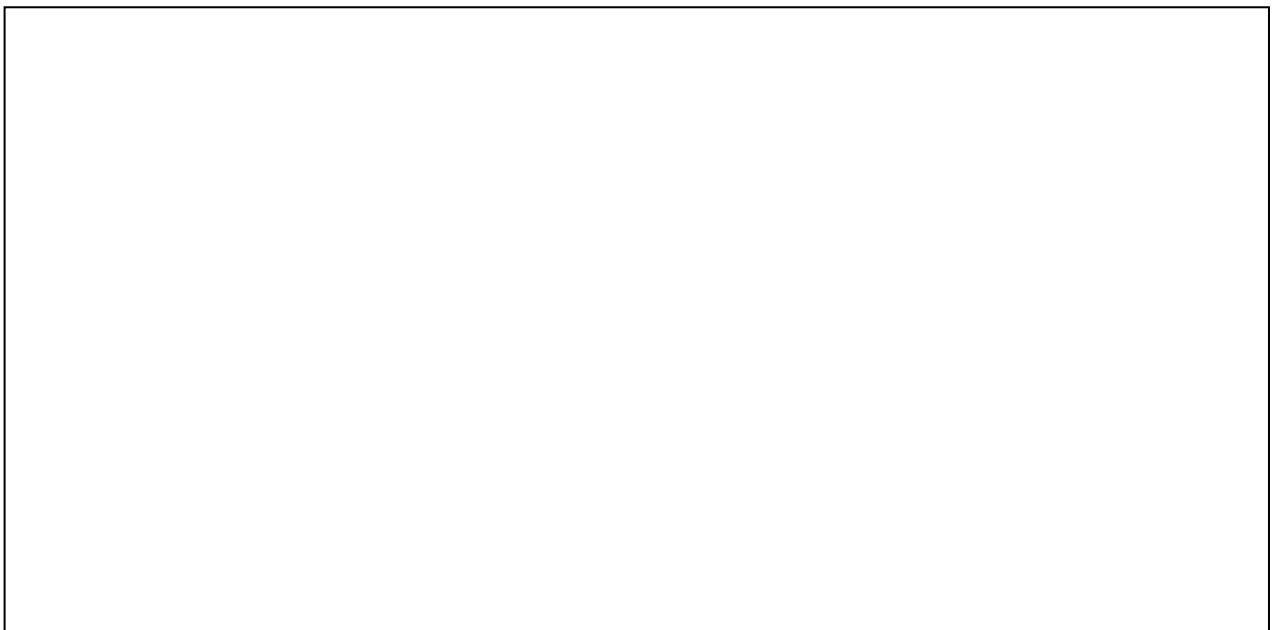
b) Menjati *Simulation stop time* počevši od 0 sa korakom 1 (slika 4.5) i odrediti periodu niza.

c) Ubaciti blok Scope (Simulink / Sinks / **Scope**), podesiti vreme simulacije 15s i pokrenuti simulaciju (slika 4.7). Nacrtati signal koji se dobija na bloku Scope koji se koristi u procesu skremblovanja.



*Slika 4.7. Posmatranje signala koji se koristi u procesu skremblovanja*

Predstaviti rezultate dobijene simulacijom.



## Vežba 5. Linearni linijski kodovi

### Teorijski deo

1. Navesti osnovni razlog zbog koga se vrši linijsko kodovanje.

2. Predstaviti blok šemu koda linearnog linijskog koda reda  $N$  i napisati izraze za prenosnu funkciju i jednačinu koda.

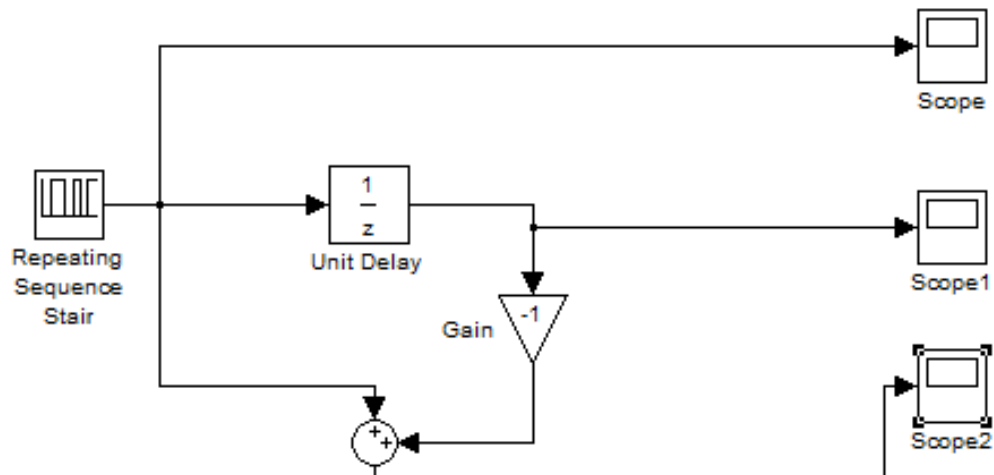
3. Predstaviti blok šemu dekodera linearnog linijskog koda reda  $N$  i napisati jednačinu dekodera.

## Практични део

**Zadatak:** Dat je koder sa prenosnom funkcijom  $H(D)=1-D$ . Realizovati šemu koder i dekodera. Ako je informacioni sadržaj na ulazu koder a  $\{b_n\} = \{\dots 0110101\dots\}$  odrediti informacioni sadržaj na izlazu koder a i dekodera ako prilikom prenosa nije došlo do greške.

### Simulacija koder a

U Simulink-u programskog paketa MatLab kreirati model kao na slici 5.1.



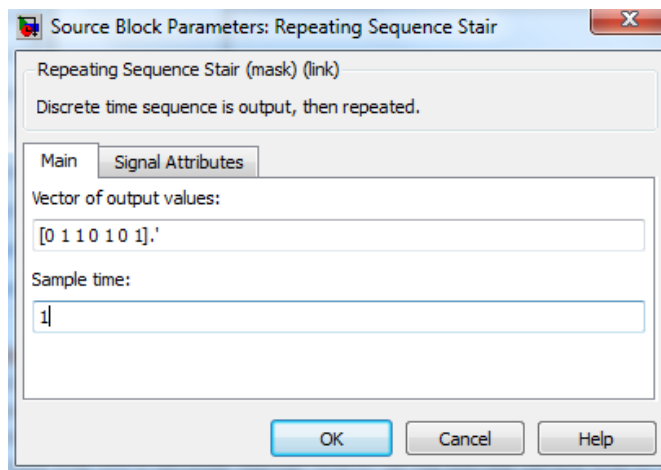
*Slika 5.1. Model koder a linearnog linijskog koda*

### Prilog uz sliku 5.1:

#### 1. Simulink / Sources / Repeating Sequence Stair

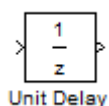


Nakon duplog klika na blok, izvršiti podešavanje kao na slici 5.2.



*Slika 5.2. Podešavanje bloka Repeating Sequence Stair*

## 2. Simulink / Commonly used Blocks / Unit delay



## 3. Simulink / Commonly used Blocks / Gain



Podesiti pojačanje -1. Za rotaciju bloka videti sliku 5.3.

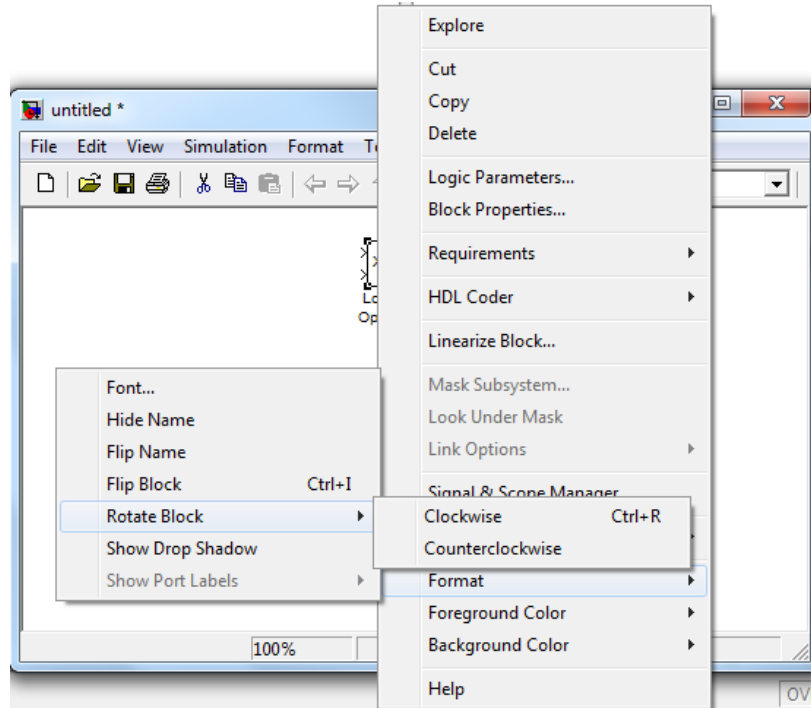
## 4. Simulink / Commonly used Blocks / Sum



Za rotaciju bloka videti sliku 5.3.

## 5. Simulink / Sinks / Scope





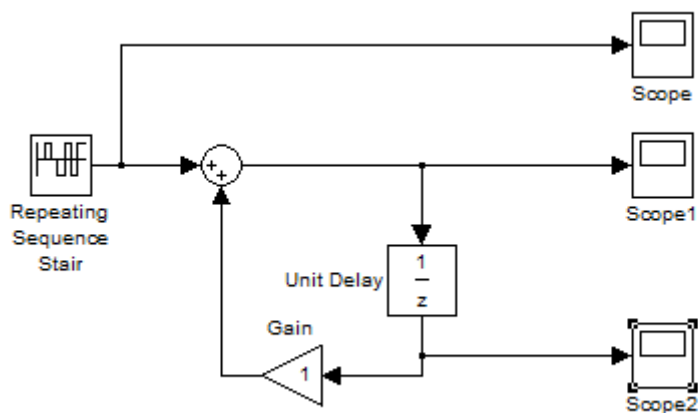
*Slika 5.3. Rotacija bloka*

Podesiti vreme simulacije 7 sekundi i pokrenuti simulaciju. Nacrtati talasne oblike na blokovima *Scope*, a zatim popuniti sledeću tabelu

$b_n$							
$b_{n-1}$							
$a_n$							

### Simulacija dekodera

U Simulink-u programskog paketa MatLab kreirati model kao na Slici 4.



*Slika 5.4. Model dekodera linearnog linijskog koda*

Vector of output values bloka *Repeating Sequence Stair* je sekvenca  $a_n$ , a *Sample time* je 1.

Podesiti vreme simulacije 7 sekundi. Nacrtati talasne oblike na blokovima *Scope*, a zatim popuniti sledeću tabelu.

$a_n$							
$b_n$							
$b_{n-1}$							



Uporediti vrednosti sekvenci  $b_n$  u prvoj i drugoj tabeli i dati komentar.



## Vežba 6. Amplitudska, frekvencijska i fazna modulacija

### Teorijski deo

#### 6.1. Amplitudska modulacija

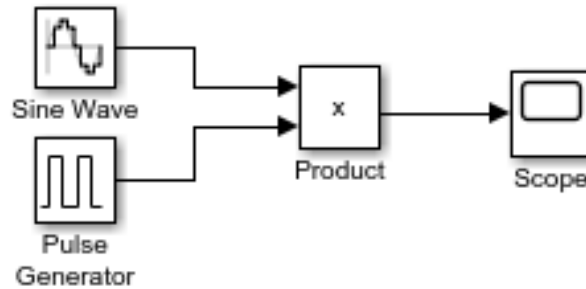
1. Šta je digitalna amplitudska modulacija (ASK)?

2. Koje su prednosti ASK modulacije?

3. Grafički prikazati ASK modulaciju.

## Практични део

U Simulink-u programskog paketa MatLab kreirati model kao na slici 6.1.



*Slika 6.1. Model*

Postupak konfiguracije bloka **Sine Wave** je sledeći:

Sine type: Time based

Time (t): Use simulation time

Amplitude: 1

Bias: 0

Frequency (rad/sec): 100

Phase (rad): 0

Sample time: 0.0001

Postupak konfiguracije bloka **Pulse Generator** je sledeći:

Pulse Type: Time based

Time (t): Use simulation time

Amplitude: 1

Period (secs): 2

Pulse Width (% of period): 50

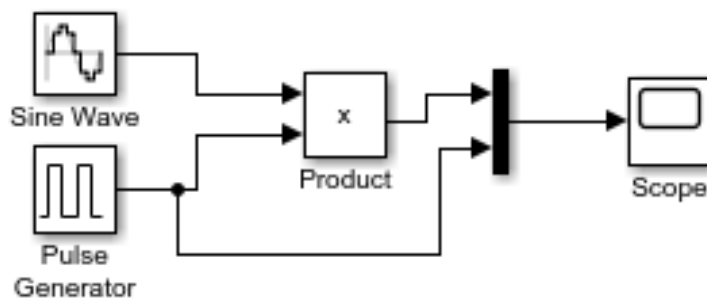
Phase delay (secs): 0

Vreme izvršenja simulacije je 4 sekundi (*Simulation stop time = 4*).

Predstaviti rezultate dobijene simulacijom.

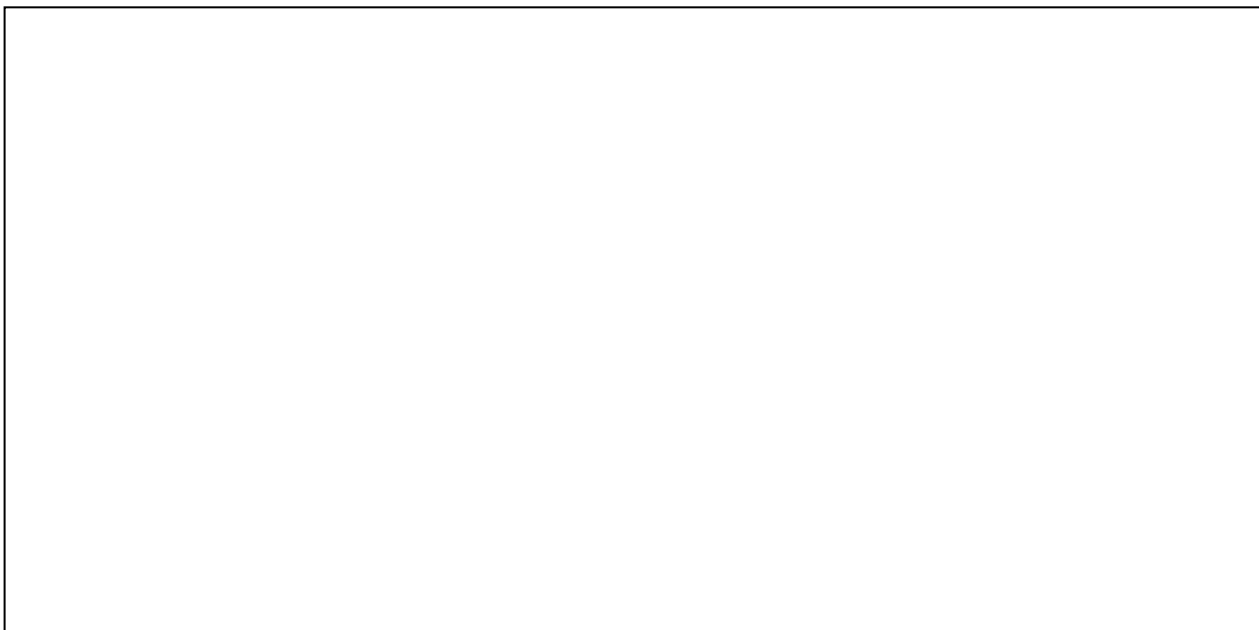


Da bi se na osciloskopu prikazao i modulišuci signal potrebno je dodati multiplekser (blok MUX), pri čemu će jedan ulaz biti povezan na izlaz iz množača (*Product*) a drugi ulaz na izlaz puls generatora (*Pulse Generator*) kao što je prikazano na slici 6.2.



*Slika 6.2. Model koji prikazuje i modulišuci signal*

Predstaviti rezultate dobijene simulacijom.



## Теоријски део

### 6.2. Frekvencijska modulacija

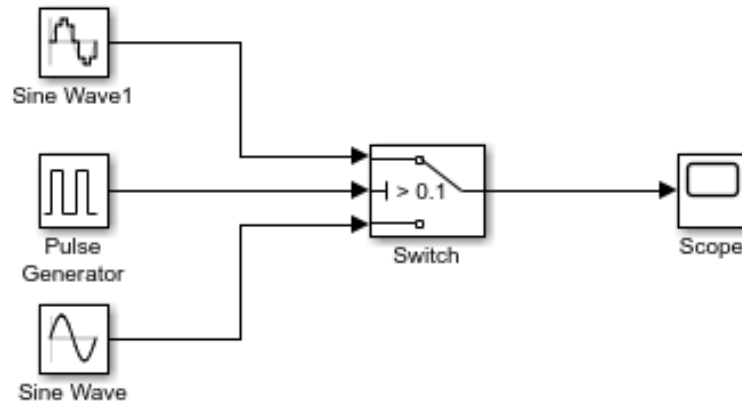
1. Šta je digitalna frekvencijska modulacija (FSK)?

2. Koje su prednosti FSK modulacije?

3. Grafički prikazati FSK modulaciju.

## Практични део

U Simulink-u programskog paketa MatLab kreirati model kao na slici 6.3.



*Slika 6.3. Model*

Postupak konfiguracije bloka *Sine Wave1* je sledeći:

Sine type: Time based

Time (t): Use simulation time

Amplitude: 1

Bias: 0

Frequency (rad/sec): 200

Phase (rad): 0

Sample time: 0.001

Postupak konfiguracije bloka *Sine Wave* je sledeći:

Sine type: Time based

Time (t): Use simulation time

Amplitude: 1

Bias: 0

Frequency (rad/sec): 100

Phase (rad): 0

Sample time: 0.001

Postupak konfiguracije bloka *Pulse Generator* je sledeći:

Pulse Type: Time based

Time (t): Use simulation time

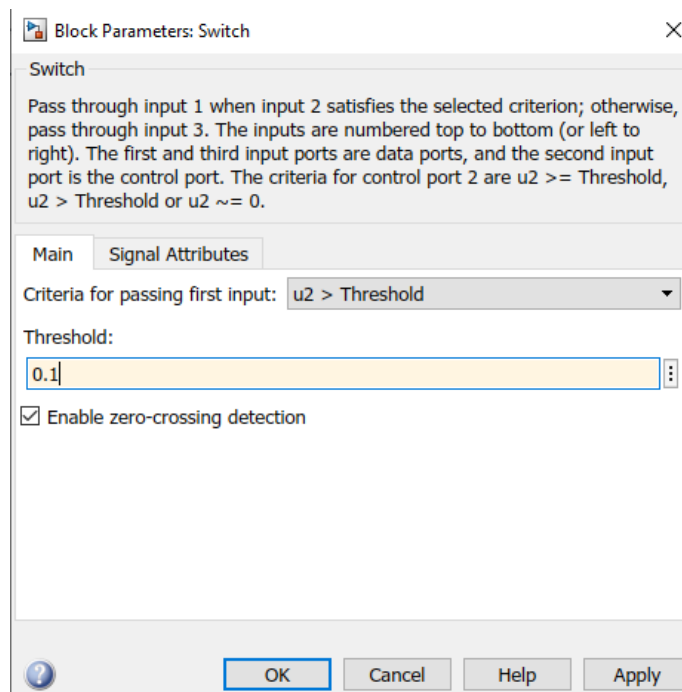
Amplitude: 1

Period (secs): 2

Pulse Width (% of period): 50

Phase delay (secs): 0

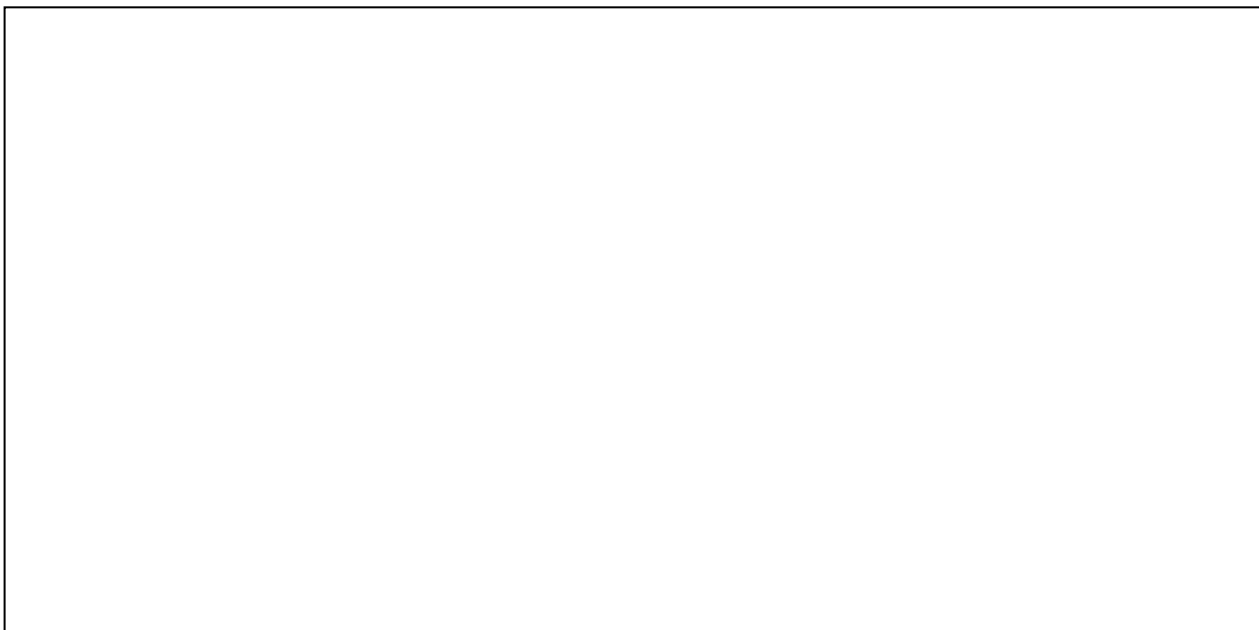
Konfiguracija *switch*-a je prikazana na slici 6.4:



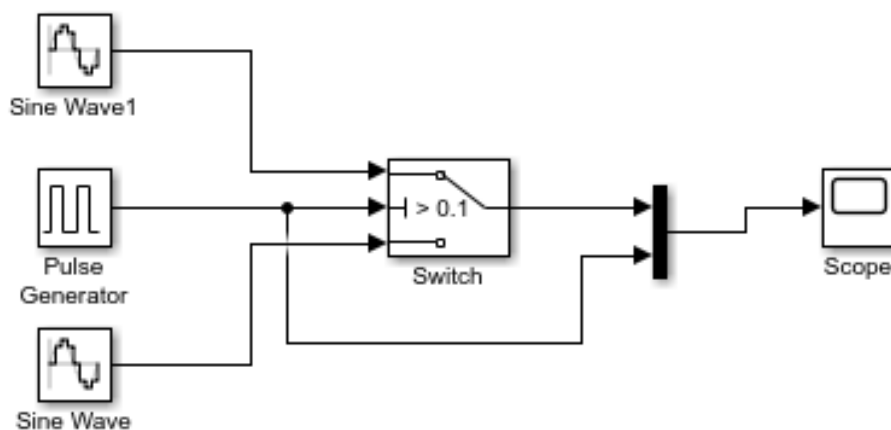
**Slika 6.4.** Konfiguracija *switch*-a

Vreme izvršenja simulacije je 4 sekundi (Simulation stop time = 4).

Predstaviti rezultate dobijene simulacijom.

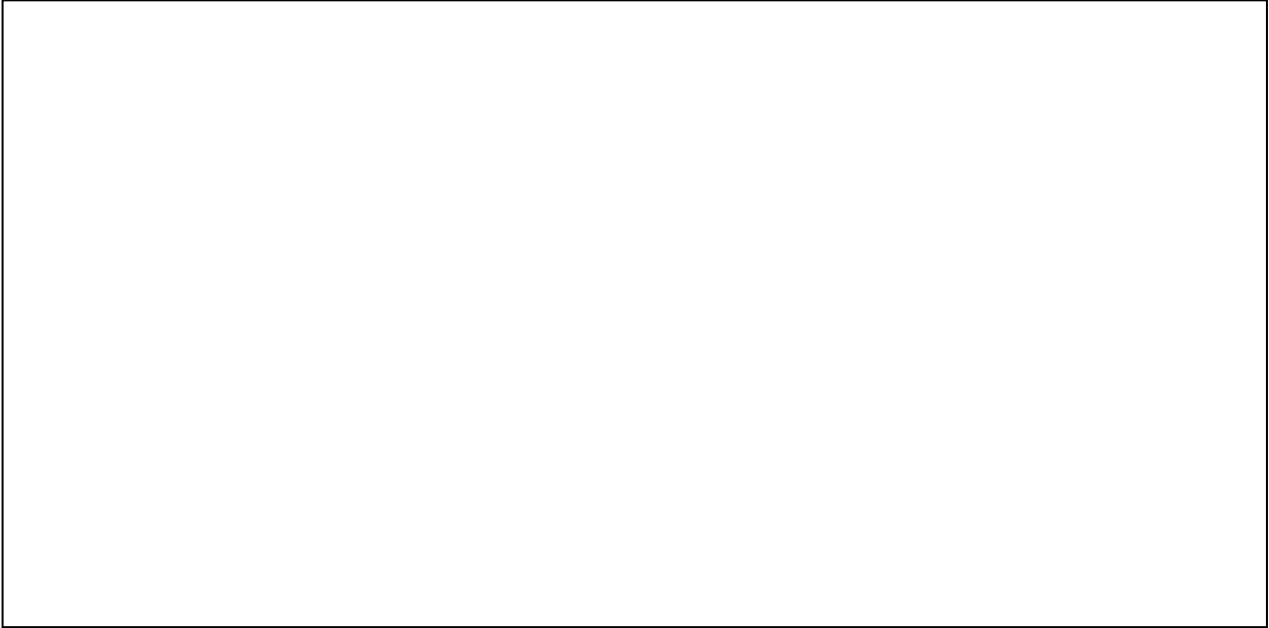


Da bi se na osciloskopu prikazao i modulišući signal potrebno je dodati multiplekser (blok MUX), pri čemu će jedan ulaz biti povezan na izlaz iz switch-a a drugi ulaz na izlaz puls generatora (*Pulse Generator*) kao što je prikazano na slici 6.5.



*Slika 6.5. Model koji prikazuje i modulišući signal*

Predstaviti rezultate dobijene simulacijom.



## Теоријски део

### 6.3. Fazna modulacija

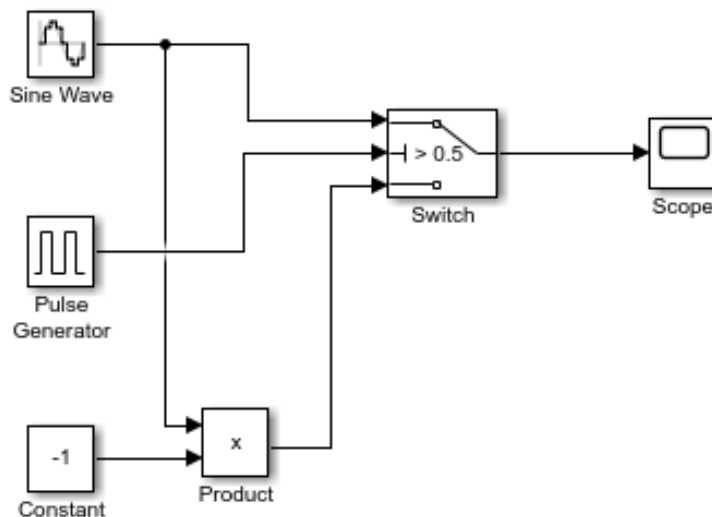
1. Šta je digitalna fazna modulacija (PSK)?

2. Koje su prednosti PSK modulacije?

3. Gde se primenjuje PSK modulacija?

## Практични део

U Simulink-u programskog paketa MatLab kreirati model kao na slici 6.6.



*Slika 6.6. Model*

Postupak konfiguracije bloka **Pulse Generator** je sledeći:

Pulse Type: Time based

Time (t): Use simulation time

Amplitude: 1

Period (secs): 2

Pulse Width (% of period): 50

Phase delay (secs): 0

Postupak konfiguracije bloka **Sine Wave** je sledeći:

Sine type: Time based

Time (t): Use simulation time

Amplitude: 1

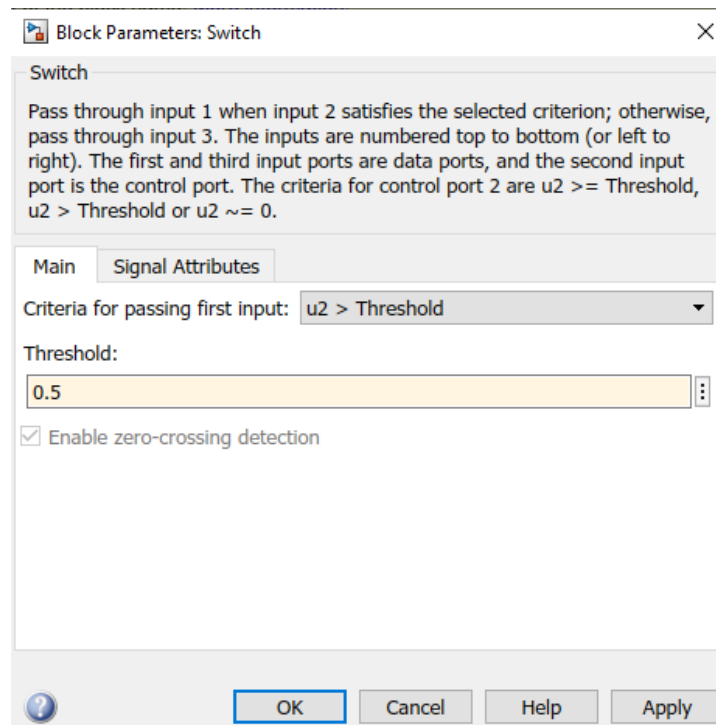
Bias: 0

Frequency (rad/sec):  $2 \cdot \pi \cdot 2$

Phase (rad): 0

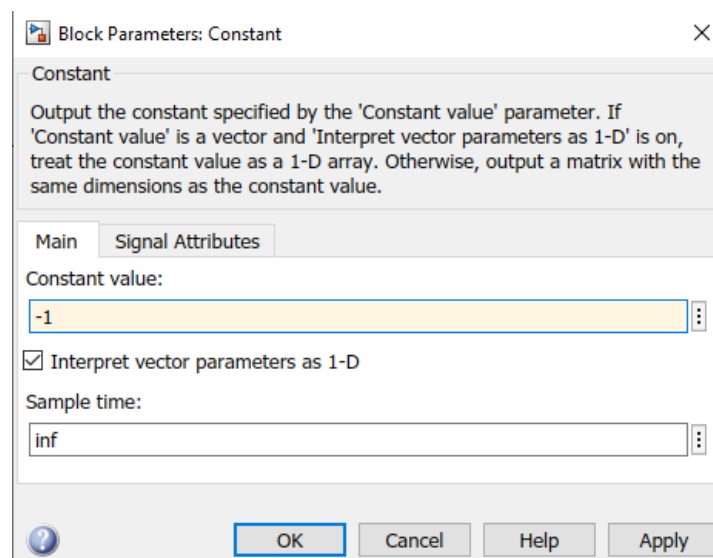
Sample time: 0.01

Konfiguracija *switch*-a je prikazana na slici 6.7:



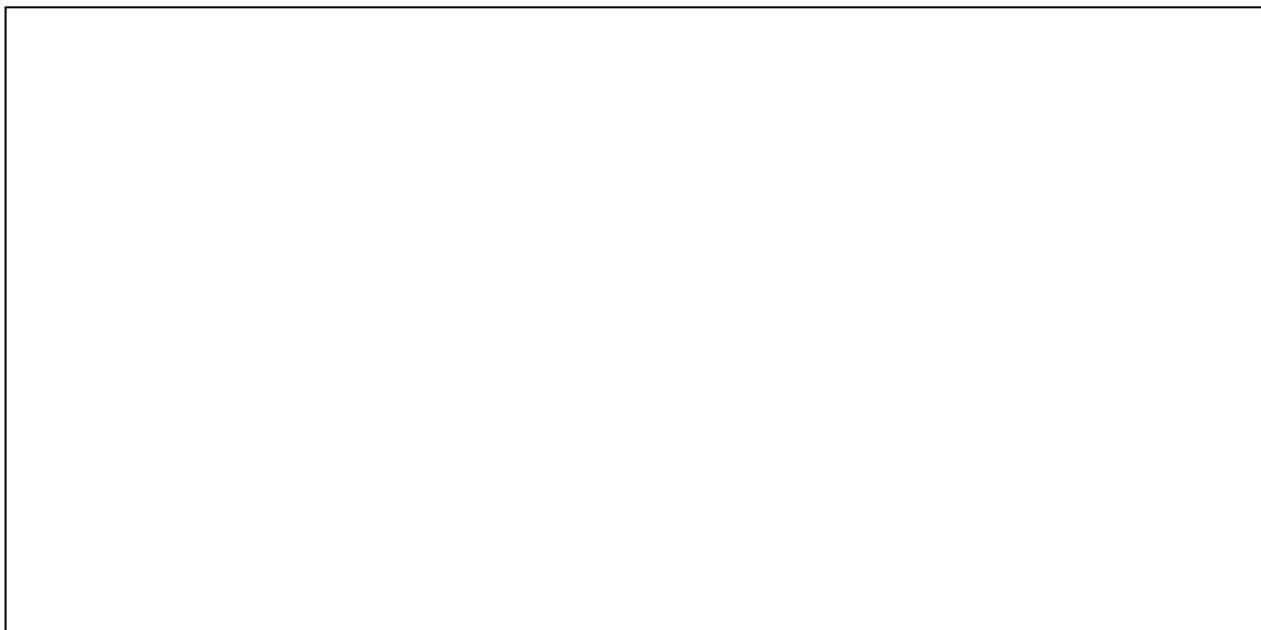
*Slika 6.7. Konfiguracija switch-a*

Konstantu podešavamo kao na slici 6.8:

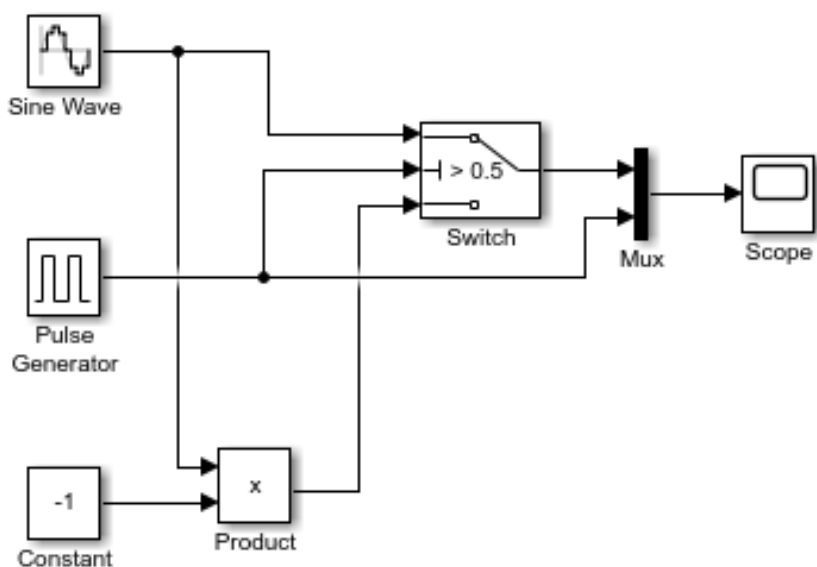


*Slika 6.8. Konfiguracija konstante*

Predstaviti rezultate dobijene simulacijom.

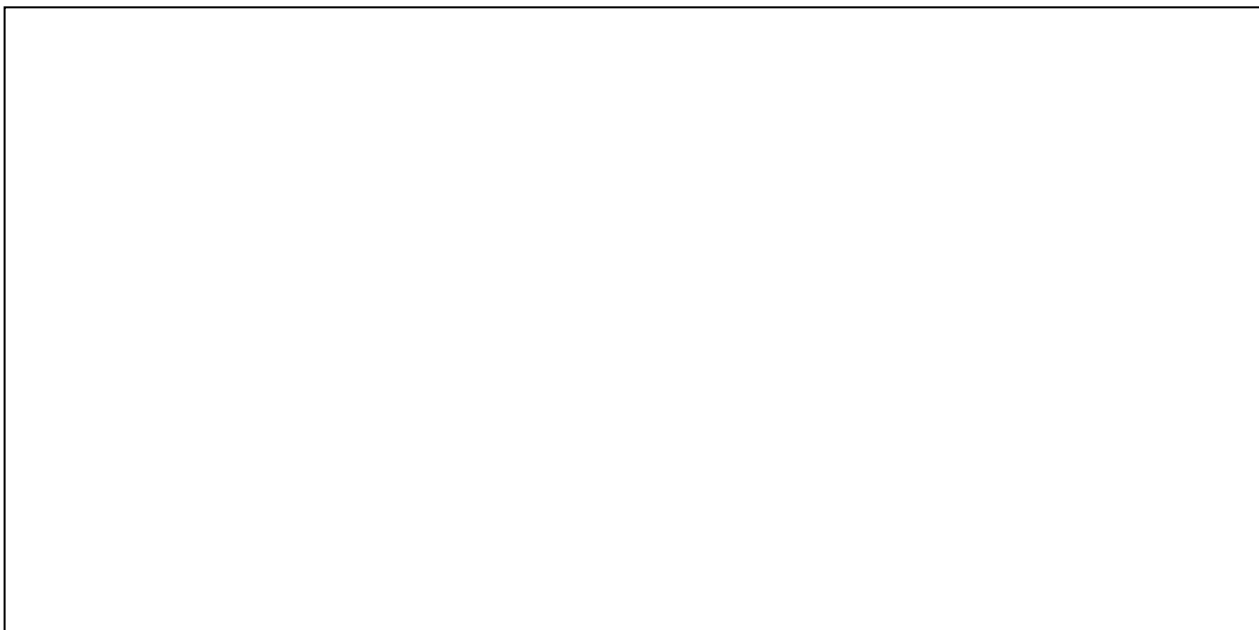


Da bi se na osciloskopu prikazao i modulišuci signal potrebno je dodati multiplekser (blok MUX), pri čemu će jedan ulaz biti povezan na izlaz iz switch-a a drugi ulaz na izlaz puls generatora (*Pulse Generator*) kao što je prikazano na slici 6.9.



*Slika 6.9. Model koji prikazuje i modulišuci signal*

Predstaviti rezultate dobijene simulacijom.



## Vežba 7. Digitalna binarna amplitudska, frekvencijska i fazna modulacija

### Teorijski deo

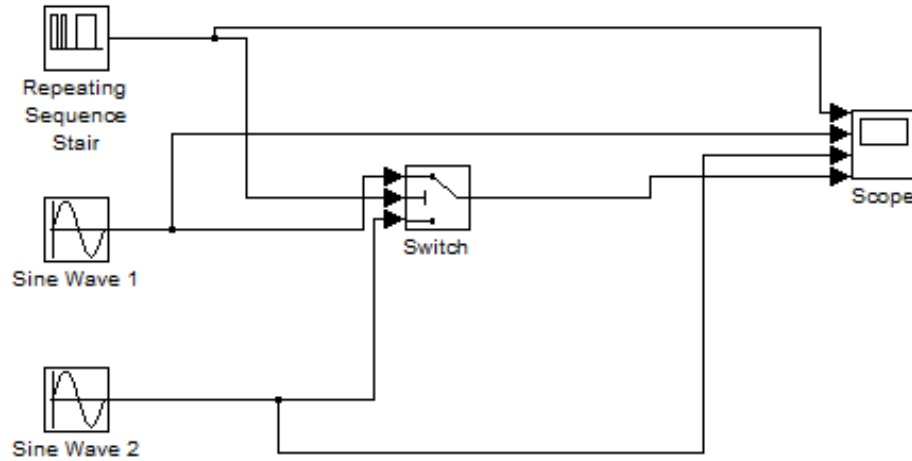
1. Objasniti razliku između linearnih i nelinearnih modulacija.

2. Nabrojati osnovne digitalne modulacione postupke.

3. Nabrojati kriterijume za izbor modulacionog postupka.

## Практични део

U Simulink-u programskog paketa MatLab kreirati model kao na Slici 1.

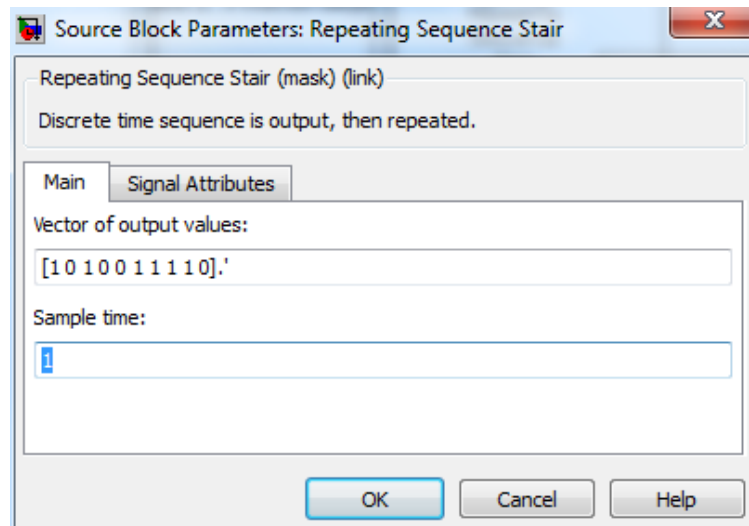


*Slika 7.1. Model kojim se simuliraju binarne digitalne modulacije*

### Prilog uz Sliku 1:

- Simulink / Sources / Repeating Sequence Stair

Podesiti *Vector of output values* 1 0 1 0 0 1 1 1 1 0 i *Sample time* =1 (videti Slika 2).



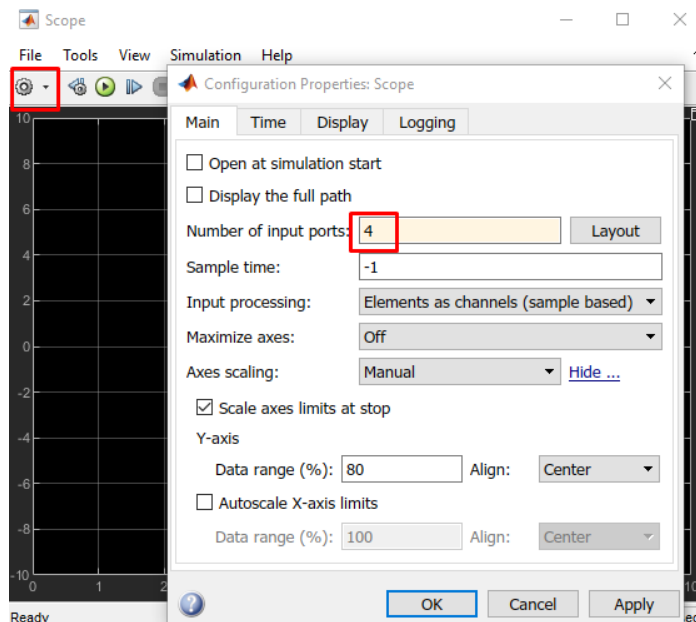
*Slika 7.2. Podešavanje bloka Repeating Sequence Stair*

- Simulink / Commonly Used Blocks / Switch

Podesiti *Threshold*=0.5

- Simulink / Sinks / Scope

Potrebno je podesiti da blok Scope ima četiri ulaza. U polju *Number of Axes* (ili *Number of input ports*), dakle, trebalo bi ukucati broj 4. Ovo podešavanje, nakon duplog klika na blok Scope, ilustrovano je na Slici 3.



*Slika 7.3. Podešavanje bloka Scope*

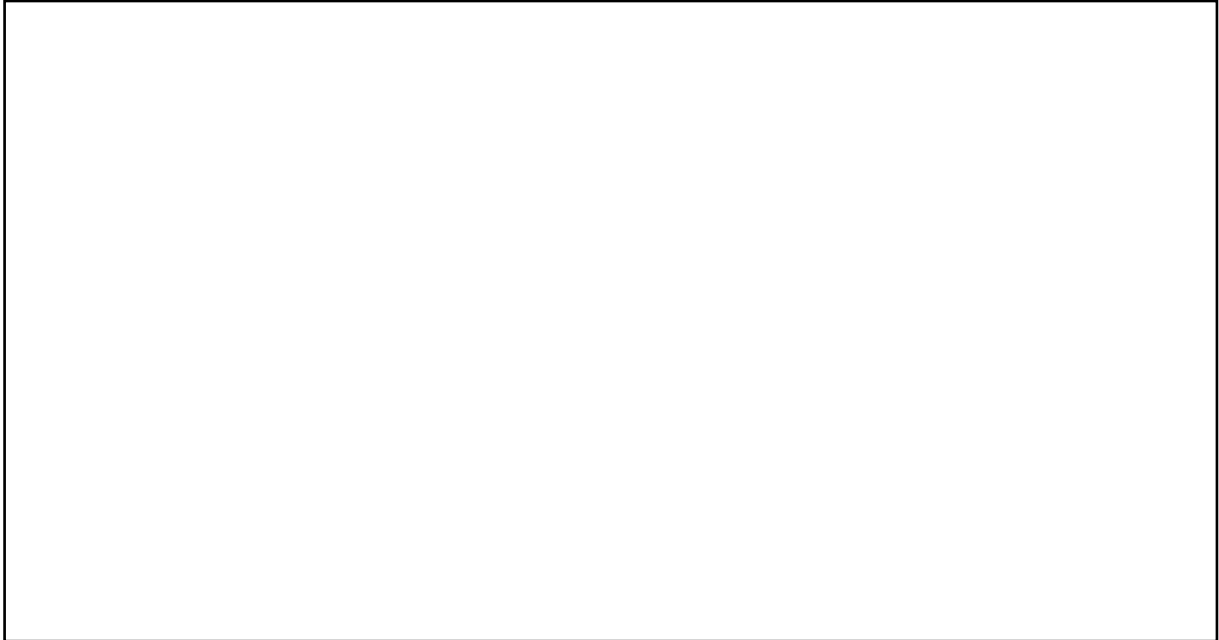
- Simulink / Sources / Sine Wave

Kod oba sinusna generatora podesiti *Sample Time* = 0.01.

Kod generatora **Sine wave 1** podesiti i sledeće: *Amplitude* = 1, *Frequency* =  $2 \cdot \pi$ , *Phase* = 0.

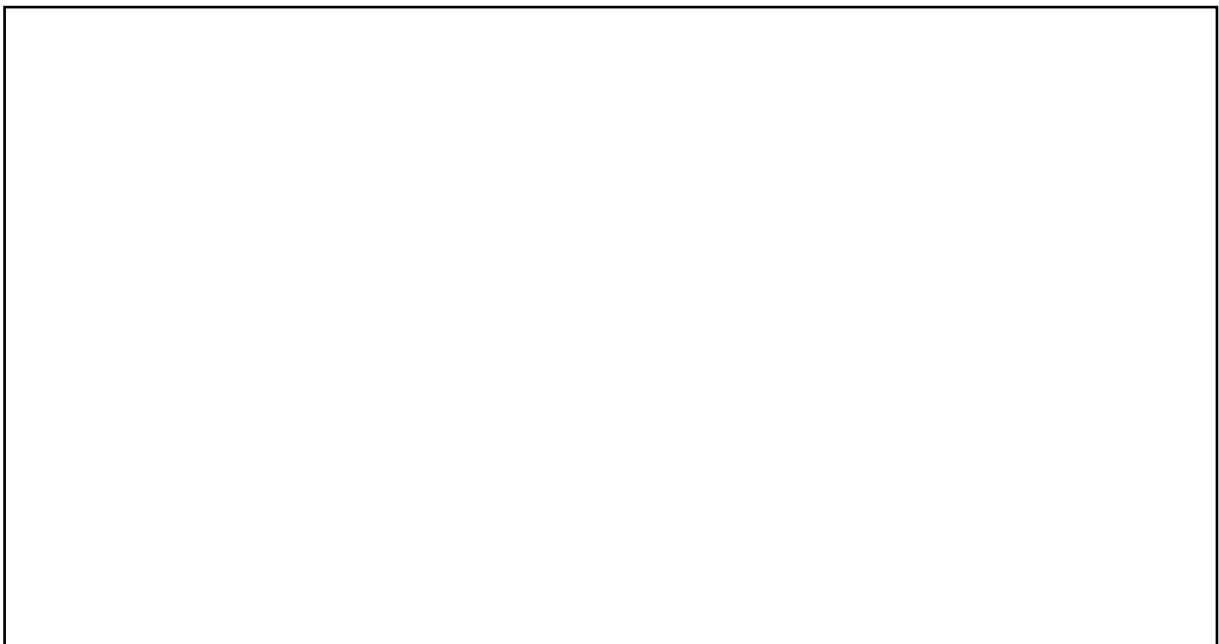
## BASK – Binarna amplitudska digitalna modulacija

**Zadatak:** Kod generatora **Sine wave 2** podesiti sledeće: *Amplitude* = 0. Nacrtati talasne oblike dobijene na bloku Scope.



## BPSK – Binarna fazna digitalna modulacija

**Zadatak:** Kod generatora **Sine wave 2** podesiti sledeće: *Amplitude* = 1, *Frequency* =  $2 \cdot \pi$ , *Phase* =  $\pi$ . Nacrtati talasne oblike dobijene na bloku Scope.



## BFSK – Binarna frekvencijska digitalna modulacija

**Zadatak:** Kod generatora **Sine wave 2** podesiti sledeće: *Amplitude* = 1, *Frequency* =  $4 \cdot \pi$ , *Phase* = 0. Nacrtati talasne oblike dobijene na bloku Scope.

