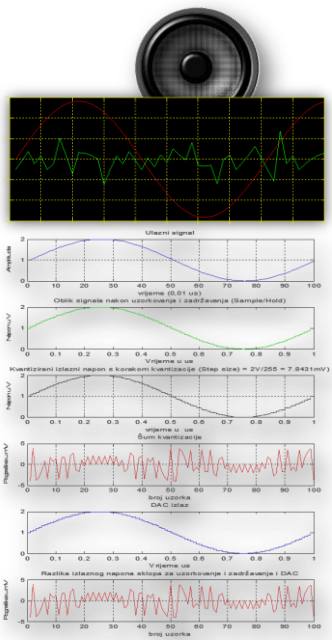
SPIE09 Obrada zvučnih signala

01_01 Kontinuirani i diskretni signali

Ozren Bilan, viši predavač

Kontinuirani i diskretni signali



Analogno digitalna i digitalno analogna pretvorba i višetaktni i naduzorkujući sustavi *str. 107*

1. UVOD
1.1. **Kontinuirani i diskretni signali**
1.2. Analiza u vremenskom i frekvencijskom području
1.3. DSP transformacije
1.4. Projektiranje digitalnih filtera
2. Zvučni signali visoke razlučivosti *HD Audio*
3. Digitalna obrada govora
4. Sažimanje zvučnih datoteka
5. Uvod u obradu zvučnih signala umjetnim neuralnim mrežama
6. Uvod u obradu i analizu zvučnih signala valićem *Wavelet*
7. Razlike DSP i procesora

30 sati predavanja + 30 sati laboratorijskih vježbi

Svi MATLAB kodovi nalaze se u skripti LAB VJEŽBE DOZS

DSP se razlikuje od ostalih područja računalske znanosti jer koristi posebne tipove podataka - **signale**. Signal **opisuje** kako se jedan parametar mijenja u odnosu prema drugom parametru. Temeljni **razlog obrade signala je izdvajanje informacije**

DSP se temelji na matematičkim algoritmima pa promjenom programa mijenjamo ponašanje sklopa. **Računalni programi su snažni** jer mogu izvršiti više od analognog sklopovlja (iako nisu uvijek jednako brzi), pa se simulacije mogu izvršavati na računalima (MATLAB, JAVA...). PC računalo s MATLAB-om je stvarni DSP sustav.

u pdf inačici nisu vidljive animacije

Što je DSP, a što je zvučni signal?

DSP je najsnažnijih tehnologija koja je oblikovala znanost na području istraživanja svemira, vojne tehnike, komunikacija, medicinske obrade slika, radara i sonara, **reprodukcije i prijenosa glazbe i govora**, istraživanja nafte i ruda, te mnogim drugim, a definira se kao **obrada diskretnih vremenskih signala**.

Svako područje primjene razvilo je vlastitu DSP tehnologiju s vlastitim algoritmima, primijenjenom matematikom i usko specijaliziranim tehnikama.

Kombinacija primijenjenog znanja i dubine zahvaćanja svakog od navedenih područja onemogućava analizu svih razvijenih DSP tehnologija, pa se uvod u DSP sastoji od dvije zadaće:

- usvajanje **općih koncepata DSP** koji se primjenjuju u cjelini
- usvajanje specijalnih tehnika za posebno područje interesa - **zvučne signale**.

Prva računala 70-ih godina bila su skupa pa se primjena DSP ograničila na vojnu, geološku i medicinsku primjenu (radar, sonar, istraživanje nafte, medicina).

80-tih godina računala postaju mnogo jača i dostupnija pa se primjena širi na potrošačku elektroniku: mobilnu telefoniju, prijenos i obradu glazbe i govora...

Ozren Bilan

Područje primjene DSP

TEORIJA KOMUNIKACIJA
NUMERICKA ANALIZA
VJEROJATNOST I STATISTIKA
ANALOGNA OBRADA SIGNALA
TEORIJA ODLUCIVANJA
DIGITALNA ELEKTRONIKA
ANALOGNA ELEKTRONIKA

DIGITALNA OBRADA SIGNALA

PRIMJENA:
SVEMIR
VOJSKA
MEDICINA
KOMERCIJALNA
KOMUNIKACIJE
INDUSTRIJSKA
ZNANSTVENA

DSP je **multidisciplinarna znanost** jer koristi razna područja znanosti, inženjeringa i matematike

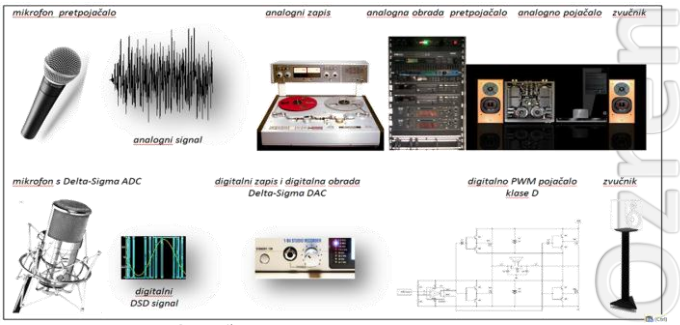
Ozren Bilan

Najopćenitije, DSP se definira kao **obrada diskretnih vremenskih signala**. Međutim, ova kratka definicija ne uključuje ogroman utjecaj DSP na tekuće tehnologije i mogućnosti korisnika koji mogu obrađivati bilo koji oblik informacije.

Najednostavnijim rječnikom, privlačnost postupaka digitalne obrade signala potječe iz činjenice da takav pristup **pretvara bilo koji informatičko inžinjerski problem u matematički**. U takvoj situaciji, ako postoji matematičko rješenje u bilo kojem obliku, onda ga je moguće rješiti digitalnom obradom signala.

Rješenje je **algoritam – postupak koji se primjenjuje programski, koji je točan i ponovljiv, a ne ovisi o sklopovlju, uvjetima okoline te se vrlo lako može promjeniti i poboljšati**. Štoviše, postupci se mogu mijenjati u realnom vremenu za različite uvjete što se ne može realizirati tradicionalnom analognom elektronikom. Skoro sve što se može realizirati analognom elektronikom, može se još bolje napraviti primjenom digitalne obrade signala, a neki postupci, kao npr. frekvencijski selektivni filteri mogu se primijeniti samo digitalnom obradom signala. **Pojačala, oscilatori, modulatori, filteri, itd. mogu se realizirati digitalnom obradom** signala zato jer se **njihova osnovna funkcionalnost može opisati matematičkim relacijama**.

Ta činjenica potpuno mijenja prirodu i postupke projektiranja elektroničkih sustava. Pogledajmo primjer kako **način formatiranja zvučnog signala mijenja način obrade i pojačanja signala**.



Ozren Bilan

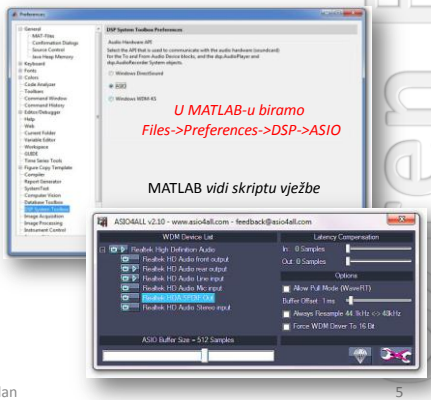
Matematička strana problema digitalne obrade signala može se činiti teška, međutim u najvećem broju slučajeva **složeniji matematički postupci i njihova programska primjena već je napravljena u programskim paketima poput MATLAB**-a.

Budući da je DSP predstavljena matematičkim i računalnim algoritmima, potpuno je neovisna o platformi računala.

Ne samo da je digitalna obrada zvučnih signala u MATLAB-u stvarna DSP u bilo kojem smislu, nego **obrađene signala možemo čuti vrhunskom kvalitetom** jer

osobno računalo s MATLAB-om PREDSTAVLJA STVARNI DSP PROCESOR.

Činjenica da postoje **DSP čipovi i procesori**, koji se koriste u mnogim profesionalnim primjenama, nije relevatna za shvaćanje i usvajanje specifičnih znanja DSP tehnologije jer se obrada izvršava **programski** ili **sklopovski**. Današnja računalna tehnologija je temeljena na primjeni **tranzistorskih binarnih platformi** pri čemu temeljni principi DSP ne ovise o tehnologiji ni na koji način. **ASIO4ALL** – je **univerzalni driver** za *Windows* operacijske sustave koji omogućava **direktnu reprodukciju digitalnog audio zapisa na razne uređaje**; npr. USB DAC. Predstavlja jedan od oblika *Kernel Streaminga*. Karakterizira ga vrlo mala latencija i neovisan je o hardveru. To znači da ga možemo koristiti na svim računalima.



Najvažnije

- DSP se razlikuje od ostalih područja računalske znanosti jer koristi posebne tipove podataka - **signale**.
- Signal **opisuje** kako se jedan parametar mijenja u odnosu prema drugom parametru
- Temeljni **razlog obrade signala je izdvajanje informacije**
- **DSP se temelji na računalskim programima** pa promjenom programa mijenjamo ponašanje sklopa
- **Računalni programi su snažni** jer mogu izvršiti više od analognog sklopovlja (iako nisu uvijek jednako brzi), pa se simulacije mogu izvršavati u računalnim programima (MATLAB, JAVA...)
- Za primjene u **realnom vremenu** koriste se **DSP čipovi - nisu skupi i slični su µ-procesorima** gdje je moguće ugraditi veliki broj različitih funkcija u jedan čip
- **Točnost im možemo podesiti po volji**, a **vremenski im se karakteristike ne mijenjaju** jer su vrlo stabilni
- osobno računalo s **MATLAB**-om je **STVARNI DSP PROCESOR**.

Što je signal?

- **protok informacije**
- **mjerena kvantiteta koja se mijenja** vremenski ili položajno
- **električni signal primljen iz pretvarača**:
  (*mikrofon, termometar, akcelerometer, antena, itd*…)
- **električni signal koji upravlja nekim procesom**

Kakvi mogu biti signali stvarnog svijeta?

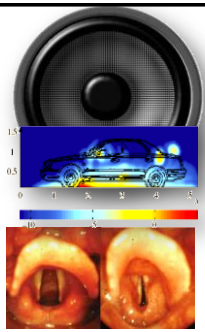
- **mehanički neoscilatorni**
  (*pomak, kutna brzina, brzina, sila, moment, tlak*...)
- **akustički signali**
  (*mehaničke vibracije, zvučni valovi, vodeni i površinski valovi*...)
- **električni signali**
  (*naponi, struje, električna i magnetska polja*...)

Što je sustav?

- **bilo koji proces koji generira izlazni signal kao odziv na ulazni signal pobude.**

Ozren Bilan

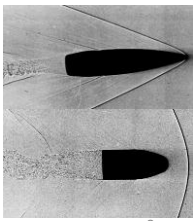
Kakvim procesima nastaje zvuk?



1) **Titranjem tijela**: Pri titranju membrana zvučnika ili dijelova bučnih strojeva dolazi do pomaka čestica zraka i promjene zvučnog tlaka.

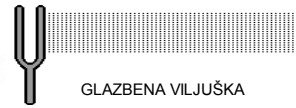
2) **Promjenom zračnog protoka**: kad govorimo glasnice se otvaraju i šire pri čemu propuštaju zrak. U sireni rupice na rotirajućoj ploči naizmjenično blokiraju i propuštaju zrak što daje vrlo glasan zvuk

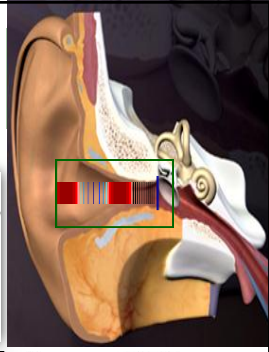
3) **Vremenski promjenjivim izvorom topline**: električna iskra daje zvuk; eksplozija daje zvučni udar zbog naglog širenja zvuka uslijed brzog zagrijavanja. Grom je posljedica munje koja zagrijava zrak.

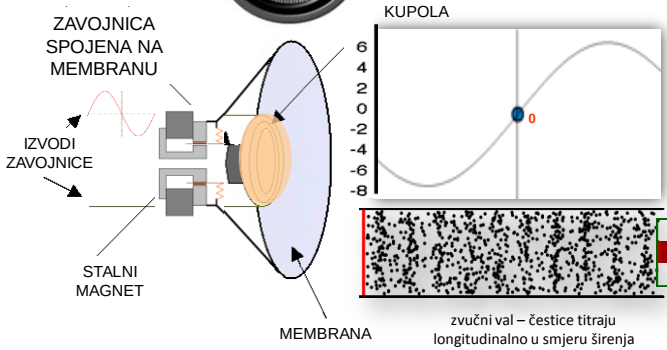


4) **Nadzvučnim protokom**: kad nadzvučni avion ili ispaljeni metak prisile zrak na gibanje brže od brzine zvuka nastaje udarni val

ZVUK

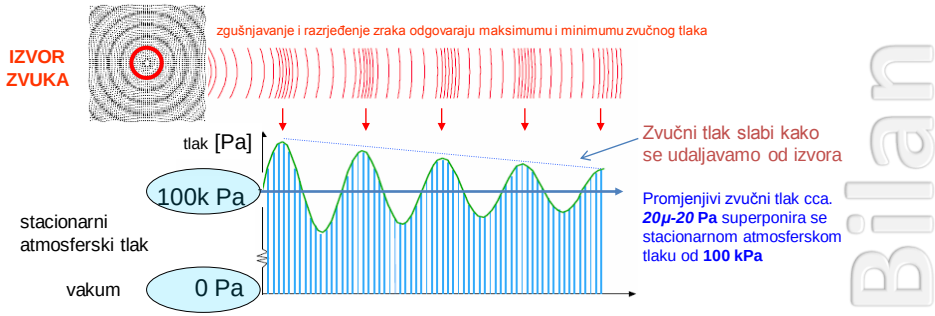




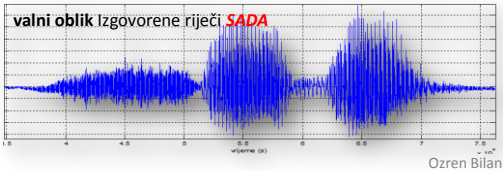


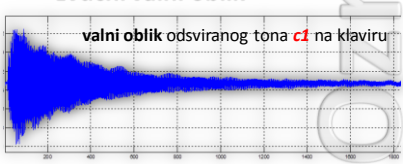
Lokalne promjene zvučnog tlaka superponirane atmosferskom tlaku šire se od izvora. Dolaskom zvučnih valova membrana titra. Mikrofon daje na izlazu kontinuirani napon proporcionalan promjeni zvučnog tlaka.

Ozren Bilan



- Lokalne promjene zvučnog tlaka izvora superponirane atmosferskom tlaku šire se od izvora
- Dolaskom zvučnih valova membrana mikrofona ili bubnjić uha titra
- Mikrofon daje na izlazu kontinuirani napon proporcionalan varijaciji zvučnog tlaka
- Dijagram vremenske promjene napona nazivamo **zvučni valni oblik**.





Ozren Bilan

Razlog obrade signala

Najvažniji razlog obrade signala iz stvarnog svijeta je dobivanje informacije.

Informacija je najčešće u obliku amplitude signala, frekvencije, spektralnog sadržaja, faze, vremenskih odnosa s drugim signalima.

Jednom kad dobijemo iz signala informaciju možemo je koristiti na različite načine.

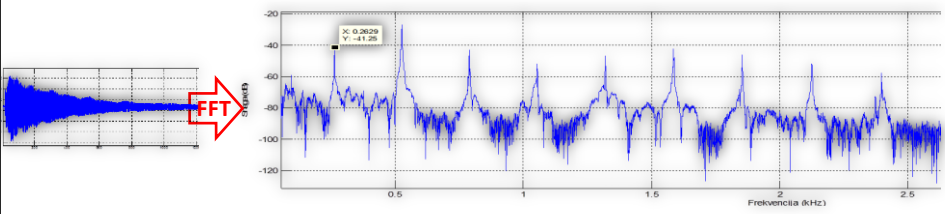
Drugi zahtjev za obradom signala je sažimanje frekvencijskog sadržaja signala, bez značajnijeg gubitka informacije, a nakon toga formatiranje i prijenos nižim brzinama. Tako se smanjuje potrebna frekvencijska širina kanala. Brzi modemi *High speed modems* i adaptivni modulacijski sustavi impulsnog koda (*ADPCM*) intenzivno koriste algoritme sažimanja, kao i digitalni mobilni radio sustavi, MPEG sustavi za snimanje i reprodukciju, te *HD* **zvučni signali visoke razlučivosti**.

RAZLOZI OBRADE SIGNALA

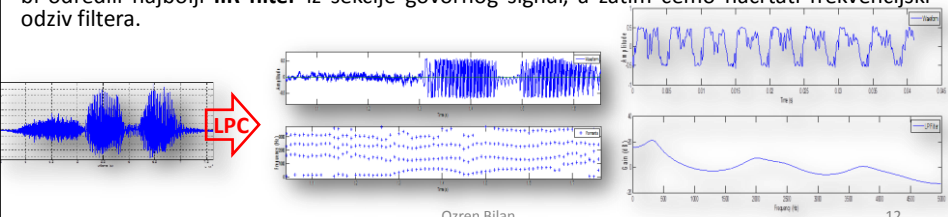
- Dobivanje informacije iz signala (*amplituda, faza, frekvencija, spektralni sadržaj, vremenski odnosi*)
- Reformatiranje signala (*telefonija*)
- Sažimanje podataka (*modemi, mobiteli, HDTV, MPEG*)
- Generiranje upravljačkih signala (*industrijsko procesno upravljanje*)
- Ekstrakcija signala iz šuma (*filtriranje, autokorelacija, konvolucija, valić, neuralne mreže*)
- Pribava i pohrana signala u digitalnom obliku za potrebe analize (*FFT tehnika*)

Najvažniji razlog obrade signala iz stvarnog svijeta je dobivanje informacije.

Npr. iz valnog oblika odsviranog tona možemo dobiti **frekvenciju fundamentala** kako bi vidjeli da li je klavir dobro naštiman. Iz **spektra harmonika** možemo odrediti tip klavira.



Iz govornog signala u slučaju estimacije frekvencije formanata zanimaju nas frekvencije rezonancije. Kako bi odredili najbolji model sustava koristimo postupak analize ***Linearna predikcija***. Model linearne predikcije **oblikuje signal kao da je generiran signalom minimalne energije koji je propušten kroz rekurzivni IIR filter**. Ideju ćemo prikazati primjenom **LPC** kako bi odredili najbolji **IIR filter** iz sekcije govornog signal, a zatim ćemo nacrtati frekvencijski odziv filtera.



Sustavi, signali, prilagođenje signala

DSP se bitno razlikuje od ostalih područja računalske znanosti jer koristi posebne tipove **podataka** - signale. U najvećem broju slučajeva **signale možemo osjetiti** jer izviru iz realnog svijeta oko nas u obliku zvučnih valova, slika, vibracija, napona, temperature itd.

Signal ima **tri bitne karakteristike**:

- to je **osjetna ili mjerljiva fizikalna veličina ili impuls**
- ima oblik **napona, struje, jačine magnetskog polja**
- njime se mogu **prenijeti poruke i informacije**.

Svi signali realnog svijeta su analogni,
bez obzira jesu li **istosmjerni, izmjenični ili u obliku impulsa**.

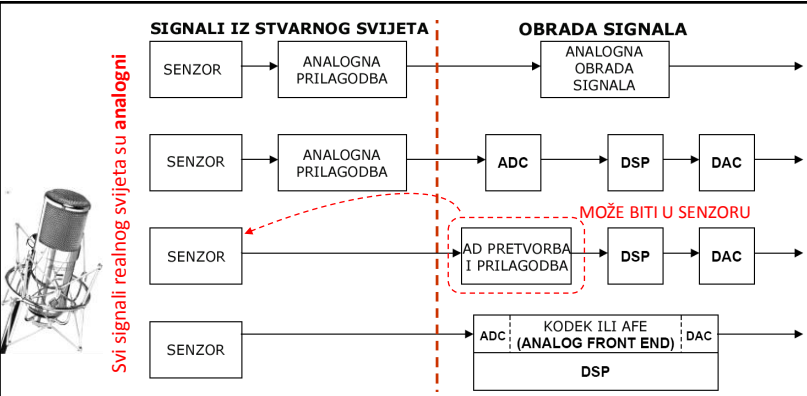
Razlikujemo *analogne* i *digitalne* signale na slijedeći način - u prirodi analogne varijable uključuju sve mjerljive fizikalne veličine, a ovdje ćemo ih ograničiti na električne varijable, načine promjene, energiju i razine snage koje sadržavaju.

Ozren Bilan

KARAKTERISTIKE SIGNALA	MJERNE JEDINICE	
Signali su fizikalne veličine	**Temperatura**	°C
Signali su mjerljivi	**Tlak**	Newton/m^2
Signali sadrže informaciju	**Masa**	kg
Svi signali su analogni	**Napon**	Volt
	Struja	Amper
	Snaga	Watt

Sve fizikalne veličine pretvaraju se senzorima u električne signale.

Ozren Bilan



Posebni elektronički sklop, koji analogno obrađuje signal, vrši funkciju pojačanja (*gain*), odvajanja (*isolation*), detekcije u prisutnosti šuma (*CMRR, line driver, line receiver*), sažimanje dinamičkog područja (*logamp, PGA*), aktivno i pasivno filtriranje.

Što je mjesto AD pretvorbe bliže senzoru to je na AD pretvaraču veća zadaća analogne prilagodbe signala. Dodatna složenost AD pretvorbe može mnogo višom frekvencijom uzorkovanja, širim dinamičkim područjem, višom rezolucijom, eliminiranjem šuma na ulazu, filtriranjem na ulazu, pojačalima s programiranim pojačanjem (*PGA*), referencijalnim naponima u čipu, itd…pojednostavniti sustav i dodati funkcionalnost.

Razlog obrade signala

Sve fizikalne veličine pretvaraju se senzorima u električne signale.

Prilagođenje signala odnosi se na pripremanje signala iz realnog svijeta za obradu. Tu podrazumijevamo senzore (npr. temperature i tlaka), odvojna i instrumentacijska pojačala itd. Npr. povratni signal radara ili ultrazvučnog sustava slika je primjer signala koji su poslijedica poznatog emitiranog signala.

Različiti od njih su *digitalni signali* gdje se stvarni signal prilagođava i formatira u binarne brojeve. Ti digitalni signali mogu ali ne moraju biti povezani sa analognim varijablama realnog svijeta; kao npr. podaci u LAN ili drugim mrežama.

U slučaju digitalne obrade signala analogni signali pretvaraju se u binarni oblik AD pretvaračem. Izlaz AD pretvarača je binarna prezentacija analognog signala, a DSP njome matematički manipulira. Nakon obrade, tako dobivena informacija može se ponovno pretvoriti u analogni oblik korištenjem DA pretvarača.

Drugi ključni koncept uključen u definiciju *signala* je taj što signal sadrži *informaciju*. To dovodi do ključnog **razloga za obradom signala** – *izvlačenje (ekstrakcija) informacije.*

Ozren Bilan

DSP je sadržana u matematici, algoritmima i tehnikama koje se koriste za manipulaciju signalima, nakon što smo ih pretvorili u digitalni oblik. Korisnik tako ostvaruje svoje najvažnije ciljeve:

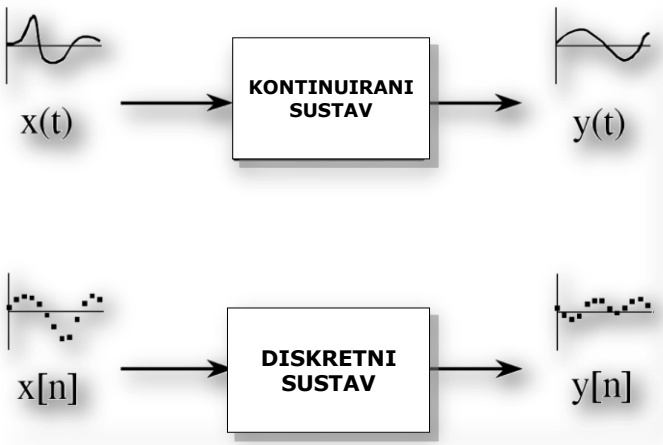
- **prepoznavanje i generiranje govora**
- **najviša vjernost reprodukcije zvučnih signala**
- **sažimanje podataka za pohranu i prijenos**
- **poboljšanje slika**

Signal opisuje kako se neki parametar mijenja promjenom drugog parametra. Npr. kako se vremenski mijenja napon u elektroničkom sklopu.

Sustav je bilo koji proces koji generira izlazni signal kao odziv na ulazni signal pobude.

Objasnit ćemo terminologiju i način označavanja…

Ozren Bilan



Terminologija: **sustav je bilo koji proces koji generira izlazni signal kao odziv na ulaznu pobudu**. Kontinuirani signali označeni su malom zagradom, a diskretni uglatom zagradom.

Signali se označavaju malim slovom, dok se velika slova koriste za označavanje frekvencijskog područja: ulazni signal se označava x(t) ili x[n], a izlazni y(t) ili y[n].

Kontinuirani signali označeni su malom zagradom, npr

$x(t)$ i $y(t)$.

Diskretni uglatom zagradom, npr

$x[t]$ i $y[t]$.

Signali se označavaju **malim slovom**, dok se **velika slova** koriste za **označavanje frekvencijskog područja**. Ime signala najčešće opisuje parametar koje predstavlja. Vremenski promjenjiv napon može biti *v(t),* dnevna promjena cijena dionice *p[d].*

Međutim, signali i sustavi mogu se označavati bez točnog saznanja koji parametar predstavljaju. Tada se koriste varijable *x* i *y* kao u algebri, bez pridruživanja fizikalnog značenja; kontinuirani ulazni signal se označava *x(t)*, a izlazni *y(t)*. Kod diskretnih sustava ulazni signal se označava x[n], a izlazni y[n].

Mnogo je razloga zbog kojih želimo shvatiti princip rada sustava. Često se događa da se ulazni signal prijenosnog sustava razlikuje od izlaznog. Razumijemo li prijenosni sustav moguće je kompenzirati ove nedostatke (npr. eliminirati jeku razglasnog sustava, izdvojiti šum elektrokardiograma…).

Drugim riječima, **sustav predstavlja fizikalni proces kojeg želimo proučiti ili analizirati**. Radar i sonar su dobri primjeri jer oba temelje rad na usporedbi emitiranog i reflektiranog signala u cilju dobivanja informacije o udaljenom objektu. Rječnikom teorije sustava, problem se svodi na **određivanje sustava koji obavlja transformaciju emitiranog signala u primljeni signal**.

Na prvi pogled čini se da je teško razumjeti sve moguće sustave, međutim najveći broj njih pripada kategoriji koju nazivamo –

linearni sustavi.

Bez poznavanja koncepta linearnih sustava morali bi smo ispitivati individualne karakteristike mnogih nepovezanih sustava. Pristupom analize linearnog sustava obuhvatit ćemo sve linearne sustave u jednu kategoriju kao cjelinu. Prvo moramo identificirati

svojstva koja čine sustav linearnim

Zahtjevi linearnosti

Sustav je linearan ako pokazuje dva matematička svojstva: **homogenost** i **aditivnost**. Ako ne pokazuje oba svojstva dokazali smo da je nelinearan.

Treće svojstvo **translacijske (pomakne) nepromjenjivosti** *shift invariance*, nije strogi zahtjev za linearnosti, ali je nužan za najveći broj DSP tehnika.

Kad god netko navede pojam **linearni sustav za DSP**, treba podrazumijevati da ispunjava i treće svojstvo.

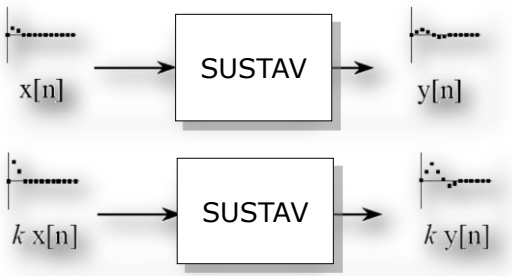
Zahtjevi linearnosti su:

- **homogenost**
- **aditivnost**
- **translacijska ili pomakna nepromjenjivost**

HOMOGENOST

AKO

ONDA VRIJEDI



Definicija homogenosti. **Sustav je homogen ako promjena amplitude na ulazu rezultira identičnom promjenom amplitude na izlaz**. Dakle, onda ako x[n] daje y[n], onda kx[n] daje ky[n], za svaki signal, x[n] i svaku konstantu k.

Slika pokazuje kako homogenost znači da će **promjena amplitude ulaznog signala rezultirati promjenom izlaznog signala**. Matematički, ako ulazni signal x[n] daje na izlazu izlazni signal y[n], onda će ulaz kx[n] dati na izlazu ky[n], za svaki ulazni signal i konstantu, k.

Primjer

Otpornik je dobar primjer i homogenog i nehomogenog sustava.

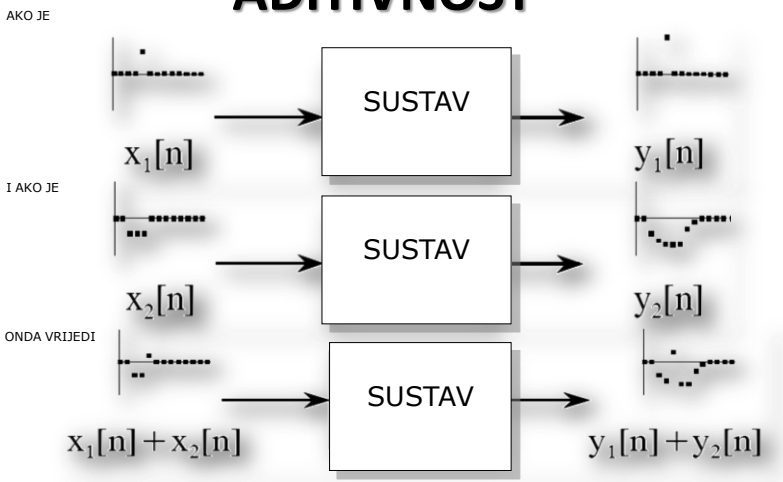
Ako je **ulaz sustava napon na otporu**, *v(t),* a **izlaz sustava struja** kroz otpor *i(t)*, **sustav je homogen**. To nam jamči Ohmov zakon; **povećanje i smanjenje napona će linearno povećati i smanjiti struju**.

Pogledajmo drugi sustav, gdje je **ulazni signal napon na otporu**, *v(t)* ali je **izlazni signal snaga** na otporu, *p(t).*

Budući da je snaga proporcionalna kvadratu napona, povećamo li ulazni signal za *faktor 2*, izlazni signal će se povećati za *faktor 4*. Sustav **nije homogen i zato nije linearan.**

Ozren Bilan

ADITIVNOST



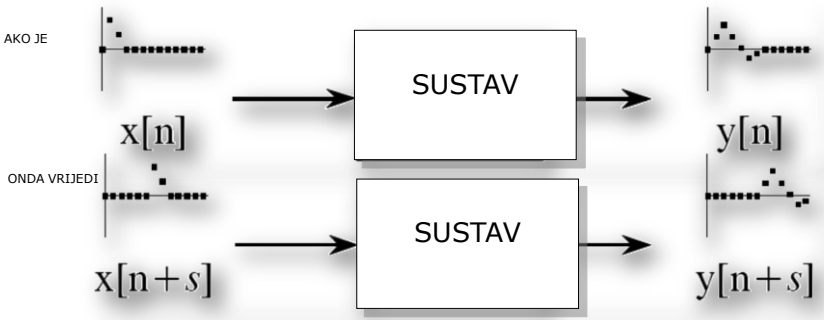
Definicija aditivnosti. **Sustav je aditivan ako sumirane signale propušta bez interakcije**. Tj., ako x1[n] daje y1[n] i ako x2[n] daje y2[n], onda će x1[n]+x2[n] dati y1[n]+y2[n].

Svojstvo aditivnosti pokazuje slika. Pretpostavimo sustav kojem ulazni signal $x_1[n]$ daje izlazni signal $y_1[n]$, a drugi ulazni signal, $x_2[n]$, daje, $y_2[n]$. Sustav je **aditivan** ako ulaz $x_1[n]+x_2[n]$ daje izlaz $y_1[n] + y_2[n]$, za sve moguće ulazne signale. Dakle, **signal dodan na ulazu, daje signal dodan na izlazu.**

Važno je naglasiti kako sumirani signali prolaze sustavom bez interakcije. Npr. poštar pri ulazu u zgradu pozvoni na nekoliko stanova istovremeno. Može se dogoditi da je vaš susjed već započeo razgovor s poštarom u trenutku kada ste prislonili slušalicu interfona. Govore li poštar i susjed istovremeno, jasno će te razabrati i susjeda i poštara jer je zvuk kojeg čujete suma dva glasa koji bi potpuno isto zvučali kao kada bi se prenosili individualno.

Primjer **neaditivnog sklopa** je miješalo radio predajnika. Tu se zbrajaju audio signal govora ili glazbe i noseći val visoke frekvencije a zatim se primjenjuje nelinearnost PN diode. Tako se stvara treći signal modulirani radio val koji prenosi informaciju na velike udaljenosti. Ne možemo radio val dovesti na zvučnik bez da ga **demoduliramo**, što znači da moramo izdvojiti zvučnu informaciju. (vježba Matlab str. 149.)

Vremenska (translacijska) nepromjenjivost



Definicija vremenske nepromjenjivosti vremenskog pomaka. **Sustav je invarijantan ako pomak ulaznog signal uzrokuje identični pomak izlaznog signala**. Matematički, ako x[n] daje y[n], onda x[n+s] daje y[n+s], za bilo koji signal, x[n] i svaku konstantu s (koristi se i oznaka **k**).

Ozren Bilan

Pomak ulaznog signala rezultira identičnim pomakom izlaznog signala. Formalnim izrazom, ako ulazni signal x[n] daje izlaz y[n], ulazni signal x[n+s] dat će izlaz y[n+s], za svaki ulazni signal i za svaku konstantu, s.

Dodavanjem konstante, s, nezavisnoj varijabli, n, valni oblik može se pomaknuti unaprijed ili unazad u horizontalnom smjeru. Ako je k=2, signal se pomiče lijevo za dva uzorka; kada je k=-2, signal se pomiče desno za dva uzorka.

Vremenska nepromjenjivost je važna jer pokazuje kako se karakteristike sustava vremenski na mijenjaju ili se ne mijenjanju promjenom nezavisne varijable.

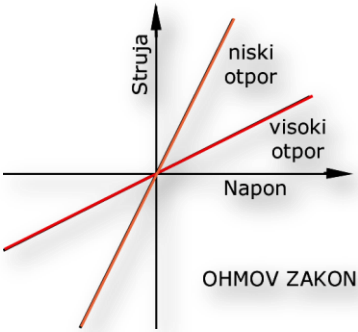
Ako A na ulazu, daje B na izlazu, možemo biti potpuno sigurni da će bilo kada, bilo koji A, dati na izlazu B.

Najveći broj sustava je **vremenski nepromjenjiv** što je korisno jer je teže raditi sa **vremenski promjenjivim** sustavima.

Statička linearnost i vjernost sinusnog valnog oblika

Homogenost, aditivnost i vremenska nepromjenjivost su vrlo važne karakteristike jer omogućavaju matematičke temelje za definiranje linearnih sustava. Svojstva sama po sebi ne omogućavaju korisnicima intuitivno poimanje linearnih sustava, pa se pomažemo pojmovima statičke linearnosti i vjernosti sinusoidalnog oblika. Ti parametri nisu toliko važni s matematičkog stajališta, nego su bliži ljudskom poimanju linearnih sustava.

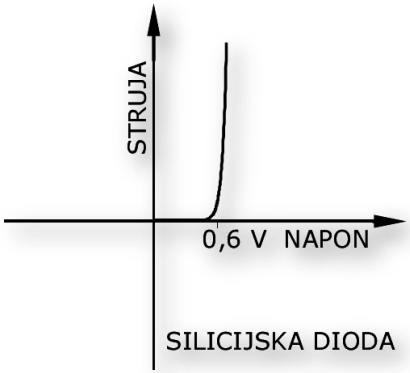
Statička linearnost određuje način odziva linearnog sustava dok se ulazni signal ne mijenja. **Statički odziv linearnog sustava je krajnje jednostavan; izlaz je jednak ulazu pomnoženom s konstantom**. Dijagram mogućih ulaznih vrijednosti prikazan u ovisnosti o izlaznim vrijednostima je pravac koji prolazi kroz ishodište koordinatnog sustava. Na slici 1) prikazana su dva linearna sustava: Ohmov zakon i Hookov zakon. Slika 2) pokazuje statičke odnose dva nelinearna sustava; PN spojne diode i magnetska svojstva željeza.

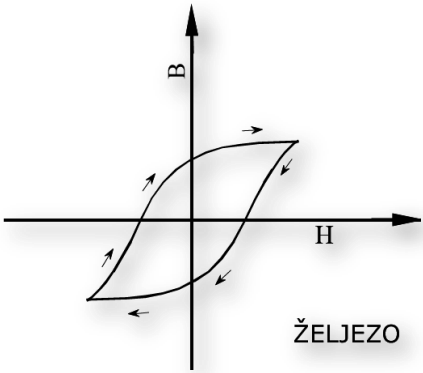


Primjeri statičke linearnosti

(a) Ohmov zakon: struja kroz otpor jednaka je padu napona na otporniku podijeljeno s vrijednošću otpora.

(b) Hookov zakon: elongacija opruge jednaka je primijenjenoj sili pomnoženoj koeficijentom krutosti opruge.





Primjeri istosmjerne nelinearnosti

(a) silicijska dioda ima eksponencijalni odnos između napona i struje.

(b) odnos magnetskog intenziteta H, i gustoće toka B, željeza ovisi o prethodnom stanju uzorka - histereza.

Statička linearnost određuje način odziva linearnog sustava dok se ulazni signal ne mijenja.

Statički odziv linearnog sustava je krajnje jednostavan; izlaz je jednak ulazu pomnoženom s konstantom.

Dijagram mogućih ulaznih vrijednosti prikazan u ovisnosti o izlaznim vrijednostima je pravac koji prolazi kroz ishodište koordinatnog sustava. Na slikama smo prikazali dva linearna sustava: Ohmov zakon i Hookov zakon. Druga slika je pokazala statičke odnose dva nelinearna sustava; PN spojne diode i magnetska svojstva željeza.

Svi linearni sustavi imaju svojstvo statičke linearnosti. Obrnuto može biti ali ne uvijek. Postoje sustavi koji pokazuju statičku linearnost, ali nisu uvijek linearni u odnosu na promjenjive signale. Vrlo raširena skupina sustava može se u potpunosti shvatiti samom statičkom linearnošću. Kod tih sustava nije važno da li se ulazni signal mijenja ili je nepromjenjiv.

Njima izlaz sustava ovisi samo o trenutnom stanju ulaza, a ne ovisi o prethodnom stanju. Dakle, takvi sustavi **ne pamte.**

Trenutna struja otpornika ovisi samo o trenutnom naponu, a ne ovisi o načinu promjene. Sustav ima statičku linearnost i ne pamti prethodne promjene – slijedi da je linearan. To nam omogućava vrlo važan način razumijevanja i dokazivanja linearnosti ovih jednostavnih sustava.

Važna karakteristika linearnih sustava je način njihovog **ponašanja pri sinusnoj pobudi**, što nazivamo – **vjernost sinusnog valnog oblika**.

Ako je na ulazu linearnog sustava sinusni valni oblik i na izlazu će se pojaviti sinusni valni oblik, potpuno iste frekvencije.

Sinusoide su jedini valni oblik koji imaju ovo svojstvo. Npr., nema razloga da na izlazu linearnog sustava očekujemo kvadratni valni oblik, ako ga pobudiom kvadratnim valnim oblikom. Iako sinusoida na ulazu jamči sinusni valni oblik na izlazu mogu postojati razlike u amplitude i fazi. Linearni sustav onda se može opisati frekvencijskim odzivom, grafom koji pokazuje kako se pojačanje i faza sustava mijenjaju s frekvencijom.

Ako sustav uvijek daje sinusoidalni izlaz na sinusoidalnu pobudu, znači li to da je linearan? Odgovor je **niječan, iako su izuzetci rijetki i očiti**.

Pretpostavimo unutar uređaja sklop koji promatra ulazni signal sustava i sinusni generator koji generira izlazni signal. Kad narinemo sinusnu pobudu, sklop izmjeri frekvenciju i pobudi sinusni generator na korespondentni izlaz. Sustav nije linearan jer nije aditivan, što lako možemo dokazati pobudom sume dva sinusna oblika. Uređaj može dati na izlaz samo jedan sinusni oblik. Primjer nije lako smisliti iako postoje **sustavi fazno zatvorene petlje** (PLL) koji rade na ovakav način.

Da bi bolje osjetili što je to linearnost sustava, zamislimo da je naš zadatak ustanoviti da li je sustav linearan. Na ulaz sustava spojit ćemo sinusni generator, a na izlaz osciloskop. Postavit ćemo sinusnu pobudu i pogledati da li je na izlazu također sinusna pobuda. Da li je izlaz odrezan na vrhu i dnu sinusoide, da li su pozitivni i negativni poluval identičnog oblika, postoji li izobličenje pri prolazu kroz nulu, itd. Nakon toga ćemo mijenjati amplitude pobude i ustanoviti da li se izlazni oblik mijenja s tim u skladu. Možda će postojati amplitudne i fazne promjene ali one su dopuštene kod linearnih sustava. Konačno, ako potvrdimo sve navedeno, zaključit ćemo kako je riječ o linearnom sustavu. Zaključak nije strogo matematički temeljen, ali točnost zaključka bit će vrlo visoka.

Primjeri linearnih sustava

Primjeri linearnih sustava

- *Valna propagacija* zvuka i elektromagnetskih valova
- *Električni sklopovi* sastavljeni od otpornika, kondenzatora i induktiviteta
- *Elektronički sklopovi* kao što su pojačala i filtri
- *Mehaničko gibanje* uslijed djelovanja masa, eleastičnosti i viskoznog prigušenja
- *Sustavi opisani diferencijalnim jednadžbama* kao što su RLC mreže
- *Množenje konstantom* kao što je pojačanje ili gušenje signala
- *Promjene signala* kao što je jeka, rezonancija, zamagljenost slike, itd.
- *Jedinični sustavi* kojima je izlaz uvijek jednak ulazu
- *Nul sustav* kojemu je izlaz uvijek jednak nuli, bez obzira na ulaz
- *Diferenciranje i integriranje* i analogne operacije *first difference i running sum* za diskretne signale
- *Male perturbacije* u inače nelinearnim sustavima, npr., mali signal pojačan ispravno polariziranim tranzistorom
- *Konvolucija*, matematička operacija koja svaku vrijednost na izlazu izražava kao sumu vrijednosti ulaza pomnoženu nizom težinskih koeficijenata
- *Rekurzija*, tehnika slična konvoluciji, s razlikom što se prethodno izračunate vrijednosti izlaza koriste u sumiranju s vrijednostima ulaza

Primjeri nelinearnih sustava

- *Sustavi koji nemaju statičku linearnost*, npr., napon i snaga otpora: $P = V^2R$ ili radijantna energijska emisija toplih tijela ovisna o temperature: $R = kT^4$, intenzitet svjetlosti transmitirane kroz prozirni materijal određene debljine: $I = e^{-\alpha T}$, itd.
- *Sustavi koji nemaju sinusoidalnu vjernost*, kao što su elektronički sklopovi: vršni detektori, kvadriranje, pretvarači valnih oblika, podvostručenje frekvencije, itd.
- *Uobičajena elektronička izobličenja* kao što su odrezivanje, preskočno izobličenje, *slewing*
- *Množenje jednog signala drugim*, kao što je amplitudna modulacija ili automatska kontrola pojačanja
- *Histerezne pojave*, kao što je ovisnost gustoće magnetskog toka o jačini polja u feromagnetskim materijalima.
- *Saturacija ili zasićenje*, u elektroničkim pojačalima i transformatorima pri jakoj pobudi
- Sustavi s pragom, npr. digitalna logička vrata ili seizmičke vibracije koje mrve promatranu stijenu.

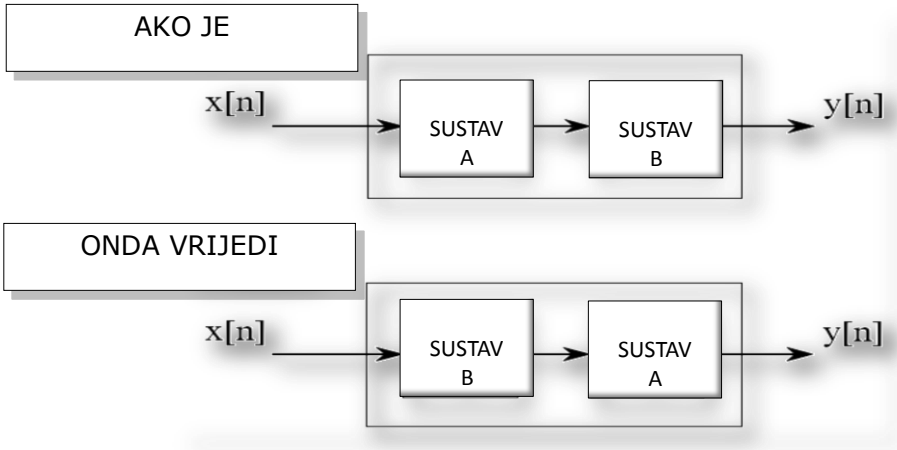
Posebna svojstva linearnosti

Linearnost je komutativna. To je svojstvo koje uključuje kombinaciju dva ili više sustava.

Ideja komutativnosti: pretpostavimo dva sustava u kaskadi, tako da je izlaz prvog sustava ulaz drugog sustava. Ako je svaki sustav linearan tada će i ukupna kombinacija također biti linearna. Komutativno svojstvo pokazuje da se poredak sustava u kaskadi može promijeniti bez utjecaja na karakteristike cjelokupne kombinacije. Ovo svojstvo poznajemo iz elektroničkih sklopova. Pretpostavimo sklop od dva stupnja, jedan za pojačanje, a drugi za filtriranje.

Pitamo se: **što je bolje**, prvo pojačati pa filtrirati ili filtrirati pa pojačati?

Ako su oba stupnja linearna **redoslijed nema utjecaja i rezultat će biti isti**. Pri tome moramo imati na umu kako će stvarna elektronika unijeti nelinearne efekte: interferencije, istosmjerni offset, šum i izobličenja.



Svojstvo komutacije linearnih sustava: **ako su dva ili više linearnih sustava u kaskadi, redoslijed sustava ne djeluje na karakteristike ukupne kombinacije** i svaki sustav s višestrukim ulazima i izlazima je linearan ako je sastavljen od linearnih sustava i sumacija signala.

Problemi mogu nastati pri **množenju** signala jer množenje može biti linearno i nelinearno, u ovisnosti o tome s čime množimo signal.

Množimo li ulazni signal **konstantnom** imamo linearni sustav. Takav sustav nazivamo **pojačalo** ili **atenuator**, u ovisnosti da li je konstanta veća ili manja od 1.

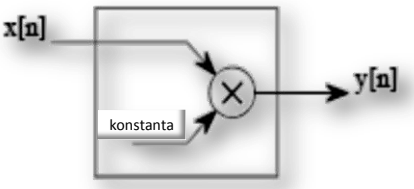
Množimo li ulazni signal drugim signalom, nastaje nelinearni sustav. Npr. množimo li sinusoidu drugom sinusoidom različite frekvencije, takav nelinearni sustav onda nazivamo **modulator**.

Drugi problem nastaje u odnosu prema parazitnim signalima kao što su istosmjerni pomak i termički šum.

Da li je sumiranje ovih signala linearno ili nelinearno?

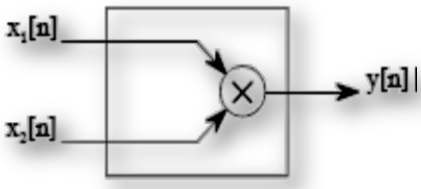
Pretpostavimo li da ovi signali **nastaju unutar sustava** proces je nelinearan jer čisti sinusoidalni signal neće rezultirati čistom sinusoidom na izlazu.

Pretpostavimo li da ovi parazitni signali **ulaze u sustav na posebnom ulazu sustava s više ulaza** riječ je o linearnom procesu koji zadovoljava zahtjev adicije.



LINEARNI SUSTAV

Množenje s konstantom



NELINEARNI SUSTAV

Množenje dva signala

Linearnost multiplikacije:

- **množenje signala konstantom je linearna operacija**
- **množenje dva signala je nelinearna**

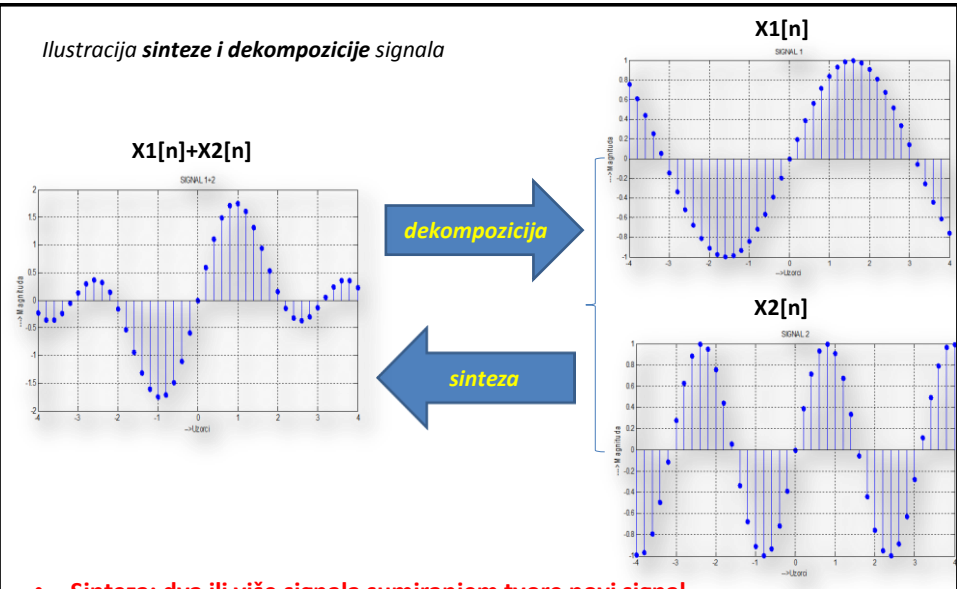
Superpozicija i sinteza: temelj digitalne obrade signala

Pri radu s linearnim sustavima jedini način kombiniranja signala je skaliranje (množenje signala konstantom) i nakon toga sumiranje. Npr. signal se ne može množiti drugim signalom. Slika pokazuje primjer: tri signala: *x0[n]*, *x1[n]*, i *x2[n]* sumiraju se u četvrti signal, *x[n]*. Ovakav **proces kombiniranja signala skaliranjem i sumiranjem** nazivamo **sinteza**.

Dekompozicija je inverzna operacija sinteze, gdje se signal rastavlja u dvije ili više aditivnih komponenti. Postupak je složeniji od sinteze jer postoji beskonačno veliki broj mogućih dekompozicija za svaki zadani signal. Npr. brojevi 15 i 25 mogu se sintetizirati (sumirati) samo u broj 40. Za usporedbu broj 40 može se dekomponirati (rastaviti) u: *1+39* ili *2+38* ili *-30.5+60+10.5*, itd.

Sad dolazimo do same biti DSP digitalne obrade signala: **superpozicije**, ukupne strategije za razumijevanje kako se signal ii sustavi analiziraju.

*Ilustracija **sinteze i dekompozicije** signala*



- **Sinteza: dva ili više signala sumiranjem tvore novi signal.**
- **Dekompozicija: obrnut proces, signal se rastavlja u dva ili više aditivnih komponenti signala**.

Pretpostavimo ulazni signal, x[n], koji prolazi kroz linearni sustav i rezultira izlaznim signalom, y[n]. Prema ilustraciji, ulazni signal može se rastaviti u niz jednostavnijih signala: *x0[n], x1[n], x2[n]*, itd. Nazvat ćemo ih **komponente** ulaznog signala.

Nadalje, svaka ulazna komponenta signala individualno se propušta kroz sustav, što daje grupu izlaznih komponenti: *y0[n], y1[n], y2[n],* itd. Komponente izlaznog signala **sintetiziraju** su u izlazni signal, y[n].

Najvažnije je što je izlazni signal dobiven ovom metodom **potpuno identičan** signalu koji bi nastao neposrednim prolazom ulaznog signala kroz sustav.

Ovo je vrlo važna ideja. **Umjesto da pokušamo razumjeti kako se složeni signali mijenjaju pri prolasku kroz sustav, sve što trebamo znati je kako se mijenjaju najjednostavniji signali**. Žargonom obrade signala, ulazni i izlazni signali promatraju se kao superpozicija (suma) jednostavnih valnih oblika. **Ova zamisao je temelj skoro svih tehnika obrade signala**.

Kao **najjednostavniji primjer superpozicije** napamet pomnožite 2041x4.

Kako to napraviti? Da li zamislimo 2041 šibicu, početverostručimo mentalnu sliku i pokušamo ih prebrojiti? Vrlo vjerojatnije je korištenje superpoziciju; 2041=2000+40+1. Svaku komponentu pomnožimo s 4 i sumiramo: 8000+160+4=8164.

Alternative linearnosti – što ako je sustav nelinearan? Homomorfna obrada

Ako je sustav nelinearan postoji samo jedna strategija koja nam je dostupna: linearizacija nelinearnog sustava. To je moguće na tri načina:

- **zanemarimo nelinearnost**. Ako je dovoljno mala, sustav aproksimiramo linearnim. Greška koja iz toga proizlazi se tolerira kao šum ili se ignorira.
- **zadržavamo vrlo male razine signala**. Mnogi nelinearni sustavi ponašaju se linearno ako su im amplitude signala vrlo male. Npr. tranzistori su nelinearni u radnom području, međutim linearno pojačavaju signale koji su reda veličine nekoliko mV.
- **primijenimo linearnu transformaciju**. Npr., dva signala se množe kako bi dali treći:

$$a[n] = b[n] \times c[n]$$

Logaritmiranjem izraza nelinearni proces množenja postaje linearni proces zbrajanja:

$$log(a[n]) = log(b[n]) + log(c[n])$$

Stručno ime ovog jednostavnog procesa je **homomorfna obrada signala,** a ima vrlo veliku primjenu.

Signali su **informacije** o podacima, glasu, zvuku, slikama i videu. Klasificiramo ih kao **analogne (vremenski kontinuirane)** ili **digitalne (vremenski diskretne).**

Obrada signala je primjena sklopova i sustava (sklopovlja i programa) koji djeluju na ulazne signale tako da ih mijenjaju na način koji mi to želimo – najčešće kako bi dobili **informaciju**.

Digitalni sustavi imaju prednosti nad analognim jer su imuni na šumove, lako se pohranjuju i prijenose.

Pretvorba analognog signala u digitalni oblik sastoji se od **uzorkovanja** u pravilnim intervalima, **kvantiziranja** uzoraka i **kodiranja** kvantiziranih vrijednosti u binarne znamenke.

Ako signal samo sampliramo dobivamo diskretne vremenski promjenjive signale. Međutim, za obradu signala u digitalnim sustavima potrebno je izvesti sva tri nabrojena koraka. Kvantizacija i kodiranje najčešće se podrazumijevaju, pa izrazi **diskretni vremenski promjenjivi** i izraz **digitalni signali** postaju ekvivalentni i zamjenjivi.

Dakle **digitalna obrada signala** (DSP) i **obrada diskretnih vremenski promjenjivih signala** (DTSP) znače isto.

Pored korisnih komponenti signali sadržavaju i neželjene komponente kao što su **šumovi, interferencije, jitter** … koje eliminiramo, minimiziramo ili potiskujemo.

Digitalni sustavi za obradu mogu biti jednostavni logički sklopovi, jednostavni programi ili kompleksne tvorevine sastavljene od sklopovlja i programa, kao što su računala.

Analizirat ćemo različite **digitalne sustave**, koji su **najčešće linearni i vremenski nepromjenjivi (LTI).** Takvi tipični sustavi su **filteri**.

VREMENSKI KONTINUIRANI ILI ANALOGNI SIGNALI

Signal je vremenska **promjena amplitude**. Amplituda može biti predstavljena naponom, strujom, snagom..., a u sklopovima i sustavima najčešće je to NAPON.

Analognim ili vremenski kontinuiranim sustavima amplituda se stalno mijenja.

Amplitudu generiraju **elektronički sklopovi**, prirodni izvori, kao što su **temperatura, glas, zvuk, video**.

U električni oblik pretvaraju je **senzori ili pretvarači**.

Signale opisujemo njihovim **valnim oblikom**. To su **grafičke ilustracije** za laku vizualizaciju promjene.

Umjesto da oblik signala opisujemo riječima ili crtamo njihov oblik, najobjektivniji i najkoncizniji način njihovog opisa je **matematički oblik**.

Kako bi shvatili ponašanje signala u vremenskom i frekvencijskom području nužno ih je predstaviti matematičkim oblikom pri analizi i projektiranju sklopova i sustava za digitalnu obradu.

Sinusoidalni signal

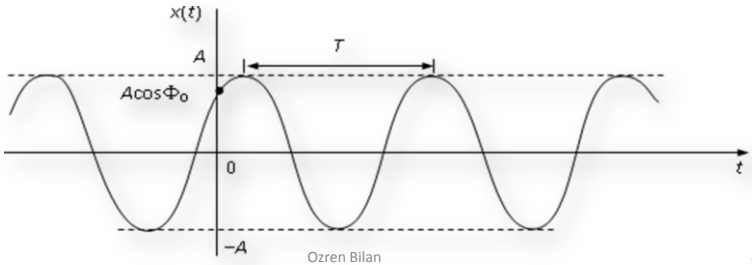
Sinusoida ili sinusni valni oblik je najčešći analogni signal. Lako se generira i posjeduje mnoga dobra svojstva i primjene. Matematički je opisujemo izrazom:

$$x(t) = A\cos(\Omega t + \Phi_o)$$

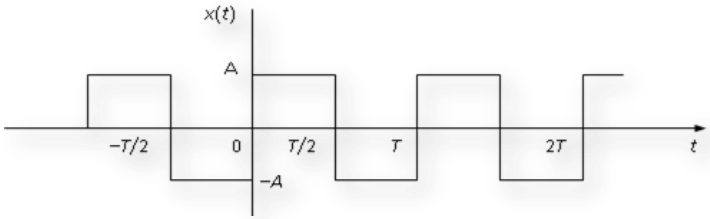
gdje je A, vršna vrijednost, Ω kutna ili kružna frekvencija (radian/s), t vrijeme u (s), Φ_o početni fazni kut (radian), tj. Faza u $t = 0$, $\Omega = 2\pi F$ *gdje je F* frekvencija (Hz),

$$T = 1 / F = 2\pi/\Omega \text{ period (s).}$$

Izraz x(t) sadržava sve potrebne parametre: amplitudu (vršnu, efektivnu i srednju), periodičnost (period, frekvenciju). Drugi valni oblici osim izraza konstantne vrijednosti (istosmjernog napona), nemaju prikazanu kompaktnost,



Simetrični kvadratni signal



Matematički izraz simetričnog kvadratnog valnog oblika signala sastoji se od dva dijela; prvi je za amplitudu, a drugi za periodičnost::

$x(t) = -A, \quad -\frac{T}{2} \le t \le 0$

$+A, \quad 0 \le t \le \frac{T}{2}$

$x(t) = x(t \pm nT), \quad n = 1, 2, 3 \ldots$

Sinusoida i kvadratni valni oblik su **deterministički**. **Slučajne** signale, općenito ne možemo prikazati matematički.

Električni šumovi i smetnje su primjeri **slučajnih signala**.

Ozren Bilan

Jedinični impuls, delta ili Dirac

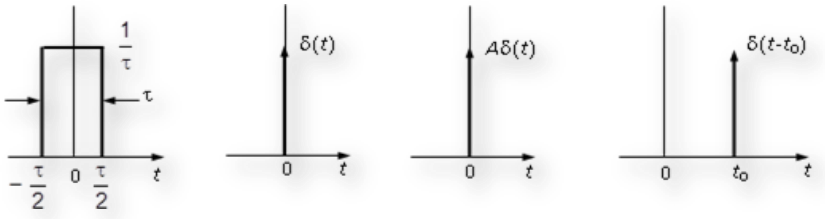
Nastaje iz simetričnog kvadratnog impulsa kada širina teži → 0.
Matematički izraz glasi

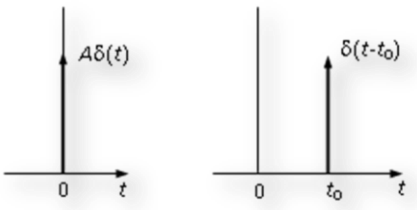
$$\delta(t) = \infty, \quad t = 0$$

$$0, \quad t \neq 0$$

$$\int_{-\infty}^{\infty} \delta(t)dt = 1$$

To znači da je površina (ili intenzitet) jediničnog impulsa 1(jedinica). U skladu s definicijom: **δ(–*t*) = δ(*t*)**





Ako impuls ima intenzitet A, umjesto 1, pišemo $A\delta(t)$ (c).
Ako je impuls translatiran za t_o pišemo $\delta(t - t_o)$ (d), tada je:

$$\delta(t - t_O) = \infty, \quad t = t_O$$

$$0, \quad t \neq t_O$$

$$\int_{-\infty}^{\infty} \delta(t - t_O)dt = \int_{t_1}^{t_2} \delta(t - t_O)dt = 1, \qquad t_1 < t_O < t_2$$

Signal $x(t)$ pomnožen pomnožen translatiranim impulsom ima vrijednost $x(t_o)$ u točki t_o:

$$x(t)\delta(t - t_o) = x(t_o)$$

Jedinična step funkcija

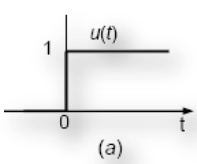
Ako **signal naglo poraste s vrijednosti nula na jediničnu vrijednost** u trenutku t=0, pa ostane nepromijenjen, slično uključivanju prekidača, matematički ga definiramo:

$$u(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

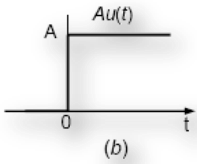
Jedinični impuls i jednični step povezani su relacijom:

$$u(t) = \int_{-\infty}^{t} \delta(t')dt' = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

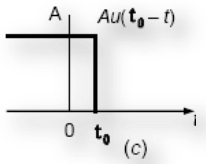
$$\delta(t) = \frac{du(t)}{dt}$$



(a)



(b)

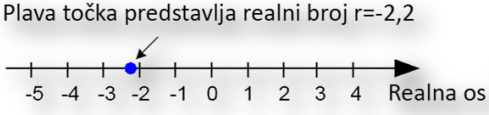


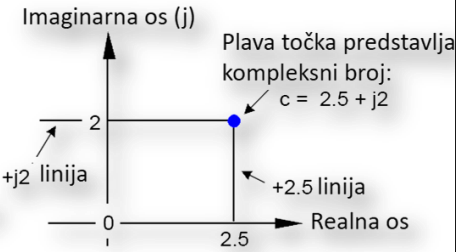
(c)

Kompleksni signali

Bilježenje i smisao kompleksnih brojeva

Realne brojeve koristimo u svakodnevnom životu; napon, temperatura, stanje na računu…Mogu biti pozitivni ili negativni, a predstavljamo ih točkom na realnoj osi.



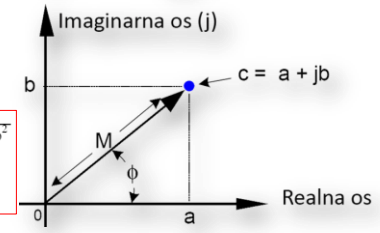


Kompleksni broj isto možemo predstaviti točkom ali u kompleksnoj ravnini. Svaki kompleksni broj ima realni i imaginarni dio.

Umjesto zapisa u pravokutnom ili Kartezijevom obliku možemo ih predstaviti **magnitudom** M i **faznim kutem** Φ.

$$M = |c| = \sqrt{a^2 + b^2}$$

$$\phi = \tan^{-1}\left(\frac{b}{a}\right)$$



U literaturi nailazimo na različite prikaze kompleksnog broja.

Naziv bilježenja	Matematički izraz	Napomena
Pravokutni ili **Kartezijev** oblik	$c=a+jb$	Najlakši za shvaćanje pri objašnjenju
Trigonometrijski oblik	$c=M[\cos(\Phi)+j\sin(\Phi)]$	Koristi se pri digitalnoj obradi signala i komunikacijama
Polarni ili **eksponencijalni** oblik	$c=Me^{j\Phi}$	**Najčešće se koristi** u matematičkim relacijama jer je kompaktan
Oblik magnitude i kuta	$C=M<\Phi$	Skraćeno pisanje prethodnog oblika, ne koristi se u formulama

Treba zapamtiti da su varijable *a, b, M* i *Φ,* predstavljene isključivo **realnim** brojevima. Magnitudu ***c***, nazivamo **modul**, a fazni kut *Φ*, nazivamo **argument**. Određeni su izrazima:

$$M = |c| = \sqrt{a^2 + b^2}$$

$$\phi = \tan^{-1}\left(\frac{b}{a}\right)$$

Eulerova relacija

$c=Me^{j\Phi}=M[\cos(\Phi)+j\sin(\Phi)]$
$c=Me^{-j\Phi}=M[\cos(\Phi)-j\sin(\Phi)]$

Ako gornje jednadžbe podijelimo s M dobivamo:

$$e^{j\Phi}=\cos(\Phi)+j\sin(\Phi)$$
$$e^{-j\Phi}=\cos(\Phi)-j\sin(\Phi)$$

Eulerova relacija je korisna jer pojednostavljuje matematičke izvode i analizu tako da:

- Pretvara trigonometrijske jednadžbe u algebarske
- Matematičke operacije s kompleksnim brojevima izvode se istim pravilima kao i realni brojevi
- Signali se sumiraju vektorskim zbrajanjem
- To je najkompaktniji zapis
- Indikativni su za primjenu DSP i komunikacijskih sustava

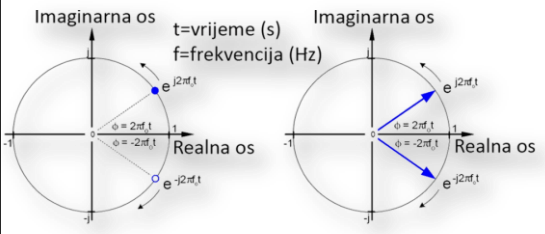
Posvetit ćemo pozornost kompleksnim brojevima koji su vremenske funkcije. Pretpostavimo kompleksni broj jedinične magnitude čiji se fazni kut vremenski raste. To je $e^{j2\Pi f_o t}$.

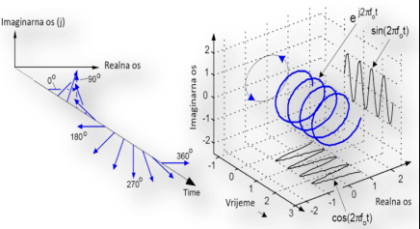
$2\Pi f_o$ označava frekvenciju u rad/s, a odgovara frekvenciji f_o u Hz.

Protekom vremena *t*, vektor broja se okreće oko ishodišta u kompleksnoj ravnini u smjeru obrnutom od kazaljke na satu.

Slika prikazuje broj u nekom trenutku *t*, označen plavom točkom. Ako je frekvencija 3Hz, vektor će se okretati 3 puta u sekundi oko ishodišta.

Možemo zamisliti i drugi broj $e^{-j2\Pi f_o t}$, koji se protekom vremena okreće u smjeru kazaljke na satu jer mu fazni kut postaje sve negativniji (označen bijelom točkom).

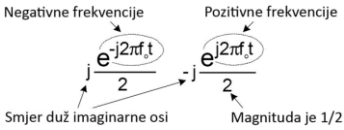




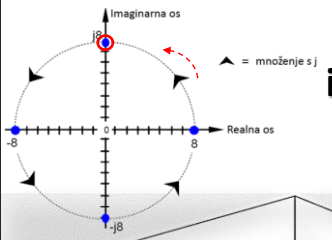
$$\cos(2\pi f_o t) = \frac{e^{j2\pi f_o t}}{2} + \frac{e^{-j2\pi f_o t}}{2} = \frac{e^{j2\pi f_o t} + e^{-j2\pi f_o t}}{2}$$

$$\sin(2\pi f_o t) = \frac{e^{j2\pi f_o t} - e^{-j2\pi f_o t}}{2j} = j\,\frac{e^{-j2\pi f_o t} - e^{j2\pi f_o t}}{2}$$

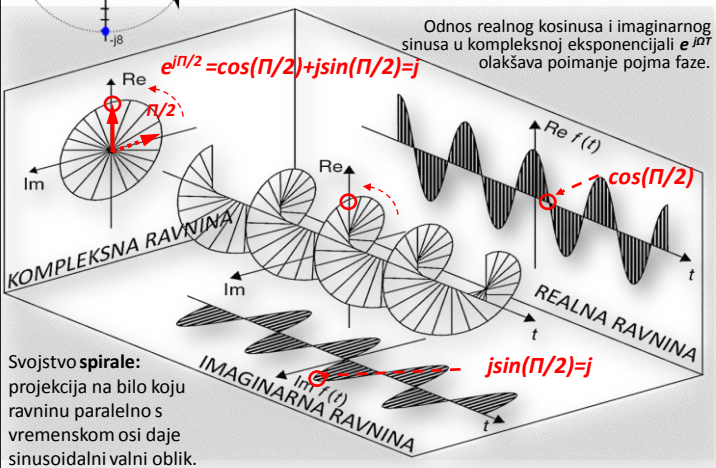
Suma $e^{j2\Pi f_o t}$ i $e^{-j2\Pi f_o t}$ bit će uvijek realni broj. Donji izraz prikazuje pravilo prikaza kompleksnih eksponencijala u frekvencijskom području.



Odnos realnog kosinusa i imaginarnog sinusa u kompleksnoj eksponencijali



Odnos realnog kosinusa i imaginarnog sinusa u kompleksnoj eksponencijali $e^{j\Omega T}$ olakšava poimanje pojma faze.



Svojstvo **spirale:** projekcija na bilo koju ravninu paralelno s vremenskom osi daje sinusoidalni valni oblik.

Uvrstimo li $\Phi=\Pi/2$ u $e^{j\Phi} = \cos(\Phi)+j\sin(\Phi) = e^{j\Pi/2} = \cos(\Pi/2)+j\sin(\Pi/2)=j$

Dakle, množenje s j odgovara rotaciji u lijevo za Π/2

Ozren Bilan

Kompleksni signali

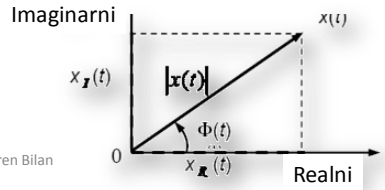
Prirodne fizikalne veličine kao što su signali imaju realne vrijednosti. Međutim, ponekad im se pridružuje imaginarni operator $j = \sqrt{-1}$ zbog matematičkih pogodnosti, jer onda možemo istovremeno prikazati i fazu između napona i struje u nekom sklopu. Primjer kompleksnog signala je izraz: $x(t) = 5\cos\Omega t - j5\sin\Omega t$

Kompleksni signal sastoji se od **realnog i imaginarnog dijela**. U polarnom koordinatnom sustavu možemo ga prikazati pomoću **magnitude i faze**: $x(t) = x_R(t) + jx_I(t) = |x(t)|e^{j\Phi(t)}$

Magnituda ili modul i faza određeni su izrazima:

$$|x(t)| = \sqrt{x_R^2(t) + x_I^2(t)}$$

$$\Phi(t) = \tan^{-1}\frac{x_I(t)}{x_R(t)}$$



Kompleksni eksponencijalni signal

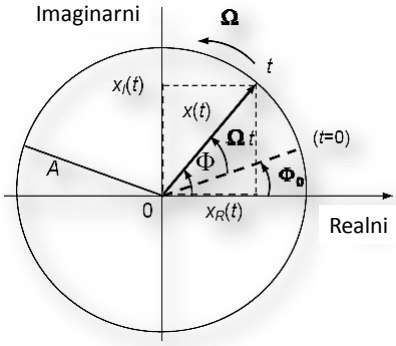
Kompleksna eksponencijala ili kompleksna sinusoida općenito glasi

$$x(t) = Ae^{j(\Omega t + \Phi_0)}$$

Fazor je vektorski prikaz signala. Periodičan je s periodom 2π.

Kompleksnoj eksponencijali, **realni** sinusoidalni dio određuje se na dva načina. Prvi: odredimo realni dio:

$$x_R(t) = Re[A\cos(\Omega t + \Phi_o) + jA\sin(\Omega t + \Phi_o)]$$
$$= A\cos(\Omega t + \Phi_o)$$



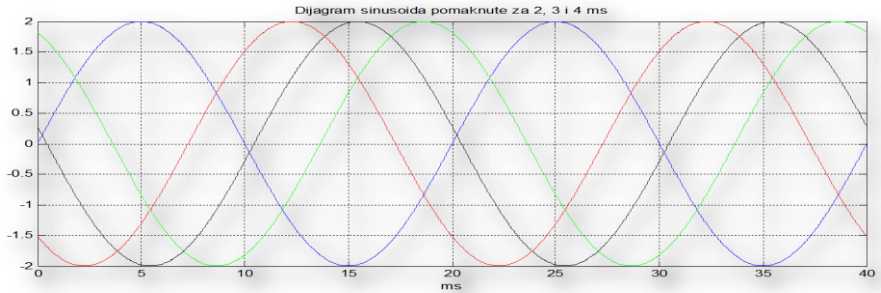
To je projekcija fazora na realnu os. Drugi način je da odredimo **kompleksno konjugiranu vrijednost**, pa vektorski sumiramo i podijelimo s 2:

$$x_R(t) = \frac{1}{2}\left[x(t) + x^*(t)\right]$$
$$= \frac{1}{2}\left[Ae^{j(\Omega_0 t + \Phi_0)} + Ae^{-j(\Omega_0 t + \Phi_0)}\right]$$

Osnovni signali u Matlabu

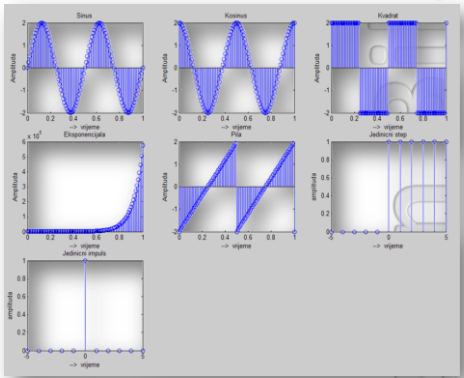
PRIKAZ FAZNO POMAKNUTIH SINUSOIDA

```
%Analogne signale u MATLAB-u crtamo
% tako da određujemo vrlo mali inkrement vremena
t=0:.04/1000:.04;
% Vrlo mali inkrement vremena asproksimi9ra analogni signal
f0=50;
%Određujemo četiri sinusoide bez faznog pomaka i pomaknute
za 2, 3 i 4 ms
s1=2*sin((2*pi*(f0)*t));
s2=2*sin((2*pi*(f0)*t)+2);
s3=2*sin((2*pi*(f0)*t)+3);
s4=2*sin((2*pi*(f0)*t)+4);
%crtamo prvu sinusoidu plavom bojom i zadržavamo sliku
plot(t*1000,s1);
hold on
%crtamo drugu sinusoidu zelenom bojom i zadržavamo sliku
plot(t*1000,s2,'g');
hold on
%crtamo treću sinusoidu crnom bojom i zadržavamo sliku
plot(t*1000,s3,'k');
hold on
%crtamo četvrtu sinusoidu crvenom bojom i zadržavamo sliku
plot(t*1000,s4,'r');
title('Dijagram sinusoida pomaknute za 2, 3 i 4 ms');
xlabel('ms');
grid;
hold off
```



Osnovni signali u Matlabu

```
% sinus
t=0:0.01:1; a=2;
b=a*sin(2*pi*2*t);
subplot(3,3,1); stem(t,b);
xlabel('--> vrijeme');
ylabel('Amplituda');
title ('Sinus');
% kosinus
t=0:0.01:1;
a=2;
b=a*cos(2*pi*2*t);
subplot(3,3,2);
stem(t,b);
xlabel('--> vrijeme');
ylabel('Amplituda');
title ('Kosinus');
% kvadrat
t=0:0.01:1;
a=2;
b=a*square(2*pi*2*t);
subplot(3,3,3);
stem(t,b);
xlabel('--> vrijeme');
ylabel('Amplituda');
title ('Kvadrat');
% eksponencijala
t=0:0.01:1;
a=2;
b=a*exp(2*pi*2*t); subplot(3,3,4);
stem(t,b);
xlabel('--> vrijeme');
ylabel('Amplituda');
title ('Eksponencijala');
% pila
t=0:0.01:1;
a=2;
b=a*sawtooth(2*pi*2*t);
subplot(3,3,5);
stem(t,b);
xlabel('--> vrijeme');
ylabel('Amplituda');
title ('Pila');
% jedinicni step
n=-5:5;
a = [zeros(1,5),ones(1,6)];
subplot(3,3,6);
stem(n,a);
Xlabel ('--> vrijeme');
Ylabel ('amplituda');
title('Jedinicni step');
% jedinicni impuls
n=-5:5;
a = [zeros(1,5),ones(1,1),zeros(1,5)];
subplot(3,3,7);
stem(n,a);
Xlabel ('--> vrijeme');
Ylabel ('amplituda');
title('Jedinicni impuls');
```



ŠUM

Svi **neželjeni signali slučajne prirode** koji se superponiraju korisnom signalu koji prenosi informaciju nazivaju se **šum**. U elektroničkim uređajima i sklopovima šum se generira uslijed slučajnog gibanja elektrona, nejednolike brzine, sudara... Nazivamo ga **termički šum**. Aktivna elektronika generira i tzv. **šum sačme**. Neke pojave kao što su grmljavina i preklapanje prekidača uzrok je **impulsnog šuma**. Procesi na suncu generiraju termičke i impulsne šumove. Šum može biti interni ili sustavni ili vanjski šum kojeg još nazivamo **interferencija** ili **smetnja**.

Poseban šum predstavlja interferenciju energetskih napojnih vodova mreže 50 Hz. Taj šum djeluje na **biološke organizme** i **elektroničke sklopove** elektromagnetskim valovima. Sklopovi napajanja unutar uređaja također unose interferencije.

Prema frekvencijskoj karakteristici šuma razlikujemo **bijeli i roza šum**.

Bijeli šum ima **konstantnu gustoću spektra snage** S(F). Prolazom bijelog šuma kroz filter promijenit će mu se karakteristike.

Funkcija gustoće vjerojatnosti *PDF* i funkcija kumulativne distribucije *CDF*

Šum se modelira kao **slučajna varijabla** x. Vjerojatnost pojave šuma pri različitim amplitudama je **funkcija gustoće vjerojatnosti *PDF*** ili **vjerojatnost**, **p(x)**. **Funkcija distribucije vjerojatnosti ili funkcija kumulativne distribucije *CDF***, P(x) određena je izrazom:

$$\mathbf{P(x) = \int_{-\infty}^{x} p(x)dx}$$

Osnovna karakteristika slučajne varijable je **srednja vrijednost**, m ili μ i **varijanca** σ^2. Određene su izrazima:

$$\mathbf{m = E[x] = \int_{-\infty}^{\infty} xp(x)\,dx}$$

$$\mathbf{\sigma^2 = E[(x-m)^2] = \int_{-\infty}^{\infty} (x-m)^2 p(x)dx}$$

Gdje je **E**, **očekivanje ili očekivana vrijednost**.

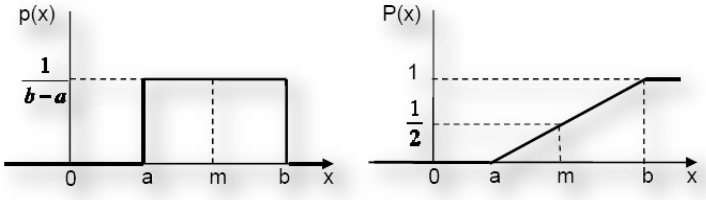
Standardna devijacija σ i jednolika distribucija

Kvadratni korijen varijance je **standardna devijacija**, označavamo je σ.

Jednolikoj distribuciji, vrijednost slučajne varijable x unutar područja pojavljivanja, jednaka je:

$$p(x)=\frac{1}{b-a}, \quad a \le x \le b$$

Funkcije *PDF* gustoće vjerojatnosti i *CDF* distribucije vjerojatnosti ili kumulativne distribucije prikazane su na slici.



Pri tome su srednja vrijednost i varijanca: $m=\frac{a+b}{2}$

$$\sigma^2=\frac{(b-a)^2}{12}$$

Gaussov šum

Mnoge slučajne realne varijable imaju *Gaussovu* ili normalnu distribuciju.

Slučajni električni šum okoline i termički šum komponenti sumira se sa korisnim naponima analognih signala.

Takav šum najčešće ima ***normalnu*** ili ***Gaussovu statističku distribuciju***.

U MATLAB-u, Gausov šum simulira se generatorom slučajnih brojeva ***randn***.

Ozren Bilan

Gaussova ili normalna distribucija

Mnoge slučajne realne varijable imaju *Gaussovu* distribuciju ili normalnu distribuciju. *PDF* i *CDF* Gaussovog bijelog šuma su:

Funkcija gustoće vjerojatnosti PDF i funkcija kumulativne distribucije CDF

$$p(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-x^2/2\sigma^2}$$

$$F(x)=\int_{-\infty}^{x}p(x)dx$$

Gdje je σ^2 varijanca, a σ standardna devijacija.

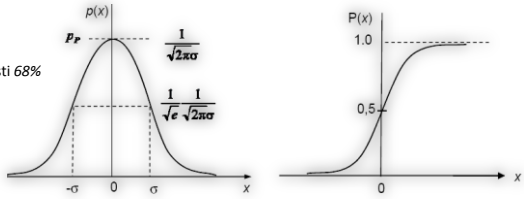
Distribucija ima oblik zvona. **Vršna vjerojatnost** u x=0 je:

$$p_P=\frac{1}{\sqrt{2\pi}\sigma}e^{-0/2\sigma^2}=\frac{1}{\sqrt{2\pi}\sigma}$$

Na udaljenosti x=+/- σ vjerojatnost je:

Što je **σ** manja zvono je uže. $p_\sigma=\frac{1}{\sqrt{2\pi}\sigma}e^{-\sigma^2/2\sigma^2}=\frac{1}{\sqrt{e}}\cdot\frac{1}{\sqrt{2\pi}\sigma}\approx 0{,}606\frac{1}{\sqrt{2\pi}\sigma}$

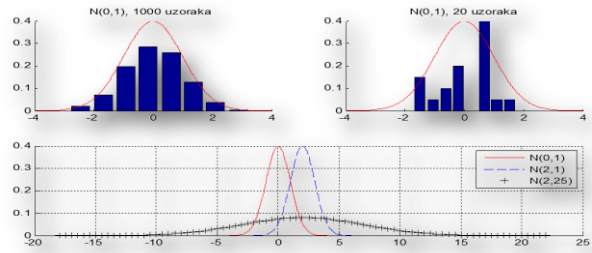
±1σ srednje vrijednosti *68%*



U MATLAB-u ćemo naredbom ***rand*** generirati 1000 normalno distribuiranih slučajnih uzoraka **nulte srednje vrijednosti** sa **jediničnom standardnom devijacijom** (varijancom). Na istom grafu nacrtat ćemo histogram i teoretsku vrijednost PDF. Ponovit ćemo isto ali ćemo promijeniti srednju vrijednost na 2, a zatim ćemo promijeniti standardnu devijaciju na 5. Konačno, smanjit ćemo broj uzoraka na 20.

Standardna devijacija

Standardna devijacija upravlja širinom *zvona*. Uzorci normalno distribuiranog signala bit će unutar *±1σ* srednje vrijednosti *68%* vremena, a unutar *±2σ 95%* vremena i unutar *±3σ 99.75%* vremena. Vjerojatnost da će se pojaviti signal amplitude veće od 10 standardnih devijacija srednje vrijednosti je tako mala da se pojavljuje u *1μs* svakih *10 milijardi* godina!

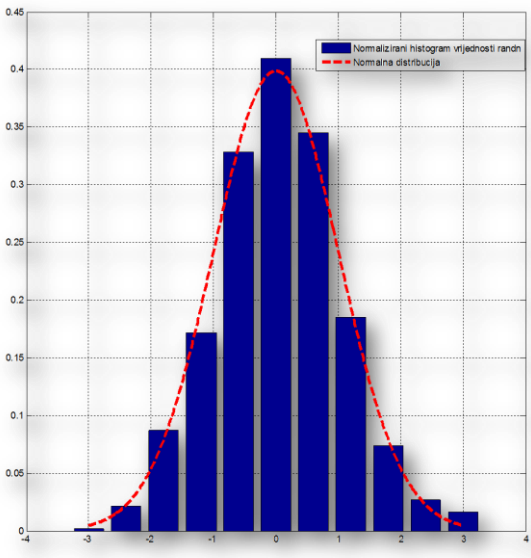


Histogram s 1000 uzoraka pravilno je u obliku zvona i blizak teoretskoj vrijednosti PDF, što nije slučaj sa 20 uzoraka. Kvaliteta procjene statističkih svojstava slučajnih procesa neposredno je povezana s brojem uzoraka. Promjenom srednje vrijednosti 0 na 2, početna krivulja (puna linija) pomiče se na desno (crtkana linija), a oblik joj ostaje isti. Promjena standardne devijacije uzrokuje nižu i širu krivulju distribucije. Površina ispod PDF krivulje uvijek je 1.

```
Ntotal=1000; %ukupni broj uzoraka
mu=0; sigma=1; %srednja vrijednost i std. devijacija
ech = randn(1,Ntotal) .* sigma + mu;
Nclas=8; dx=.01;
[N,X]=hist(ech,Nclas);
x=mu-4*sigma:dx:mu+4*sigma;
y=exp(-0.5*((x-mu)/sigma).^2)/(sqrt(2*pi)* sigma);
figure; clf; zoom on;
subplot(221); hold on;
bar(X,N/Ntotal); plot(x,y,'r');
title('N(0,1), 1000 uzoraka ');
subplot(212);
hold on;plot(x,y,'r')
%%mijenjamo srednju vrijednost mu
mu=2;sigma=1;x=mu-4*sigma:dx:mu+4*sigma;
y=exp(-0.5*((x-mu)/sigma).^2)/(sqrt(2*pi)* sigma);
subplot(212);
hold on;plot(x,y,'b--')
% mijenjamo standardnu devijaciju
mu=2;sigma=5;
x=mu-4*sigma:dx:mu+4*sigma;
y=exp(-0.5*((x-
mu)/sigma).^2)/(sqrt(2*pi)*sigma);subplot(212);
hold on;plot(x(1:50:end),y(1:50:end),'k+');grid;
legend('N(0,1)','N(2,1)', 'N(2,25)')
%mijenjamo broj uzoraka sa 1000 na 20
Ntotal=20;mu=0;sigma=1;
ech=randn(1,Ntotal)*sigma+mu;
Nclas=8;[N,X]=hist(ech,Nclas);
x=mu-4*sigma:dx:mu+4*sigma;
y=exp(-0.5*((x-mu)/sigma).^2)/(sqrt(2*pi)* sigma);
subplot(222); hold on;
bar(X,N/Ntotal);
plot(x,y,'r');
title('N(0,1), 20 uzoraka')
```

Normalna distribucija *randn*

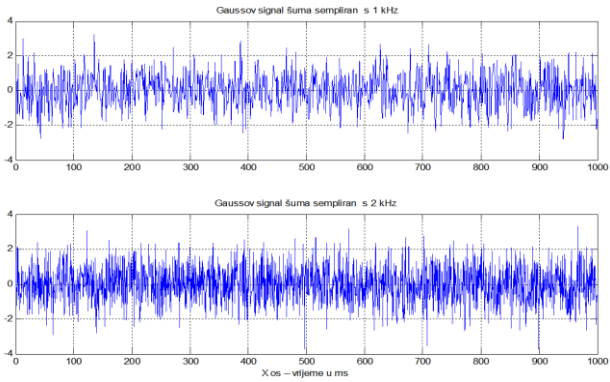
```
% Stvori randn vektor s 1000 vrijednosti:
N=1:1000;
rn=randn(size(N));
% Izračunaj histogram randn vrijednosti u
    11 grupa od -3 do 3
xr=-3:6/10:3;
M=hist(rn,xr);
% Nacrtaj normalizirani histogram
    jedinične površine vrijednosti randn
bar(xr,M/(.6*sum(M)))
% Izračunaj normalnu pdf nulte srednje
    vrijednosti i jedinične varijance od -3
    do 3
xn=-3:.01:3;
y=pdf('normal',xn,0,1);
% Nacrtaju pdf na histogramu
hold
plot(xn,y,'r--','LineWidth',3);
grid;
hold off
legend('Normalizirani histogram
    vrijednosti randn','Normalna
    distribucija')
```



Gaussov signal šuma

```
N1=1:1000;
N2=1:2000;
t1=N1;
% Određujemo dužinu svakog signala na 1000 ms
t2=N2/2;
n1=randn(size(N1));
% n1 je 1000 uzoraka, 1 ms po uzorku
n2=randn(size(N2));
% n2 je 2000 uzoraka, 0.5 ms po uzorku
subplot(2,1,1),
plot(t1,n1),
title(' Gaussov signal šuma sempliran  s 1 kHz')
axis([0 1000 -4 4])
subplot(2,1,2),
plot(t2,n2),
title('Gaussov signal šuma sempliran  s 2 kHz')
axis([0 1000 -4 4])
xlabel('X os – vrijeme u ms')
```

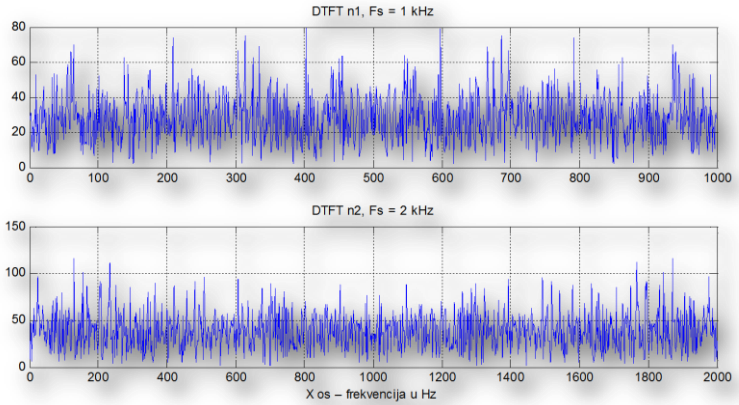
Amplituda **randn** signala šuma ima vrlo veliku vjerojatnost pojavljivanja između -3 i 3 jer joj je srednja vrijednost jednaka 0, a standardna devijacija jednaka 1.



Gaussov šum u frekvencijskom području

```
[X1,f1]=dtft_demof(n1,0,1000,1024,1000);
[X2,f2]=dtft_demof(n2,0,2000,1024,2000);
subplot(2,1,1),plot(f1,abs(X1)),title('DTFT n1, Fs = 1 kHz')
subplot(2,1,2),plot(f2,abs(X2)),title('DTFT n2, Fs = 2 kHz')
xlabel('X os – frekvencija u Hz')
```

Za obje frekvencije sempliranja snaga šuma je **jednoliko raspodijeljena** u frekvencijskom području



Gustoća snage šuma i frekvencija sempliranja

$$P = \frac{1}{2\pi N}\int_{-\pi}^{\pi} |X_N(\Omega)|^2 \, d\Omega \qquad P = \int_{-f_s/2}^{f_s/2} \frac{|X(f)|^2}{N f_s} \, df$$

$$d\Omega = \frac{2\pi}{f_s} \, df$$

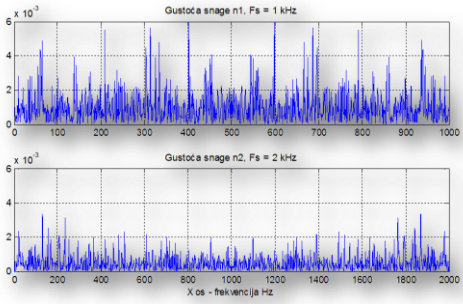
$$f = f_s/2 \quad \text{za} \quad \Omega = \pi$$

$$P = \left(\frac{1}{2\pi N}\right)\left(\frac{2\pi}{f_s}\right)\int_{-f_s/2}^{f_s/2} |X_N(f)|^2 \, df$$

$$\boxed{p(f) = \frac{|X(f)|^2}{N f_s}}$$

Gustoća snage šuma u frekvencijskom području **obrnuto je proporcionalna** frekvenciji sempliranja

```
p1=(abs(X1)).^2/(1000*length
    (n1));
% gustoća snage signala n1
p2=(abs(X2)).^2/(2000*length
    (n2));
% gustoća snage signala n2
subplot(2,1,1),plot(f1,p1),
grid;title('Gustoća snage n1,
    Fs = 1 kHz')
axis([0 1000 0 6e-3])
subplot(2,1,2),plot(f2,p2),
grid;title('Gustoća snage n2,
    Fs = 2 kHz')
axis([0 2000 0 6e-3])
xlabel('X os - frekvencija Hz')
>> mean(p2)
% srednja vrijednost gustoće
    snage signala n2
>> mean(p1)
% srednja vrijednost gustoće
    snage signala n1

ans =
  5.0565e-004
ans =
  8.9208e-004
```



GENERIRANJE SLUČAJNIH BROJEVA

Slijedeća naredba generira 1 x 1000 matricu 1000 slučajnih brojeva između 0 i 1.:

```
rand(1,1000);
```

Generiranje brojeva između a i b:

Slijedeći kod generira cijele brojeve između a=1 i b=100.

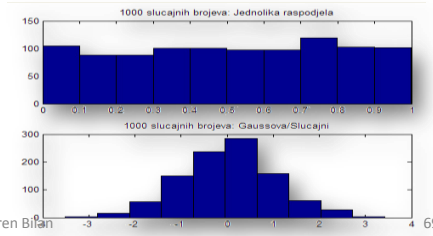
```
a = 1;
b = 100;
x = round(a + (b-a) * rand);
```

a i b se mogu podešavati po potrebi.

Za normalne *Gaussove* slučajne brojeve, možemo koristiti *randn* funkciju umjesto *rand*.

Matlab kod za donju sliku.

```
% Generiranje slučajnih brojeva u Matlab-u
clear all;
close all;
% Generiranje 1000 slucajnih brojeva s rand i randn
% rand funkcija
% kreira jednoliku raspodjelu
% randn funkcija
% generira normalnu Gaussovu raspodjelu
x = rand(1,1000); %generira 1 x 1000 matricu
figure('Color',[1 1 1]);
subplot(2,1,1);
hist(x);
title('1000 slucajnih brojeva: Jednolika raspodjela');
x = randn(1,1000); %generira 1 x 1000 matricu
subplot(2,1,2);
hist(x);
title('1000 slucajnih brojeva: Gaussova/Slucajni');
% Generiranje slucajnih brojeva izmedju 1 i 100
% generiranje slucajnih brojeva izmedu
% 1 (a) i 100 (b). Funkcija round
% zaokruzuje vrijednost na najblizi cijeli broj.
a = 1;
b = 100;
x = round(a + (b-a) * rand)
```



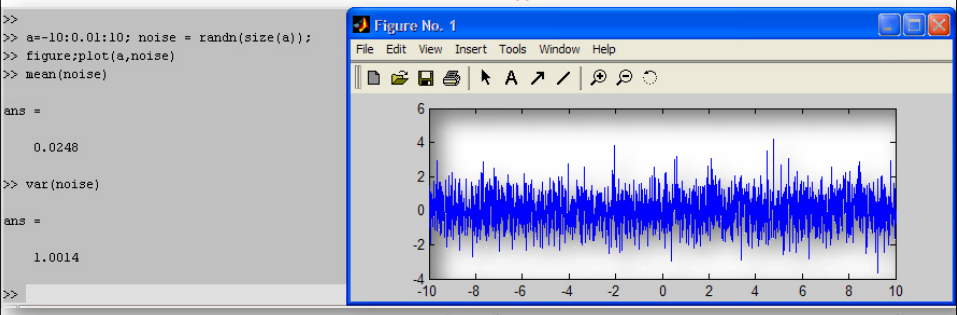
Generiranje šuma u Matlabu

nulte srednje vrijednosti s jediničnom varijancom

Generiraj bijeli gaussov šum od -10<a<10

Nulte srednje vrijednosti i varijancom 1.

```
>> a=-10:0.01:10; noise = randn(size(a));
>> figure;plot(a,noise)
>> mean(noise)
ans =
   0.0248
>> var(noise)
ans =
   1.0014
```



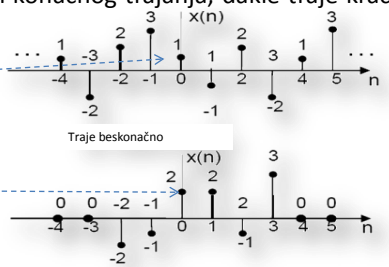
DISKRETNI VREMENSKI SIGNALI

Uzorci vremenski ustaljenog (analognog) signala tvore diskretan (digitalni) vremenski signal. Te uzorke treba **kvantizirati i binarno kodirati** kako bi postali digitalni signal. Kvantizaciju i kodiranje najčešće podrazumjevamo, pa kad govorimo **vremenski diskretan ili digitalan** najčešće mislimo na isti oblik. Slika pokazuje primjer. Amplitude uzoraka $x(n)$ mogu predstavljati bilo što: pozitivne ili negativne, nulu ili beskonačnost, cjelobrojne ili decimalne vrijednosti, realne ili kompleksne vrijednosti (obično podrazumijevamo realne). Signal može trajati beskonačno, dakle, postoji stalno ili može biti konačnog trajanja, dakle traje kraće vrijeme. Najčešće ga uzimamo oko nule.

Signale na slikama možemo zapisati:

$x(n) = [\ldots 1, -2, 2, 3, \mathbf{1}, -1, 2, -2, 1, 3 \ldots]$

$x(n) = [-2, -1, \mathbf{2}, 2, -1, 3]$



Traje beskonačno

U ovom obliku digitalne signale nazivamo **sekvence** (ili diskretne sekvence). Pri tome **uzorke u izvorištu** označavamo debljim znakom, podcrtano ili sa strelicom.

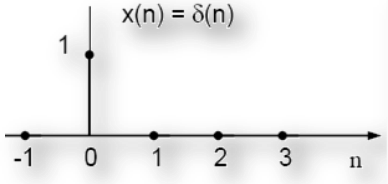
Osnovni vremenski diskretni signali
Jedinični uzorak

U principu, vremenski diskretni signali su općenito uzorci pripadajućih vremenski kontinuiranih signala.

Međutim, postoje i osnovni vremenski diskretni signali, slični vremenski kontinuiranim slučajevima. **Jedinični uzorak**, koji se naziva i **jedinični impuls** je signal koji ima amplitudu jednaku 1 u izvorištu, a inače mu je vrijednost jednaka 0:

$\delta(n) = 1 , n = 0$

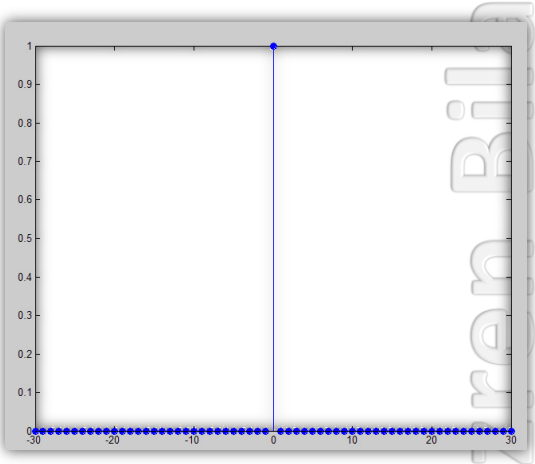
$0 , n \neq 0$



Potrebno je uočiti kako vremenski diskretni signal **nije samplirana verzija analognog jediničnog impulsa** ali isto vrijedi da je $\delta(n) = \delta(-n)$.

Jedinični uzorak u Matlabu

```
>> n=-30:30;
% odredi index n
>>delta=(n==0);
% odredi delta sekvencu
>>stem(n,delta, 'filled')
% nacrtaj delta sekvencu
```



Ozren Bilan

Pomak, refleksija i skaliranje jediničnog uzorka

Zadanu sekvencu *x(n)*, sastavljenu od skaliranih jediničnih uzoraka možemo promijeniti na tri načina:

- **Vremenskim pomakom** za k koraka ili uzoraka, gdje je k cijeli broj. To bilježimo *x(n-k)*. Sekvenca *x(n-k)* označava sekvencu *x(n)* pomaknutu za *k* koraka ili uzoraka na desno ako je *k>0*, a sekvenca *x(n+k)* lijevo ako je *k<0*.
- **Savijanje, refleksija ili zrcaljenje** s obzirom na okomitu os, označava **promjenu toka vremena** što označavamo *x(-n)*. Signal *x(-n)* odgovara refleksiji signala *x(n)* oko osi u izvorištu *n=0*. Pri tome se mijenja predznak *n* u *-n*.
- Treća operacija je skaliranje

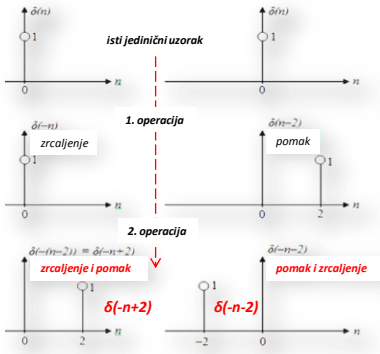
Za kontinuirane signale i za diskretne signale vrijedi **da operacije pomaka i savijanja nisu komutativne**.
Ako sekvencu prvo pomaknemo pa savijemo nećemo dobiti isti rezultat kao kada je prvo savijemo pa pomaknemo.

Primjer s jediničnim uzorkom

1) Zadani jedinični uzorak *δ(n)*, prvo zrcalimo pa kasnimo za dva koraka.
2) Nakon toga isti početni uzorak *δ(n)*, prvo kasnimo pa zrcalimo.

Rezultat operacije **zrcali pa pomakni** je prikazan s lijeve strane, a rezultat operacije **pomakni pa zrcali** s desne. Pri tome zrcaljenje podrazumijeva promjenu predznaka n, a pomak zamjenu (n) sa (n-2). Rezultat u prvom slučaju je:

(1) $\delta(n) \rightarrow \delta(-n)$
(2) $\delta(-n) = \delta(-(n)) \rightarrow \delta(-(n-2)) = \boldsymbol{\delta(-n+2)}$



U drugom slučaju, prikazanom s desna prvo vršimo pomak zamjenom *(n)* s *(n-2);* zatim zrcalimo promjenom predznaka *n*. Rezultat je:
(1) $\delta(n) = \delta((n)) \rightarrow \delta((n-2))$
(2) $\delta((n-2)) = \delta(n-2) \rightarrow \boldsymbol{\delta(-n-2)}$
Konačni rezultati su različiti što dokazuje sa operacije pomaka i zrcaljenja nisu komutativne:

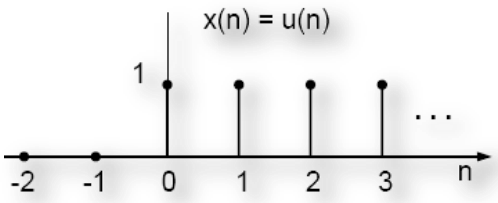
zrcali (pomakni (δ(n))) ≠ pomakni (zrcali (δ(n)))

Jedinični step

Definiran je:

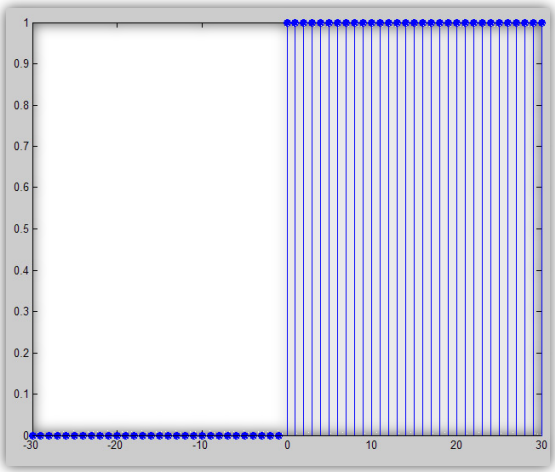
$$u(n) = 1 \text{ , } n \geq 0$$

$$0 \text{ , } n < 0 \text{ (ili } n \leq -1)$$



Jedinični step u Matlabu

```
>>n=-30:30; % odredi indeks n
>>u_step=(n>=0); % odredi sekvencu jediničnog stepa
>>stem(n, u_step, 'filled') % nacrtaj sekvencu

>> n=-30:30
n =
  Columns 1 through 6
   -30   -29   -28   -27   -26   -25
  Columns 7 through 12
   -24   -23   -22   -21   -20   -19
  Columns 13 through 18
   -18   -17   -16   -15   -14   -13
  Columns 19 through 24
   -12   -11   -10    -9    -8    -7
  Columns 25 through 30
    -6    -5    -4    -3    -2    -1
  Columns 31 through 36
     0     1     2     3     4     5
  Columns 37 through 42
     6     7     8     9    10    11
  Columns 43 through 48
    12    13    14    15    16    17
  Columns 49 through 54
    18    19    20    21    22    23
  Columns 55 through 60
    24    25    26    27    28    29
  Column 61
    30
>> u_step=(n>=0);
>> stem(n, u_step, 'filled')
>>
```

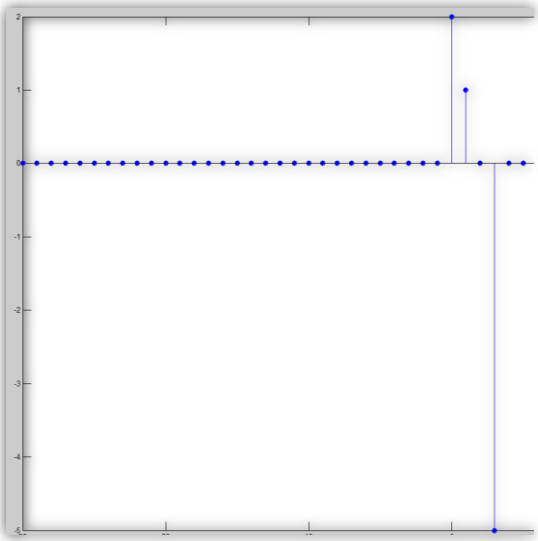


Sekvenca u Matlabu

$$x[n] = 2\delta[n] + 3\delta[n-1] - 5\delta[n-3]$$

```
>> n=-30:30;
>> xn=2*(n==0)+((n-1)==0)-5*((n-3)==0);
>> stem(n, xn, 'filled')
>>
```

```
>>n=-30:30;
% odredi index n
>>xn=2*(n= =0)+((n-1)= =0)-5*((n-3)= =0);
% definiraj sekvencu x[n]
>>stem(n, xn, 'filled')
% nacrtaj sekvencu x[n]
```

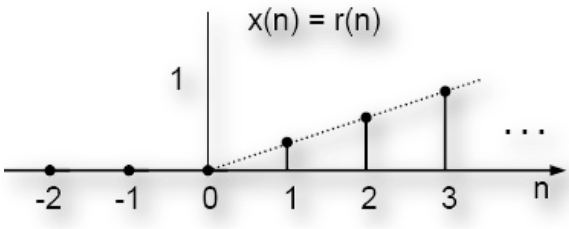


Jedinični ramp signal

To je divergentni signal kojemu amplituda teži ∞ kad n teži ∞), definiran kao:

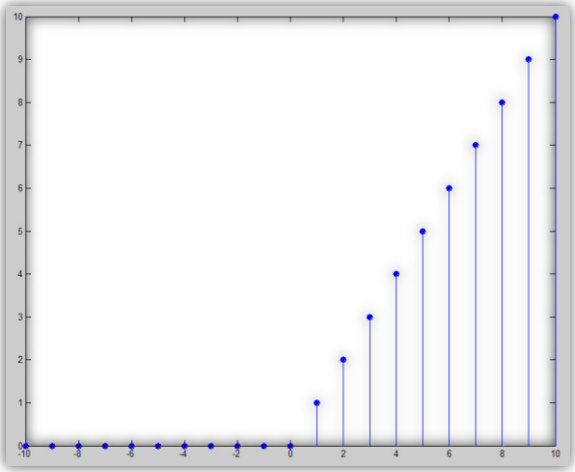
$$r(n) = n \text{ , } n \geq 0$$

$$0 \text{ , } n < 0 \text{ (ili } n <= -1)$$



Jedinični ramp signal u Matlabu

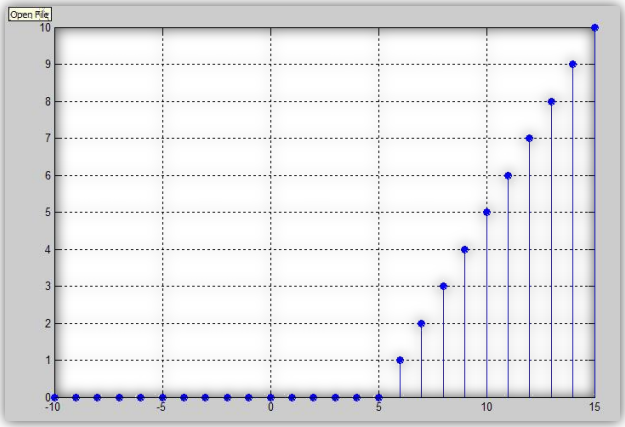
```
>>n=-10:10;
% odredi index n
>>ramp=n.*(n>=0);
% odredi ramp signal
>>stem(n,ramp, 'filled')
% nacrtaj ramp

>> n=-10:10;
>> ramp=n.*(n>=0);
>> stem(n,ramp, 'filled')
>>
```



Pomaknuti ramp signal u Matlabu

Nacrtaj: r[n-5]

```
>>n=-10:15;
% odredi index n
>>x=(n-5).*((n-5)>=0);
% odredi translatiranu
% verziju rampa
>>stem(n,x, 'filled');grid
% nacrtaj translatiranu
% verziju rampa
```

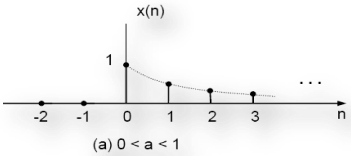


Realna eksponencijala

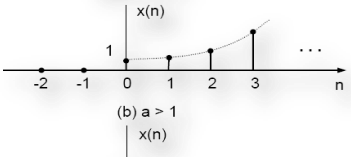
Realna eksponencijalna funkcija definirana je:

$$x(n) = a^n,\ n \geq 0$$
$$0,\ n < 0$$

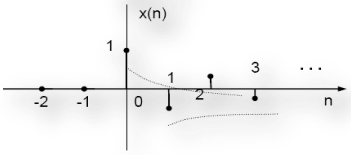
Gdje je a realna konstanta. Postoje četiri različita slučaja; dva divergiraju, a dva konvergiraju.



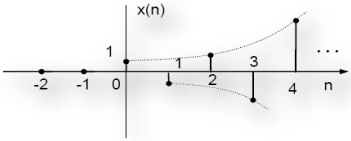
(a) 0 < a < 1



(b) a > 1



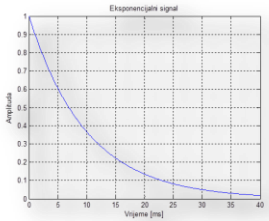
(c) -1 < a < 0

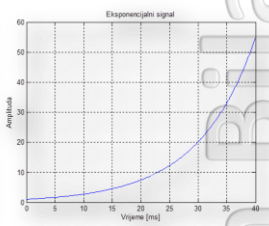


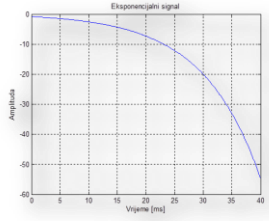
(d) a < -1

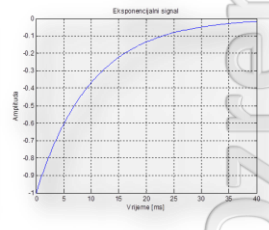
Realna eksponencijala u Matlabu

```
% generiraj eksponencijalni signal x(t)=exp(-0.1t)
t=0:0.1:40;
x=exp(-0.1*t);
plot(t,x);
grid;
title('Eksponencijalni signal');
xlabel('Vrijeme [ms]');
ylabel('Amplituda');

% generiraj eksponencijalni signal x(t)=exp(0.1t)
t=0:0.1:40;
x=exp(0.1*t);
plot(t,x);
grid;
title('Eksponencijalni signal');
xlabel('Vrijeme [ms]');
ylabel('Amplituda');

% generiraj eksponencijalni signal x(t)=-exp(0.1t)
t=0:0.1:40;
x=-exp(0.1*t);
plot(t,x);
grid;
title('Eksponencijalni signal');
xlabel('Vrijeme [ms]');
ylabel('Amplituda');

% generiraj eksponencijalni signal x(t)=-exp(-0.1t)
t=0:0.1:40;
x=-exp(-0.1*t);
plot(t,x);
grid;
title('Eksponencijalni signal');
xlabel('Vrijeme [ms]');
ylabel('Amplituda');
```









Ozren Bilan

Sinusoida, digitalna frekvencija, periodičnost, kompleksna eksponencijala

Kosinusoidalni signal sampliran periodom *T glasi* :

$$x(t) = A\cos(\Omega t + \Phi_0) = A\cos(\Omega nT + \Phi_0)$$

A je amplitude, Ω = 2π*F* je kutna frekvencija (rad/s), *F* frekvencija (Hz), Φo početna faza (fazni kut) (rad),

$T = 1/F = 2\pi/\Omega$ period (s).

Pišemo sličnu jednadžbu za vremenski diskretnu (digitalnu) kosinusoidu:

$$x(n) = A\cos(\omega n + \Phi_0)$$

Gdje je *A* amplituda, *n* indeks (vremenski) , a **veličinu ω ćemo objasniti**.

Digitalna i kružna digitalna frekvencija

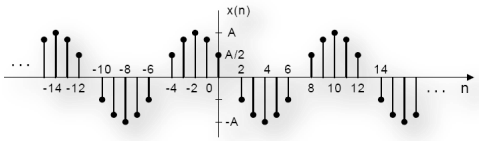
Na slici je prikazana sekvenca koja je počela u prošlosti i proteže se u budućnost. Sadašnjost je **uzorak u n=0** i u tom trenutku uzorci se mijenjaju kako je prikazano dijagramom sekvence. Sekvenca predstavlja neki digitalni protok ili struju uzoraka. Kažemo da ima frekvenciju sempliranja *fs*, ako u jednoj sekundi prođe ***fs*** uzoraka.

Promjena frekvencije sempliranja digitalnog protoka uzoraka ***fs*** za vrijeme reprodukcije zvučnog signala promijenit će percipiranu prirodnu frekvenciju protoka. Audio CD ima frekvenciju sempliranja 44100 uzoraka u sekundi, a zvučni signali visoke rezolucije mogu imati 88,2 kHz, 96 kHz, 192 kHz ili 385 kHz.

Linearna digitalna frekvencija **f** jednaka je ***F/fs*** za neku kontinuiranu prirodnu frekvenciju ***F***. Budući da *pojam vremena* praktički nema smisla u cjelovitom digitalnom diskretnom sustavu, apscisu ne označavamo kao vrijeme nego koristimo pojam uzorka. **f** onda predstavlja digitalnu analogiju kontinuirane prirodne frekvencije ***F***.

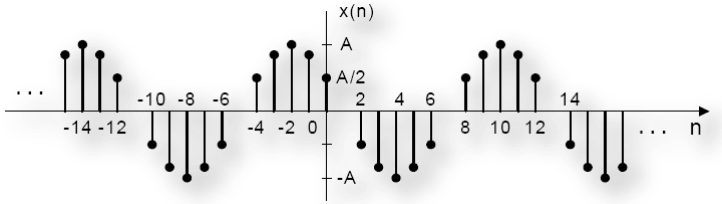
Digitalnu frekvenciju možemo izraziti i kao kružnu digitalnu frekvenciju u radijanima po uzorku. Kako bi pretvorili iz kontinuirane prirodne frekvencije ***F,*** u kružnu digitalnu frekvenciju **ω** primjenjujemo jednadžbu

$$\omega = 2\pi f = 2\pi\, F/fs$$



Signal na slici možemo napisati u obliku: $x(n) = A\cos(\frac{\pi}{6}n + \frac{\pi}{3})$

Nacrtan je na slici s očitim sinusoidalnim valnim oblikom i periodičnosti.



Usporedimo li jednadžbe slijedi: $\omega = \Omega T = \frac{\Omega}{f_s}$

Jedinica za **ω** je (rad/s) (s)=rad, ali najčešće se interpretira kao **radijana/uzorku**. **ω** nazivamo **kružna digitalna frekvencija**.

Također možemo definirati **ω=2π*f*** *s* f linearnom **digitalnom frekvencijom** (perioda/uzorku). Digitalna sinusoida puni period kada je *n*ω = 2π *radijana* ili ω= 2π/n *radijana/uzorku*.

Kada su uzorci jednoliko raspoređeni na kružnici čije je centar ishodište, **ω** možemo shvatiti kao **kut između dva uzastopna uzorka**.

Zato jer je Ω = 2π*F* , a *T* = 1/ *fs* onda je $\omega = 2\pi\frac{F}{f_s}$ ili $f = \frac{F}{f_s}$

Treba zapamtiti:

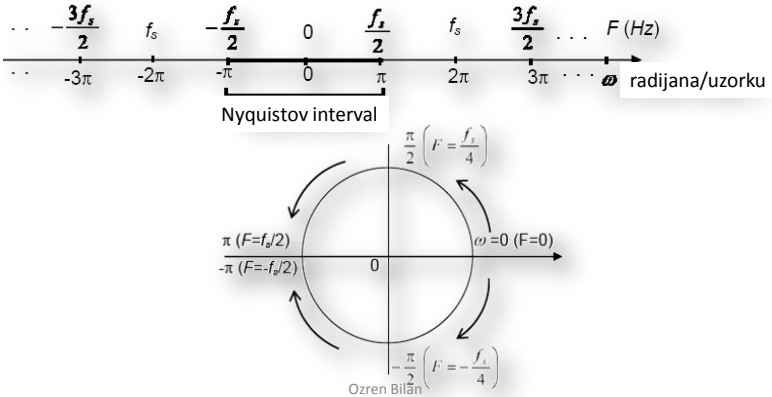
F *je* analogna frekvencija u Hz, ***fs*** je frekvencija sampliranja u *Hz*, ***f*** je digitalna frekvencija u *Hz*.

Digitalna frekvencija **ω** i ***f*** ovise i o analognoj frekvenciji ***F*** i frekvenciji sampliranja ***fs*.**

Istovremeno su **ω** i ***f*** **kontinuirane** (samo je ***fs*** **diskretna**). Digitalnu kružnu frekvenciju **ω** koristiti ćemo često, a nazivat ćemo je digitalna frekvencija iako neki autori koriste *f*.

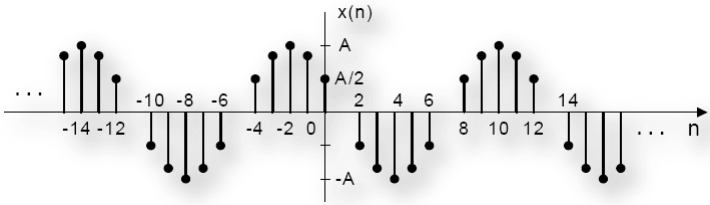
Na slikama je prikazan odnos ω i *F*. Gore je odnos na linearnoj skali, a dolje na kružnici. Analogna frekvencija *F nije* periodična, može imati bilo koju vrijednost između − ∞ do ∞. Digitalna frekvencija ω pokazuje kružnu prirodu, mijenja se kružno s periodičnošću 2π i s centralnim periodom [−π,π] odgovara Nyquistovom intervalu [-*fs*/2, *fs*/2].

To znači da će sinusoide različitih frekvencija ω u tom intervalu biti odijeljene, dok će sinusoide s frekvencijama izvan tog interval biti alias u tom intervalu.



Periodičnost diskretne sinusoide

Na slici je signal ***cos (πn/6 + π/3).*** Anvelopa uzoraka očito je sinusoidalna i periodična. Diskretni signal stvarno je periodičan s periodom 12 uzoraka (Možemo izbrojati uzorke i izračunati period *2π/(π/6) = 12*).



Postoje slučajevi kada je diskretni signal periodičan, a anvelopa ne izgleda sinusoidalno iako može izgledati periodično.

Takav je signal na idućoj slici : ***x(n) =cos5π n/6*** .

Anvelopa nema oblik sinusoide, ali signal izgleda periodično i periodičan je izbrojimo li uzorke jer mu je period 12 uzoraka.

Izračunamo li periodičnost, *2π/(5π/6)* rezultat nije jednak 12 ?

Isto tako postoje slučajevi gdje uzorci vremenski diskretnog signala leže na sinusoidalnoj periodičkoj anvelopi dok signal nije periodičan jer uzorci ne čine periodičku sekvencu.

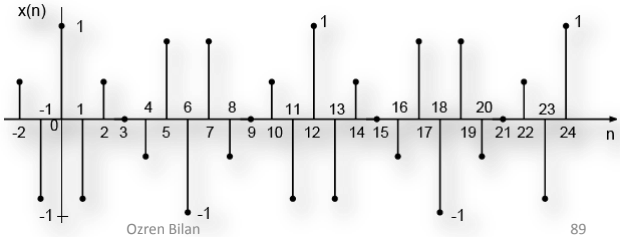
Periodičnost diskretne sinusoide je zbunjujuća, pa očekujemo **kriterij**. Pogledajmo periodičku diskretnu sinusoidu *cos ωn* s periodom *N uzoraka*: $\cos\omega n = \cos\omega(n+N) = \cos(\omega n + \omega N)$

Uvjet je:

mora postojati neki cijeli broj *m koji* pomnožen s *2π daje ωN*

$\omega N = 2\pi m$, m *cijeli broj*

$$\frac{\omega}{2\pi} = f = \frac{m}{N}$$



To znači da *ω/2π* (ili *f*) mora biti racionalna funkcija (odnos dvije cjelobrojne vrijednosti). Stvarni period *N* bit će jednak nazivniku *m/N* nakon skraćenja. Ako ne dobijemo racionalni broj signal nije periodičan (onda je neperiodičan ili aperiodičan). Kako znamo, iako diskretna sinusoida može, a ne mora biti periodična njena **anvelopa uzoraka uvijek je periodična ali ne mora izgledati sinusoidalno**. Primjeri će to dodatno razjasniti:

Signal **cos(*πn/6*)** je periodičan jer je *ω/2π =(π/6)/2π= 1/12* racionalna funkcija, a period je 12.

Signal **cos(5πn/6)** je periodičan jer je racionalna funkcija, a period je 12 (period nije *2π/(5π/6)* kao analognoj sinusoidi).

Signal **cos(*πn/8*)** je periodičan s periodom 16.

Naglasit ćemo nešto o periodičnosti diskretne sinusoide:

Male promjene digitalne frekvencije vode na velike promjene perioda:
npr., *ω/2π =51/100* period je 100, a
***ω/2π* = 50/100 = 1/2, period je 2.**

Kompleksna eksponencijala

Analiziramo signal tipa:

$x(n) = a^n$

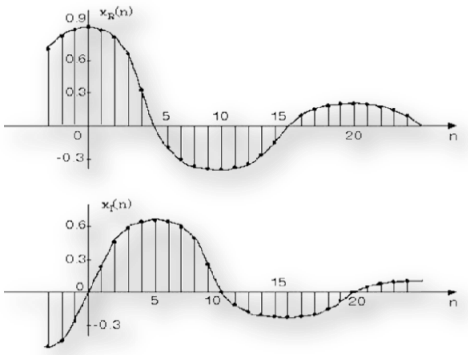
s kompleksnom konstantom *a*, koju nazivamo kompleksna eksponencijala ili kompleksna sinusoida. Neka je

$a = r\,e^{j\omega}$

onda je signal

$$x(n) = (r\,e^{j\omega})^n = r^n e^{j\omega n} = 2^n(\cos\omega n + j\sin\omega n)$$

Jednadžbu možemo nacrtati:



Realna i imaginarna komponenta i magnituda i faza su:

$$x_R(n) = r^n \cos n\,\omega$$
$$x_I(n) = r^n \sin n\,\omega$$
$$|x(n)| = \sqrt{x_R^2(n) + x_I^2(n)} = r^n$$
$$\Phi(n) = \tan^{-1}\frac{x_I(n)}{x_R(n)} = \tan^{-1}(\mathrm{tg}\,n\,\omega) = n\,\omega$$

VREMENSKI DISKRETNI SUSTAVI

Sustav djeluje na ulazni signal *x(n)* kako bi na izlazu proizveo izlazni signal *y(n)*. Označimo li sa *S[.]* djelovanje sustava možemo napisati:

$$y(n)=S[x(n)].$$

Npr. ako sustav množi s 2; S=2* onda je

$$y(n)=2*x(n)$$

Interesira nas **samo djelovanje sustava na ulazni signal**, a **ne interesira nas struktura** (*hardware i software*) sustava. Jedan od načina opisa sustava je specificiranje **ulazno izlazne jednadžbe diferencija** sustava (isto kao diferencijalne jednadžbe analognih sustava). Primjer ovog sustava daje jednadžbu $y(n)=2x(n)$, međutim, u najvećem broju slučajeva jednadžbe su složenije, što ćemo vidjeti na primjerima...

Ulazno izlazna jednadžba diferencija

Četiri sustava su opisana jednadžbom diferencija:

(a) $y(n) = x(n-1)$

(b) $y(n) = x(n+1)$

(c) $y(n) = (1/3)\,[x(n-1) + x(n) + x(n+1)]$

(d) $y(n) = \max [x(n-1), x(n), x(n+1)]$

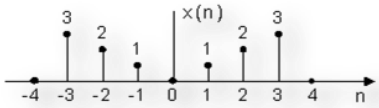
Ulazni signal je $x(n) = \begin{cases} |n|, & -3 \le n \le 3 \\ 0, & \text{inače} \end{cases}$

Potrebno je odrediti izlaz za svaki slučaj.

Rješenje

Ulazna sekvenca za sva četiri slučaja je

$x(n) = [\,0, 3, 2, 1, \mathbf{0}, 1, 2, 3, 0\,]$

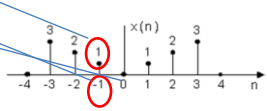


Jedan od (*sirovih*) načina je da odredimo izlazni signal primjenom jednadžbe za $n = 0, 1, 2, \ldots$ sve dok izlaz ne postane jednak nuli. Nakon toga računamo izlaz za točke $n = -1, -2, \ldots$ sve dok ne postane jednak nuli.

Bolji i brži način je promatranje signalne jednadžbe kako bi se shvatilo njeno djelovanje, a zatim zaključimo na oblik izlaza

(a) $y(n) = x(n-1)$

$y(0) = x(-1) = 1$
$y(1) = x(0) = 0$
$y(2) = x(1) = 1$
$y(3) = x(2) = 2$
$y(4) = x(3) = 3$
$y(5) = x(4) = 0$
$y(-1) = x(-2) = 2$
$y(-2) = x(-3) = 3$
$y(-3) = x(-3) = 0$



Za ulaznu sekvencux(n) = [0, 3, 2, 1, **0**, 1, 2, 3, 0] izlazna sekvenca će biti $y(n) = [\,0, 3, 2, \mathbf{1}, 0, 1, 2, 3, 0\,]$

Izlazni signal jednak je ulaznom koji kasni za jedan indeks (uzorak), npr. izlaz $y(0)$ u $n = 0$ je uzorak $x(-1)$ u $n = -1$. Sustav je **jedinično kašnjenje *unit delay***.

(b) $y(n) = x(n+1)$

Analizom jednadžbe vidimo da je izlaz $y(0)$ u $n = 0$ jednak ulazu $x(1)$ u $n = 1$, pa je sustav **jedinično prethođenje *unit advance***.

(c) $y(n) = \frac{1}{3}[x(n-1) + x(n) + x(n+1)]$

Izlazni signal na indeksu n je srednja suma prethodnog ulaza $x(n-1)$, trenutnog ulaza $x(n)$ i slijedećeg (budućeg) ulaza $x(n+1)$. Izlazni signal računamo na slijedeći način:

$y(0) = (1/3)\ [x(-1) + x(0) + x(1)] = (1/3)\ (1+0+1) = 2/3$
$y(1) = (1/3)\ [x(0) + x(1) + x(2)] = (1/3)\ (0+1+2) = 1$
$y(2) = (1/3)\ [x(1) + x(2) + x(3)] = (1/3)\ (1+2+3) = 2$
... nastavljanje procesa dovodi nas do cjelovite izlazne sekvence
$y(n) = [\,0, 1, 5/3, 2, 1, \mathbf{2/3}, 1, 2, 5/3, 1, 0\,]$

(d) $y(n) = \max[x(n-1), x(n), x(n+1)]$

Trenutni izlaz je maksimalna vrijednost tri ulaza: prethodnog, sadašnjeg i budućeg. Proračun slijedi:

$y(0) = \max[x(-1), x(0), x(1)] = \max(1,0,1) = 1$
$y(1) = \max[x(0), x(1), x(2)] = \max(0,1,2) = 2$
$y(2) = \max[x(1), x(2), x(3)] = \max(1,2,3) = 3$

Nastavak daje cjelovitu izlaznu sekvencu:

$y(n) = [\,0, 3, 3, 3, 2, \mathbf{1}, 2, 3, 3, 3, 0\,]$

Temeljni blokovi diskretnog sustava

Struktura vremenski diskretnog sustava sastoji se od različitih **gradbenih blokova**.

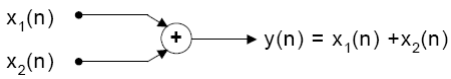
Blokovi se mogu realizirati

- ***hardwareom* (logički sklopovi),**
- ***softwareom* (programima) ili**
- **kombinacijama**.

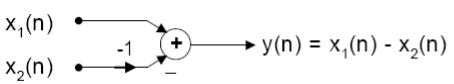
Upoznat ćemo sve sustavske gradbene blokove i shematske simbole.

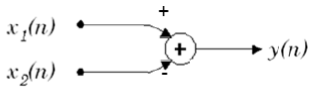
Njihovom primjenom **možemo nacrtati blok dijagram** ili dijagram toka koji opisuje **bilo koji sustav**.

Sumator (zbroj) signala



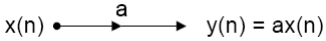
Subtraktor (oduzimanje) signala



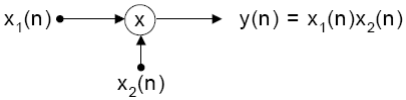


Skalarni **množitelj**

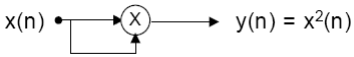
Ako je skalar realan i pozitivan svi ulazi u sumator su pozitivni



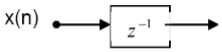
Signalni **množitelj**

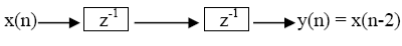


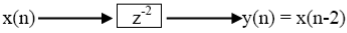
Kvadriranje signala



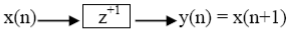
Jedinično i dvostruko **kašnjenje**



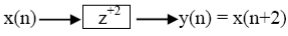




Jedinični i dvostruki **pomak unaprijed**



dvostruki pomak unaprijed



Koristimo simbol z^{-1} jer u z-transformaciji **množenje s z^{-1} ima učinak kašnjenja za jedinično vrijeme**. Neki autori koriste simbole D i T umjesto z^{-1} .

TIPOVI DISKRETNIH SUSTAVA

Vremenski diskretni (digitalni) sustavi sastoje se od **nekoliko osnovnih tipova različitih karakteristika**.

Njihovo **kategoriziranje** daje nam dublje **razumijevanje sustava** i izbor **najprikladnije metode analiziranja**.

Sustavi sa i bez memorije

Sustavi bez memorije ili **statički** sustavi ne trebaju memoriju. Ulazni i izlazni signali odvijaju se istovremeno. Npr:

$$y(n) = 2x(n)$$

$$y(n) = 2x(n) - x2(n)$$

U stvarnosti postoji vrlo malo kašnjenje izlaza i izlaza zbog propagacijskog kašnjenja sustava

Sustavi s memorijom ili **dinamički** sustavi trebaju memoriju za pohranu prošlih i budućih vrijednosti potrebnih za obradu . Npr.:

$y(n) = x(n) + 0.8x(n\text{-}1)$: jedna memorijska ćelija

$y(n) = \frac{1}{3}[x(n\text{-}1) + x(n) + x(n+1)]$: dvije memorijske ćelije

$y(n) = \sum_{k=-\infty}^{+\infty} x(n-k)$: beskonačna memorija

Kauzalni (uzročno-posljedični) i nekauzalni signali

Zamislite najednostavniji elektronički sklop. Ako mu na ulaz primijenite kratki impuls, na izlazu će te vidjeti izlaz. To su uzrok (*lat. causa*) i posljedica na kojima se temelji sav naš poznati svijet. Sa sigurnošću možemo tvrditi: **posljedica se javlja nakon uzroka**. To je temeljna karakteristika vremena kakvog poznajemo. Usporedimo navedeno svojstvo s DSP sustavom koji mijenja ulazni signal u izlazni, pri čemu su oba pohranjena u spremniku računala. Ako je taj sustav isti kao sustav stvarnog svijeta u kojem živimo, onda mora biti temeljen na istim principima uzročno posljedičnih veza. Npr. vrijednost uzorka broj 7 ulaznog signala može djelovati samo na uzorak broj 7 ili viši izlaznog signala. Sustavi koji na tom principu temelje svoj rad nazivaju se **kauzalni**. DSP procesori ne moraju nužno funkcionirati na taj način.

Budući da su ulazni i izlazni signali brojevi u spremniku računala, bilo koji ulazni signal može djelovati na bilo koji izlazni signal.

U DSP POSLJEDICE MOGU DJELOVATI NA UZROK

Kauzalni i nekauzalni sustavi

U kauzalnom (uzročno-posljedičnom) sustavu rezultat nastaje nakon uzroka ili najranije istovremeno (simultano). To znači da izlaz na indeksu n ovisi samo o ulazu u n, n–1, n–2,…, a nikako ne može ovisiti o n+1, n+2,…

U **nekauzalnim sustavima** izlaz ovisi i o budućim ulazima zato jer se svi nalaze u memoriji računala. Npr.

(a) $y(n) = 2x(n) - 3x^2(n)$: kauzalan

(b) $y(n) = \frac{1}{3}[x(n\text{-}1) + x(n) + x(n+1)]$: nekauzalan zbog posljednjeg izraza

(c) $y(n) = \sum_{k=0}^{\infty} x(n-k)$: kauzalan

(d) $y(n) = \sum_{n=-\infty}^{\infty} x(n)$: nekauzalan

(e) $y(n) = \sum_{n=-\infty}^{\infty} x(n-k)$: nekauzalan

(f) $y(n) = x(-n)$: nekauzalan

(g) $y(n) = x(n^2)$ nekauzalan

(h) $y(n) = x(2n)$: nekauzalan

Sustavi za obradu u stvarnom (realnom) vremenu (tzv. **on-line processing**), moraju biti kauzalni. Pri naknadnoj obradi (**off-line processing ili batch processing ili block processing**), sustavi mogu biti nekauzalni budući da su svi uzorci pohranjeni u memoriji, a mnogi pod njih imaju buduću vrijednost obzirom na izbor nulte točke.

Koncept kauzalnosti se primjenjuje i na signale.

U nekom specifičnom sustavu signal $x(n)$ klasificira se kao:

- **Kauzalan** ako je $x(n) = 0$ za $n < 0$
- **Antikauzalan** ako je $x(n) = 0$ za $n \geq 0$
- **Dvostran ili bilateralan** ako $x(n)$ postoji za sve n (<0 i ≥ 0)

Tako npr. **jedinični step** $u(n)$ **je kauzalan**, a $u(-n-1)$ **je antikauzalan**, $a^{|n|}$ **je bilateralan**.

Ako ih nacrtamo, dijagrami ovih signala jasno pokazuju razlike.

Vremenski-nepromjenjivi i vremenski-promjenjivi sustavi

Karakteristike sustava **mogu se vremenski** mijenjati tako da izlaz ovisi o ulazu ali i o **trenutku primjene ulaznog signala**. Takav sustav nazivamo **vremenski promjenjivi**.

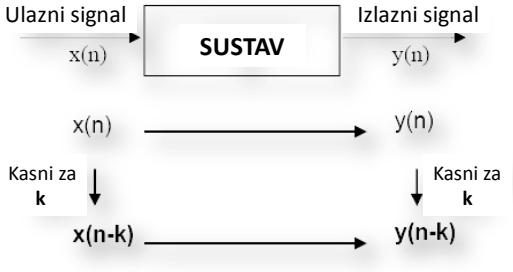
S druge strane mnoge sustave možemo smatrati **vremenski nepromjenjivim** jer im izlaz ne ovisi o vremenu primjene ulaza.

Vremenska nepromjenjivi sustav ocjenjuje se kriterijem:

Ako je $x(n) \longrightarrow y(n)$

Onda vrijedi $x(n-k) \longrightarrow y(n-k)$

Kriterij ilustrira slika:



Linearni i nelinearni sustavi

Značaj linearnosti i nelinearnosti vremenski diskretnih sustava isti je kao i za analogne sustave. Pretpostavimo dva ulazna signala *x1*(*n*) i *x2*(*n*) koji su primijenjeni odvojeno na sustav koji daje korespodentne izlaze *y1*(*n*) i *y2*(*n*). Ako linearna kombinacija ova dva ista ulaza daje istu linearnu kombinaciju izlaza sustav je linearan. Ako ne daje onda je sustav nelinearan. Dakle, **linearnost implicira skalabilnost (proporcionalnost) i superpoziciju**. Definicija linearnosti ilustrirana je na slici.

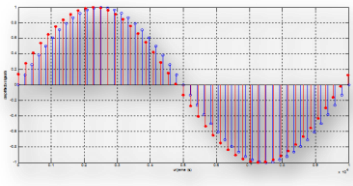
Ako je $x_1(n) \rightarrow y_1(n)$

Onda vrijedi $x_2(n) \rightarrow y_2(n)$

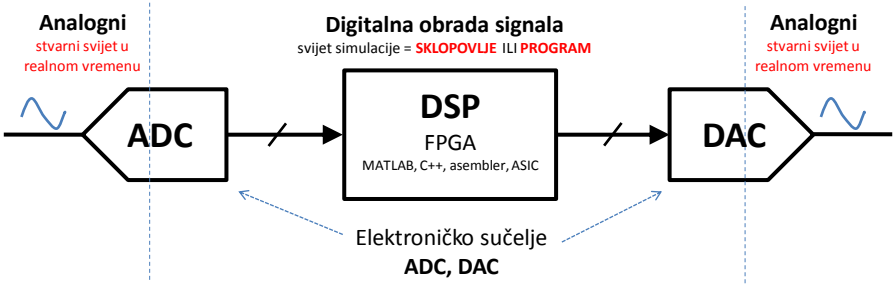
$a_1x_1(n) + a_2x_2(n) \rightarrow a_1y_1(n) + a_2y_2(n)$

Analogno digitalna i digitalno analogna pretvorba i višetaktni i naduzorkujući sustavi

Analogni signali su općenito vremenski kontinuirani. Pri izvođenju DSP ne koristimo cjeloviti analogni signal nego ga zamjenjujemo vrijednostima amplitude uzete u pravilnim intervalima. To je **sempliranje**. Problem je u tome da moramo semplirati signal na takav način kojim će odabrani uzorci točno predstavljati signal. Dakle iz odabranih uzoraka moramo **točno rekonstruirati** izvorni analogni signal iako ne uvijek idealno. Sempliranjem se analogni vremenski kontinuirani signal pretvara u odgovarajući **diskretni vremenski promjenjivi signal** što nam omogućava obradu na digitalnim sustavima (računalima ili DSP procesorima). Sempliranje prate još dvije operacije; **kvantizacija** i **binarno kodiranje**. **Analogno-Digitalni pretvarači** (skraćeno ADC ili A/D) obavljaju sva tri koraka.



Analogno digitalna i digitalno analogna pretvorba



- **Stvarni svijet**: kontinuirano vrijeme, signali kontinuirane amplitude.
- **Digitalni svijet**: diskretno vrijeme, signal predstavljen diskretnom amplitudom.
- **Sklopovi sučelja stvarnog i digitalnog svijeta**: AD i DA pretvarač. Prema primjeni imaju različite zahtjeve brzine i rezolucije (točnosti).

Mogući scenarij primjene

Bilo koja primjena koja koristi senzore i aktuatore

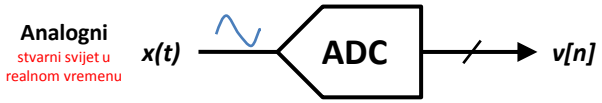
- Bežična: *RF Rx* i *Tx*
- Parica, žična : *ADSL* modem, *USB, SPDIF*
- Koaksijalni: Kabelski modem, *COAX*
- Serijsko/Optička veza: 10G+ *ADC* modulacija i ekvalizacija
- Biomedicinska primjena (npr. *senzor razine glukoze u krvi uključuje inzulinsku pumpu*),.....

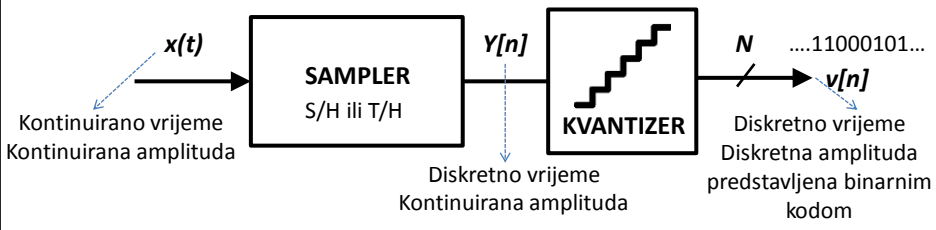
Interesira nas

- snimanje zvuka: 16, 24, 32-bit dvokanalni *ADC*
- *FX*, rekvantizacija, resempliranje, editing...
- reproduktori pohranjenih zvučnih datoteka: zvučni *DAC*
- *HDD, Flash* reprodukcija datoteka: disk prema mikroprocesoru...

Zahtjevi brzine i rezolucije mijenjaju se prema aplikaciji.

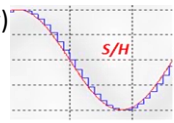
Analogno digitalni pretvarač *ADC*

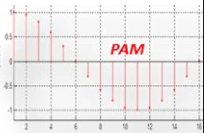




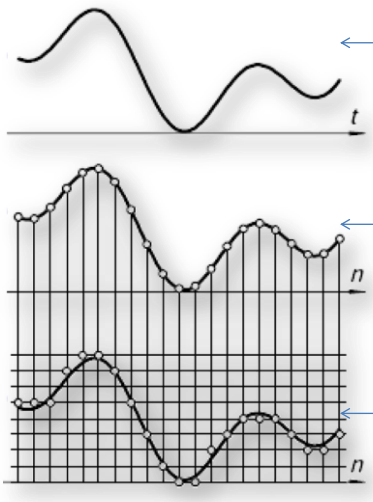
AD pretvarač izvršava

- **Sempliranje** (diskretiziranje po vremenu *fs=1/Ts*)
  - **Principom zadržavanja (*Sample/Hold*)** ili
  - **Pravokutnim impulsima (*PAM postupak*)**
- **Kvantiziranje** (diskretiziranje po amplitudi).
- **Kodiranje** (pridruživanje koda diskretnim vrijednostima).





Analogno digitalna pretvorba



Kontinuirani signal

SEMPLIRANJE

Diskretni signal

KVANTIZACIJA

Niz diskretnih binarno kodiranih vrijednosti

Ozren Bilan

Sempliranje signala

Analogni signali su općenito vremenski kontinuirani. Pri izvođenju DSP ne koristimo cjeloviti analogni signal nego ga zamjenjujemo vrijednostima amplitude uzete u pravilnim intervalima. To je **sempliranje**.

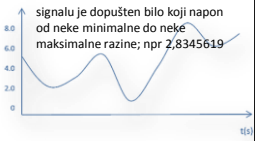
Problem je u tome da moramo semplirati signal na takav način kojim će odabrani uzorci točno predstavljati signal. Dakle iz odabranih uzoraka moramo **točno rekonstruirati** izvorni analogni signal iako ne uvijek idealno.

Sempliranjem se analogni vremenski kontinuirani signal pretvara u odgovarajući **diskretni vremenski promjenjivi signal** što nam omogućava obradu na digitalnim sustavima (računalima ili DSP procesorima)

Sempliranje prate još dvije operacije; **kvantizacija** i **binarno kodiranje**. **Analogno-Digitalni pretvarači** (skraćeno ADC ili A/D) obavljaju sva tri koraka.

Sempliranje, kvantizacija i binarno kodiranje

Na dijagramu je prikazan ustaljeni signal kojeg nazivamo **analogni signal**. Svaku vrijednost uzorka **x** možemo prikazati i vrlo visokom točnošću, npr. ako ne zaokružujemo vrijednost ***x(1) = 2,8…*** na 2 ili 3. Međutim, tada su nam potrebni vrlo veliki registri i duge riječi u memoriji. Zbog riječi konačne dužine, zaokružujemo vrijednosti uzoraka. U slučaju ***x(1) = 2,8…*** zaokružujemo na ***2***. Tako se unosi pogreška koju nazivamo **šum kvantizacije**.



napon V kont.	razina x(t)	OBLIK ANALOGNOG I DISKRETNOG SIGNALA								Maksimalna razina 3b sustava = $7_{10} = 111_2$
8	8 =?									
7	7									
6	6									
5	5									
4	4									→ VRIJEME
3	3									
2	2									
1	1									
0	0									
nT→		0	1T	2T	3T	4T	5t	6T	7T	8T	→ VRIJEME
n→		0	1	2	3	4	5	6	7	8	← BROJ KORAKA
x(n) →		5.5	2.8	3.8	5.3	1.5	4.6	8.4	6.9	7.3	← semplirani signal vrijeme je diskretno amplituda kontinuirana
		0.5	0.8	0.8	0.3	0.5	0.6	1.4	0.9	0.3	← Greška kvantizacije
Diskretnom signalu su dopuštene samo naponi razina kvantizacije od 0 do 7.		5	2	3	5	1	4	7	6	7	← kvantizirano diskretni signal po vremenu i amplitudi
		101	010	011	101	100	111	110	110	111	← kodiran 3bita/uzorak

↑ točka izvan radnog područja (*zasićenje ili saturacija*) zaokružena je na 7, a ne na 8

Signal na slici sempliran je u pravilnim intervalima

$$t = n\,T$$

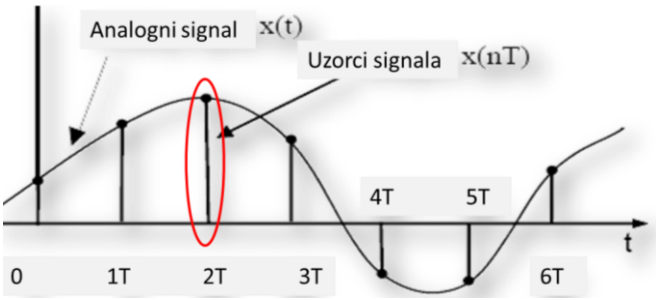
gdje *je* n cjelobrojna vrijednost, pozitivna i negativna,

$$n = 0, 1, 2,.., -1, -2,…$$

To je **uniformno sempliranje** koje se rutinski koristi. Vrlo rijetko koristi se i neuniformno sempliranje.

Uzorke signala $x(t)$ označavamo

$$x(nT) \text{ ili } \hat{x}(n).$$



Kako sempliramo signal?

Na vrhu je signal $x(t)$,

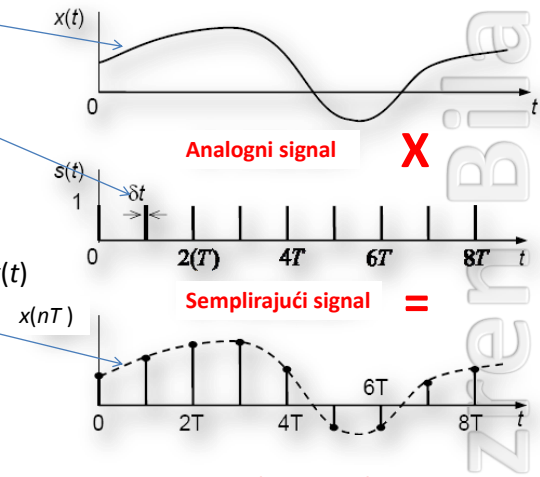
u sredini je semplirajući signal $s(t)$

koji je predstavljen pravilnom **sekvencom** uskih impulsa δt amplitude 1.

Množenjem ova dva signala dobivamo trenutne vrijednosti $x(t)$ predstavljene uzorcima (nT).

Dakle **sempliranje je množenje** analognog signala x(t) i semplirajućeg signala ili semplirajuće funkcije s(t):

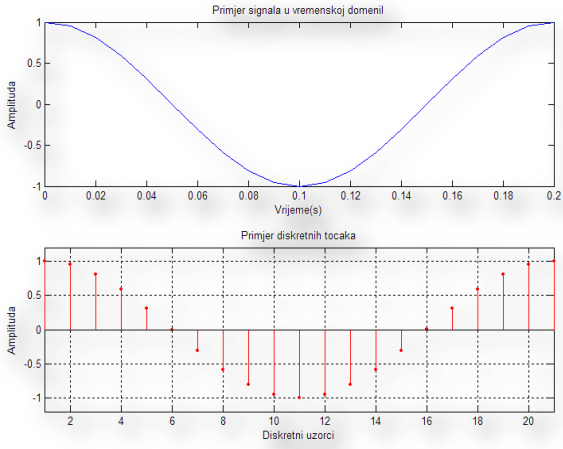
$$x(t) = x(nT) = x(t)s(t)$$



Ozren Bilan

Prikaz u MATLAB-u

```
clear all;
close all;
freq = 5; % 5Hz
n=0:.01:.2; %  broj diskretnih točaka na osi X
x = cos (2*pi*freq*n); % cosinus signal
figure('Color',[1 1 1]);
subplot(2,1,1);
plot(n,x);
title('Primjer signala u vremenskoj domeni');
xlabel('Vrijeme(s)');
ylabel('Amplituda');
ns=1:length(n);
subplot(2,1,2);
stem(ns,x,'.r');
title('Primjer diskretnih točaka');
xlabel('Diskretni uzorci');
ylabel('Amplituda');
axis([1 21 -1.2 1.2]);
grid on;
```

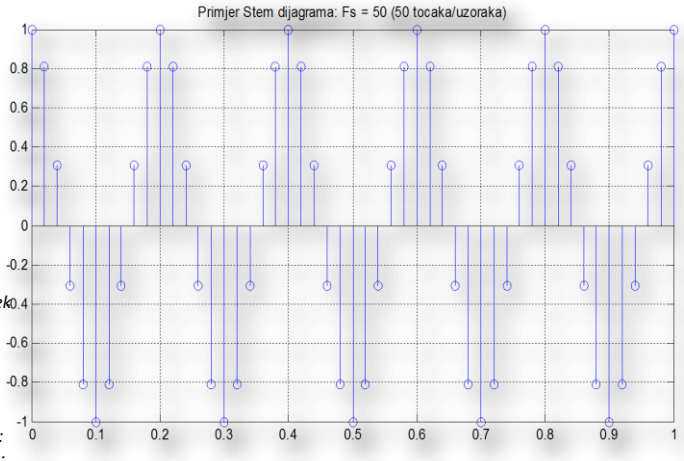


Ozren Bilan

Kako bi točno konstruirali sinusoidu u Matlabu potrebno je odrediti jednadžbu sinusoide, zatim, koliko često treba uzimati uzorke i koliko će to trajati. U biti **naređujemo**: gledaj 5Hz sinusoidu u trajanju 1 sekunde. U tom intervalu uzmi 50 uzoraka pa nacrtaj. Moramo vidjeti uzorak svakih 0.02 sekunde, ukupno 51 točku. Kako bi odredili signal moramo odrediti:

1. frekvenciju sempliranja FS
2. period uzoraka Ts
3. broj uzoraka
4. vremenski indeks Tn
5. signal x(t) i dijagram

```
Fs = 50;
Ts = 1/Fs;
n =0:1*Fs;
Tn =Ts*n;
clear all;
fs = 50;
Ts=1/fs;
n=0:1*fs;
% vremenski otvor prozor
%(fs=broj uzoraka u sek), 1 sek
tn=n*Ts; % indeks uzorka
sig_a=cos(2*pi*5*tn);
figure('Color',[1 1 1]);
stem(tn,sig_a);
grid
title('Primjer stem dijagrama:
Fs = 50 (50 točaka/uzoraka)');
axis('tight');
```



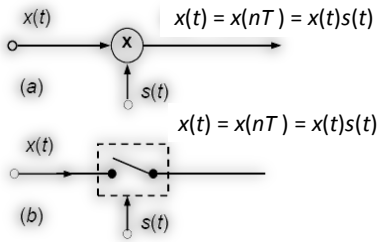
Proces sempliranja možemo ilustrirati ***a)* množenjem** ili ***b)* prekidačem**. Slika *b)* pokazuje električni prekidač kao način primjene sampliranja: kad kontakti zatvore (za neki kratki interval), signal prolazi, a tada se kontakti otvaraju pa nema prolaska signala.

Interval *T* nazivamo **interval sempliranja** ili **period sempliranja**,

fs = *1/T* je **frekvencija sempliranja** (Hz ili uzoraka/s).

Uzorke bilježimo *x*(*nT*) pri čemu *T* uobičajeno ima jediničnu vrijednost.

Cjelobrojna vrijednost *n,* onda postaje **indeks**, kojeg ponekad nazivamo vremenski indeks ili uzorak.



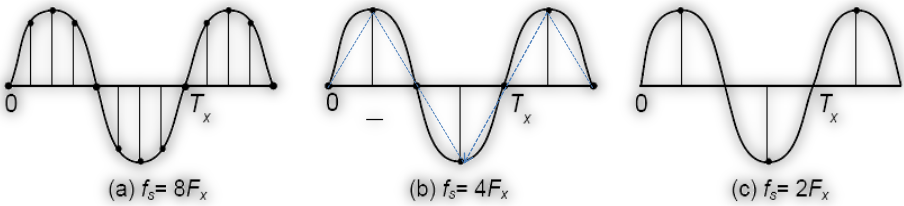
Kako ćemo odrediti da li je sempliranje ispravno, tj. da li su uzorci preblizu, predaleko ili su točno udaljeni?

Pogledajmo sempliranje sinusnog vala x(t) perioda *Tx* i frekvencijom *Fx* = 1/*Tx,* frekvencijom sempliranja *fs*. Slika prikazuje isti sinusni valni oblik sempliran s tri različite frekvencije *fs*.

a) ***fs* = 8*Fx***, uzorci su vrlo blizu i vrlo dobro predstavljaju valni oblik. Dakle iz uzoraka možemo vrlo dobro rekonstruirati signal.

b) ***fs* = 4*Fx*** , uzorci još uvjek dobro predstavljaju signal. Tako ako povežemo pravcima sukcesivne uzorke dobit ćemo trokutasti valni oblik. Propustom kroz **analogni** niskopropusni filter izgladit ćemo valni oblik i dobiti sinus.

c) ***fs* = 2*Fx*** , frekvencija sempliranja jednaka je dvostrukoj frekvenciji signala. Taj **slučaj je kritičan** jer uzorci mogu, a ne moraju predstavljati izvorni signal, što ovisi o položaju točaka sempliranja.



(a) $f_s = 8F_x$ (b) $f_s = 4F_x$ (c) $f_s = 2F_x$

Teorem sempliranja glasi:

Kako bi uzorci točno predstavljali izvorni analogni signal, frekvencija sempliranja mora biti viša od dvostruke maksimalne frekvencijske komponente analognom dignala: *fs* > 2*FM*

Granična frekvencija 2*FM zove se* **Nyquistova frekvencija**, a **centralni frequencijski interval [-*fs*/2, *fs*/2] zove se Nyquistov interval.**

Ako analogni signal sadržava fundamentalnu frekvenciju 1 kHz i drugi harmonik 2 kHz, tada frekvencija sempliranja mora biti viša od 2 x 2 kHz = 4 kHz, npr. 5 kHz ili viša.

Drugi primjer je **govorni signal u telefoniji**. On se graničava visokokvalitetnim analognim filterom na *FM* = 3.4 kHz, tada frekvencija sempliranja mora biti viša od 2 x 3.4 = 6.8 kHz, recimo 8 kHz ili više.

CD ima gornju graničnu frekvenciju 20kHz, a *fs 44,1kHz*.

DVD-Audio ima frekvenciju sempliranja od *48 kHz* do 192 kHz

DXD sustavi *385 kHz*.

SACD i DSD sustavi *2,8 MHz* ili *5,6 MHz*, a postoje sustavi *24,576 MHz*

Prikazat ćemo tablično **frekvencije sempliranja za zvučne signale**

Izbor frekvencije sempliranja za zvuk

8 kHz telefon, mobilne mreže

11,025 kHz PCM niske kvalitete, MPEG audio

22,050 kHz PCM niže i srednje kvalitete, MPEG audio i audio analiza niskofrekvencijskog područja, dovoljno za digitalizaciju starih audio zapisa na 78 okr/min)

32 kHz Mini DV digitalni video, video zapis s dodatnim audio kanalima DAT, DSR, NICAM

44,056 kHz PCM adapter za NTSC video sustav

44,100 kHz Audio CD, audio zapis temeljen na PAL video sustavu, najčešće korišten

47,250 kHz prvi komercijalni digitalni uređaj za snimanje Denon

48 kHz najniža frekvencija za profesionalnu primjenu digitalnog signala DAT, DASH, editori

50 kHz idealna frekvencija uzorkovanja za audio signal, nije prihvaćena

88,2 kHz u profesionalnoj primjeni masteringa CD.

96 kHz DVD-Audio, kvalitetno poluprofesionalno i profesionalno snimanje

176,4 kHz Profesionalna primjena masteringa CD, HD signal.

192 kHz DVD-Audio, visoko-kvalitetno profesionalno snimanje i editor audio zapisa

384 kHz DXD zapis i editiranje

2,8 MHz SACD, 1-bit sigma-delta modulacija u postupku *Direct Stream Digital*, *SONY Philips*

5,6 MHz Dvostruka frekvencija sempliranja DSD, *Accuphase* postupak

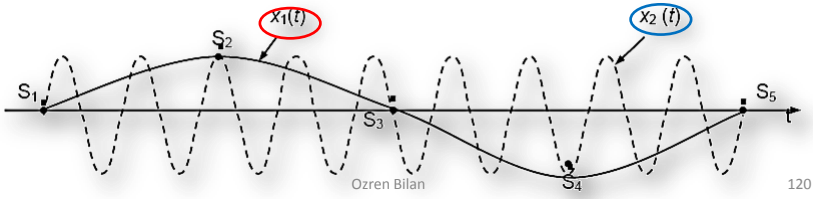
24,576MHz *512x44,1kHz l*aboratorijski *DSD* postupak sveučilište *Waseda*

Alias

Što će se dogoditi ako nije zadovoljen Nyquistov teorem pa signal sampliramo frekvencijom nižom od Nyquistove. Na slici je signal *x1*(*t*) sampliran 4 puta. To su uzorci ili točke *S1*, *S2*, *S3 i S4 u* jednom periodu signala, dakle $fs = 4Fx1$.

Iz uzoraka možemo rekonstruirati ***x1*(*t*)**. Za visokofrekvencijski signal ***x2*(*t*)** postoje ista 4 uzorka *S1, S2, S3 i S4* u 9 perioda, pa je frekvencija sampliranja (4/9) *Fx2*, dakle niža od Nyquistove. Iz uzoraka *x2*(*t*) koje smo dolbili sampliranjem, možemo rekonstruirati *x1*(*t*) ali ne možemo *x2*(*t*).

Dakle, visoko frekvencijski signal sampliran frekvencijom nižom od Nyquistove rekonstruirat će se kao niskofrekvencijski signal. **Rekonstruiran pogrešni niskofrekvencijski signal naziva se alias izvornog visokofrekvencijskog signala.**



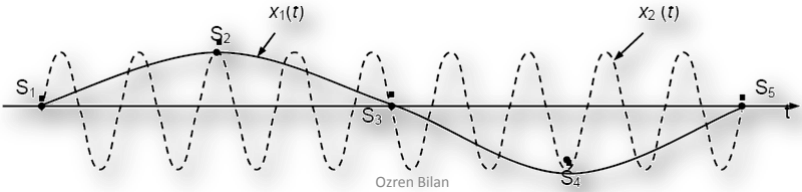
Kako bi spriječili pojavu aliasa postoje dva pristupa:

- **Povisiti frekvenciju sempliranja** u skladu s teoremom ili
- **filtrirati nepotrebne visokofrekvencijske komponente** iz vremenski kontinuiranog (analognog) signala.

Frekvenciju signala ograničavamo niskopropusnim filtriranjem, filtrom kojega nazivamo **antialias prefilter.** Nakon filtriranja najviša frekvencija signala mora biti manja od polovine frekvencije kojom ćemo signal samplirati. Ako filter nije idealan, potrebno je postrožiti uvjete. Tako pri obradi glasa može se dogoditi da filter propušta komponente iznad 3,4kHz s nekim malim amplitudama. U tom slučaju frekvencija sampliranja mora biti 8kHz ili viša.

Općenito se može pokazati da analognom signalu frekvencije F sempliran frekvencijom *fs* , prvo dodajemo i oduzimamo frekvencije prema relaciji:

f0 = F ± mfs , gdje je m= 0, 1, 2, . . onda pogledamo koje frekvencije leže unutar *Nyquistovog intervala*, to će biti rekonstruirane frekvencije.



Kvantizacija

Analogni signali su kontinuirani, a digitalni signali su diskretnih vrijednosti zbog konačnog broja bita u binarnim kodovima. Zbog toga se sempliranе vrijednosti analognog signala moraju kvantizirati u diskretne razine jednolike širine kako bi odgovarale binarnim kodovima. Tu zadaću izvršavaju analogno-digitalni pretvarači (ADC ili A/D). Kvantizacijski postupak možemo zamisliti kao propuštanje analognog ulaznog signala kroz konstantno razlomljenu stepeničastu funkciju.

Ako je ulazni signal u području od -4 do 4 (volt) podijeljen u 8 različitih **kvantizacijskih razina** označenih $x_q(n)$, a svaka razina je širine 1 (volt): razina 1 je od -4 do -3, razina 2 od -3 do -2,…, razina 7 od 3 do 4. Budući da postoji 8 razina, koristimo binarni kod sa 3 bita kako bi prikazali sve razine kvantizacije. Zbog toga će pripadajući binarni izlazni signal biti 000, 001,…, 111. Područje koje predstavlja udaljenost dvije susjedne razine naziva se **korak kvantizacije**, a označava se najčešće sa **q**.

Zvučni signali pri obradi kvantiziraju se za **CD sa 16 bita**, a zvučni signali visoke rezolucije **od 24 do 32 bita** dok se pri profesionalnoj obradi koriste **64 bitne riječi**. Pri kodiranju govora minimalna vrijednost kodiranja fonema je 6 bita, telefonski govor 12 bita, a s kompandiranjem 8 bita. ADPCM sustav ima i preporuku za kodiranje sa samo 2 bita uz brzinu prijenosa od 16 kbit/s.

Na slikama je grafički prikazan postupak sempliranja i kvantizacije analognog signala. Pri tome mora biti zadovoljen teorem sempliranja.

Vrijednosti dobivene sempliranjem imaju beskonačnu preciznost i nazivaju se **vremenski diskretni signal**, a kvantizirane vrijednosti konačne preciznosti nazivaju se **digitalni signal**.

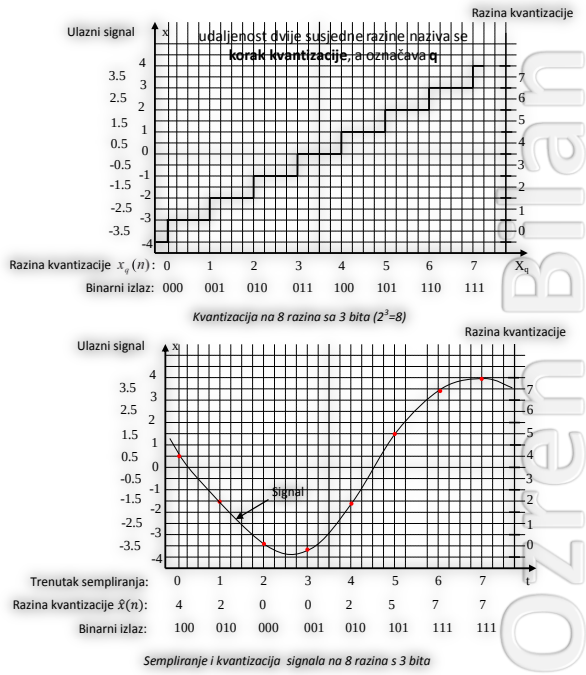
Vrijednosti digitalnog signala pokazuju pogrešku u području **[-q/2, q/2]** za svaku sempliranu vrijednost. Budući da je ova pogreška slučajna naziva se **šum kvantizacije**, a označava se ***e(n)***. Tada kvantizirane vrijednosti $\hat{x}(n)$ možemo zapisati u obliku

$$\hat{x}(n) = x(n) + e(n)$$

Šum kvantizacije ***e(n)*** modelira se kao slučajna varijabla jednoliko raspodijeljena u intervalu **[-q/2, q/2]** s gustoćom vjerojatnosti ***p(x)=1/q***. Efektivna vrijednost ili očekivana vrijednost ***e²(n)***, je

$$E[e^2(n)] = \int_{-\infty}^{\infty} p(x)x^2 dx = \frac{1}{q}\int_{-q/2}^{q/2} x^2 dx = \frac{q^2}{12}$$

Ako je kvantizer rezolucije ***B*-bita** njegovo ukupno radno područje bez preopterećenja bit će ***2^B·q***



Iznos **snage šuma kvantizacije** ili **varijancu** označavamo σ^2. Iznos koraka kvantizacije je *(B>>1)*

$$q = \frac{V_{fz}}{2^{B-1}} \approx \frac{V_{fz}}{2^B}$$

Pogreška kvantizacije Ako ADC ima B bita, a broj kvantizacijskih razina 2^B

$$\sigma_e^2 = \frac{q^2}{12}$$

Gdje je *Vfs* naponsko područje pune skale ADC s **bipolarnim ulaznim signalom**. Na slici je ova vrijednost *Vfs=4-(-4)=8*. Sinusnom naponu amplitude A, *Vfs* predstavlja napon vrh-vrh jednak 2A, pa će korak kvantizacije biti

$$q \approx \frac{2A}{2^B}$$

Analogno digitalnom pretvaraču ADC pogreška kvantizacije približno je jednaka polovini najmanje značajnog bita LSB. Tako 12-bitnom ADC uz područje ulaznog napona od *+/-10V (Vfs=20V)* najmanje značajan bit LSB je $20/2^{12}$=*4.9 mV* pa pogreška kvantizacije iznosi *2.45mV*. Sinusnom ulaznom signalu je srednja snaga signala dana izrazom $A^2/2$. Odnos signal šum u decibelima onda je:

$$SNR = 10\log_{10}\frac{A^2/2}{q^2/12} = 10\log_{10}\frac{3\times 2^{2B}}{2}$$
$$= 6.02B + 1.76dB$$

Izraz predstavlja **teoretski maksimum**. U praksi ulazni signali imaju stvarnu vrijednost odnosa signal šum manju od teoretski maksimalne vrijednosti zbog nesavršenosti sklopovlja i vanjskih smetnji.

Važno je uočiti kako se **odnos signal šum povećava porastom broja bita B**, a svaki bit pridonosi 6 dB. Pri pretvorbi digitalnih signala u pripadajuće analogne signale nastaje druga pogreška koja se naziva kvantizacija izlaznog signala.

Maksimalna pogreška uzorka digitiziranog signala je ±½ LSB pa je aditivni šum jednoliko distribuiran između ±½ LSB. Srednja vrijednost jednaka mu je nuli, a **standardna devijacija** iznosi $1/\sqrt{12}$ LSB = **0.29 LSB**. Dakle, propustom analognog signal kroz 8 bitni pretvarač dodat ćemo efektivni šum od 0.29/256 ili oko 1/900 vrijednosti pune skale pretvarača. 12 bitna pretvorba dodat će šum 0.29/4096=1/14000, a 16 bitna pretvorba dodat će 0.29/65536=1/227000. Budući da je **greška kvantizacije slučajna**, povećanjem broja bitova određujemo preciznost podataka.

Multirate ili ***višetaktna* obrada** predstavlja jednu od tehnika koja se koristi kako bi se umanjio učinak ulazno/izlazne kvantizacije signala.

Neka je:

- analogni signal maksimalne amplitude **1.0 volt**, i
- slučajni šum **1.0 mV$_{eff}$** (*odnos S/Š=**60dB***, jer je *20 log 1000=60*).

Pretvorba ovog signala u digitalni oblik s b=8 bita rezultirat će da

- 1.0 volt postane digitalni broj $N=2^b-1=$**255**, a
- 1.0 mV postat će **0.255 LSB**.

Signali slučajnog šuma se kombiniraju zbrajanjem njihovih varijanci. To znači da se zbrajaju kvadrati signala slučajnog šuma signala i greške kvantizacije,

$$(A^2 + B^2)^{1/2} = C.$$

Ukupni šum digitalnog signal određen je izrazom $(0.255^2 + 0.29^2)^{1/2}$ = **0.386 LSB**.

To je povećanje od *51%* iznad postojećeg šuma u analognom signalu.

Pretvorba ovog signala u 12 bitno ne bi rezultirala povećanjem šuma, a ništa se ne bi izgubilo zbog kvantizacije. Dakle, ako se suočimo s odlukom koliko je bita potrebno u sustavu, postavimo dva pitanja:

(1) Koliko šuma ima u analognom signalu?

(2) Koliki iznos šuma podnosi digitalni signal?

Kada opisani model kvantizacije ne vrijedi? Samo onda **ako se greške kvantizacije ne mogu analizirati kao slučajne**. To se javlja u situacijama u kojim analogni signal poprima istu vrijednost za veliki broj uzastopnih uzoraka – **sporo promjenjivi signal**. Onda **izlaz ostaje ista digitalna vrijednost za veliki broj uzoraka** iako se analogni signal može mijenjati unutar ±1 LSB.

Umjesto da se greška kvantizacije javlja u obliku aditivnog slučajnog šum, počinje nalikovati na *učinak praga (thresholding effect)* ili harmoničko izobličenje.

b=8, $N=2^b-1$, Uvv=1V, q=Uvv/N=0,392mV; $N=q^2/12$

Zadatak: Nyquistov pretvarač, snaga signala i šuma

Nyquistov A/D pretvarač radi sa **8 bitnom** kvantizacijom, a dinamičko područje kvantizatora je od -2 do 2V.

a) odrediti ukupan broj razina kvantizatora i interval kvantizacije.

b) Kolika je maksimalna pogreška kvantizacije?

c) Kolika je snaga šuma kvantizacije (mjerena na R=1 ohm)

d) Kolika je snaga signala (sinusni) koji pobuđuju cijelo područje kvantizatora?

e) Odrediti odnos signal šum uz sinusni signal koji pobuđuje cijelo i pola područja kvantizatora.

f) Koliki je tok podataka ako se uzorkovanje izvodi sa 44,1 kHz?

Rješenje:

a) Broj bitova b=8, broj razina $N=2^b-1$, Uvv=4V, q=Uvv/N=15.68mV;

b) Max. pogreška kvantizacije je q/2;

c) $N=q^2/(12*R)=2.0505*10^{-5}W$

d) $S_1=A^2/(2*R)=2W$, $S_{1/2}=(A/2)^2/(2*R)=0.5W$,

e) A=2V maksimalna amplituda sinusa

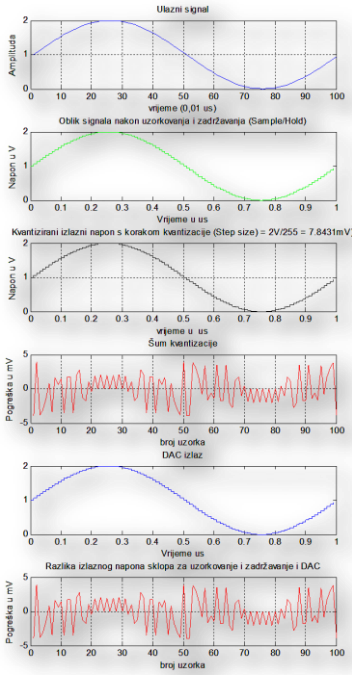
f) $S/N[dB]=10*\log_{10}(S/N)$;

g) $R=44100s^{-1}*8bit=352,8kbits/s$

Ozren Bilan

Simulacija 8 bitnog AD i DA pretvarača

```
% simulacija 8 bitnog ADC i DAC

% Ulazni signal
f = 10^6; % 10MHz ulazni signal
Fs = 100*f; % 1000MHz f. sempliranja
amp = 1; % amplituda ulaznog signala
Dynamic_Range = 2; % dinamičko područje kvantizera
DC_Offset = Dynamic_Range / 2;
t = 0:1/Fs:1/f ;

subplot(6,1,1);
sig=DC_Offset + amp*sin(2*pi*f*t);
plot(sig); % Grafički prikazujemo pomaknuti ulazni signal kako bi dobili samo pozitivne napone
grid;
xlabel('vrijeme (0,01 us)');
ylabel('Amplituda');
title('Ulazni signal');

% SAMPLE/HOLD
t = 0:1/Fs:1/f ;
SH_op = DC_Offset +
amp*sin(2*pi*f*t); % dodajemo offset kako bi dobili samo pozitivne napone

% A/D pretvorba

% Kvantizacija

No_rep_levels = 256; % 2^8 = 256 razina
Stepsize =
Dynamic_Range/(No_rep_levels - 1); % korak kvantizacije = 2V/255
Rep_levels = [];
for i = 0:1:No_rep_levels - 1
Rep_levels = [Rep_levels i*Stepsize]; % 256 razina udaljenih za korak kvantizacije "Stepsize"
end
Quantized_op = [];
for i = 1:1:length(SH_op)
[minValue,minIndex] = min(abs(SH_op(i) - Rep_levels)); % Odredi razinu najbližu svakom SH_op
Quantized_op = [Quantized_op Rep_levels(minIndex)]; % Kvantiziraj na tu vrijednost
end

subplot(6,1,2);
stairs(t*10^6,SH_op, 'g');
xlabel('Vrijeme u us');
ylabel('Napon u V');
title('Oblik signala nakon uzorkovanja i zadržavanja (Sample/Hold)');

subplot(6,1,3);
stairs(t*10^6,Quantized_op, 'k');
grid;
xlabel('vrijeme u  us');
ylabel('Napon u V');
title(['Kvantizirani izlazni napon (Korak kvantizacije (Step size) = ',num2str(Dynamic_Range),'/255 = ',num2str(Stepsize*10^3), 'mV')]);

% pogreška kvantizacije
Quantization_Noise = Quantized_op - SH_op;

subplot(6,1,4);
plot(Quantization_Noise*10^3, 'r');
grid;
xlabel('broj uzorka');
ylabel('Pogreška u mV');
title('Šum kvantizacije');
Q_Noise_power = var(Quantization_Noise);
% Simulirano
Q_Noise_power_th = Stepsize/12;
% Teoretski

%_Generiranje digitalnog koda
ADC_op = [];
ADC_op = [ADC_op dec2bin(Quantized_op / Stepsize)];
% Kraj AD pretvaraća

%D/A pretvorba
DAC_op = [];
for i = 1:1:length(ADC_op)
b7 = str2num(ADC_op(i,1))*128;
b6 = str2num(ADC_op(i,2))*64;
b5 = str2num(ADC_op(i,3))*32;
b4 = str2num(ADC_op(i,4))*16;
b3 = str2num(ADC_op(i,5))*8;
b3 = str2num(ADC_op(i,5))*8;
b2 = str2num(ADC_op(i,6))*4;
b1 = str2num(ADC_op(i,7))*2;
b0 = str2num(ADC_op(i,8))*1;
D_to_A = (b7 + b6 + b5 + b4 + b3 + b2 + b1 + b0) * (Dynamic_Range / 255);
DAC_op = [DAC_op D_to_A];
end

subplot(6,1,5);
stairs(t*10^6,DAC_op);
xlabel('Vrijeme us');
title('DAC izlaz');

subplot(6,1,6);
plot((DAC_op - SH_op)*10^3, 'r');
grid;
xlabel('broj uzorka');
ylabel('Pogreška u mV');
title('Razlika izlaznog napona sklopa za uzorkovanje i zadržavanje i DAC');
% Kraj simulacije 8 bitnog ADC i DAC
```



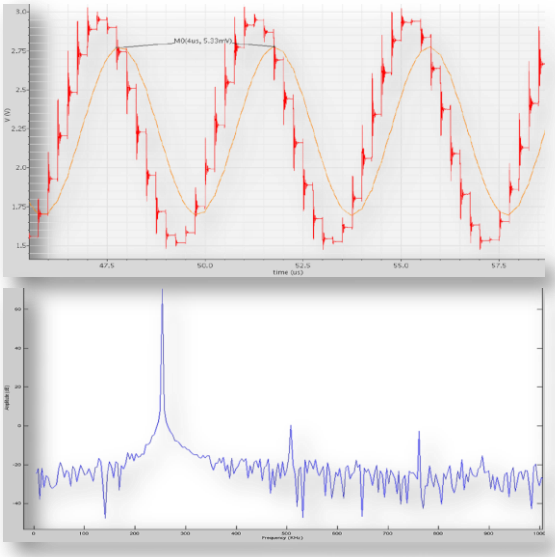
Ozren Bilan

Stvarni 8 bitni AD i DA pretvarač

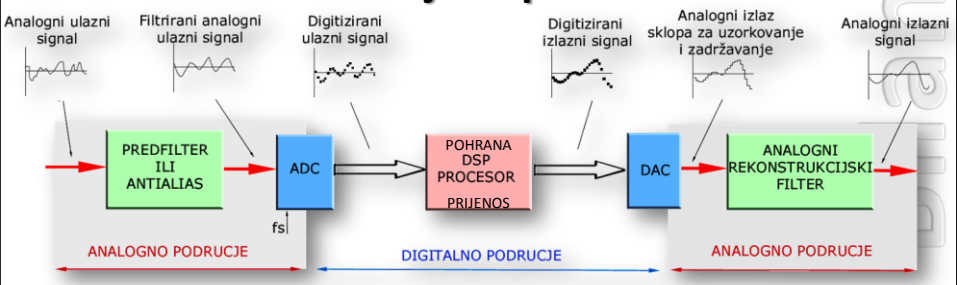
Sinusni izlaz DAC sa ulaznim sinusnim valnim oblikom od 250KHz

Izlazni signal i prikaz spektra za izlazni sinusni signal frekvencije od 250KHz.

Izlaz DAC se propušta kroz *niskopropusni filter* kako bi se izgladili rubovi signala.



Prikažimo cijeli opisani sustav



Prije A/D pretvarača, ulazni signal se propušta kroz aktivni niskopropusni filtar kako bi se eliminirale sve frekvencije iznad Nyquistove frekvencije (polovina iznosa uzorkovanja). Cilj postupka je sprječavanje aliasa za vrijeme uzorkovanja pa se naziva antialias filtar.

Kada se signal nađe u digitalnom području možemo ga lako pohraniti, prenijeti ili obraditi.

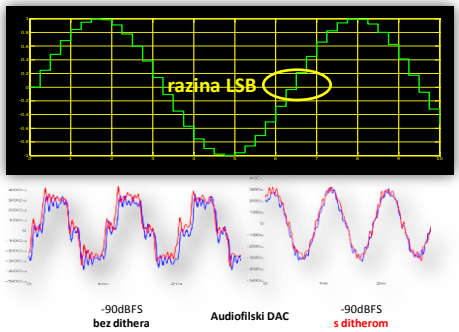
Na drugoj strani digitizirani signal propušta se kroz digitalno-analogni pretvarač i još jedan nisko propusni filtar podešen na Nyquistovu frekvenciju. Ovaj izlazni filtar naziva se rekonstrukcijski filtar.

U ovom jednostavnom modelu nastaju ozbiljni problemi jer tehnička ograničenja elektroničkih filtara mogu biti loša kao i problemi koje trebaju spriječiti. Koji su to problemi?

Dither šum

Drugi problem nastaje pri kvantiziranju razina koje su toliko niske da su reda veličine 1 bita ili razini šuma analognog sustava. Na 1-bitnim razinama greška kvantizacije postaje izlazni signal tj. 100% izobličenje. Taj problem rješava se dodavanje ***dither*** šuma signalu. Intelektualno je teško pojmiti kako se šum i greške mogu smanjiti dodavanjem šuma signalu, ali sustav funkcionira. Temelj ove pojave je svojstvo ljudskog sustava sluha koji posjeduje osobinu kojom detektira signal maskiran širokopojasnim šumom. Tako ljudski sustav sluha lako određuje *srednjetonski* govorni signal u šumu koji je više razine.

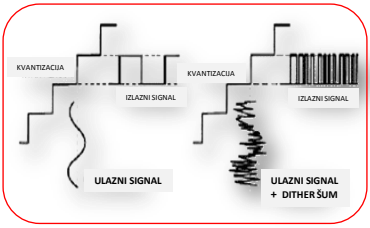
Kad na ulazu u A/D pretvarač imamo vrlo mali sinusni signal amplitude jedne do dvije kvantizacijske razine, na izlazu dobivamo gotovo pravokutni signal. Složenim signalima tako nastaju izobličenja. Pojavu sprječavamo dodavanjem slučajnog šuma male amplitude s *Gauss*-ovom ili drugom gustoćom raspodjele čija je efektivna vrijednost jednaka kvantizacijskoj razini pretvarača. Postupak linearizira prijenosnu karakteristiku A/D pretvarača koja je u području od nule do prve kvantizacijske razine nelinearna. ***Dither*** **omogućava i kvantizacija signala manje amplitude od intervala kvantiziranja LSB**.



Kako kvantizirati signal u intervalu manjem od intervala kvantizacije ?

Maksimalni izlazni signal standardnog DAC pretvarača je 2 V efektivno pri punoj modulaciji. Signal 2 V_{eff} ima vrijednost +2.828 V i -2.828 V od vrha do vrha. Dakle dinamičko područje maksimalnog signala= 5.656V. Pri tome, za razliku od analognog signala koji može poprimiti bilo koju vrijednost, digitalnim sustavima točno je definirana minimalna vrijednost. Sve ispod te vrijednosti je **digitalna nula**. 16 bitova daje 2^{16}=65536 razina kvantizacije. Podijelimo li 5.656V s 65536 razina dobijemo interval kvantizacije koji je ujedno jednak najnižoj razini 86.3 mV. Ne postoji napon manji od toga. Najniža razina **LSB** iznosi 86.3 mV koja **ili postoji ili ne postoji**. Nastaje problem: kako *modulirati napon manji od najmanje razine*. Odgovor: **dither** šumom. Ako kvantiziramo signal napona 86 mV, LSB okida pa mu dodamo napon šuma jednak dijelu razine LSB, npr. 43.15 mV, dither će promijeniti slučajnog okidanje. Umjesto da LSB bude 50% vremena uključen, a 50% vremena isključen, sada će biti 75% vremena uključen, a 25% vremena isključen. Dakle zbog dithera, intenzitet signala prenosi se u postotak vremena u kojemu će LSB biti uključen ili isključen. Dakle, LPCM **za prikaz signala manjeg od LSB koristi *oblik impulsno širinske modulacije***. Na niskim razinama, umjesto digitalne nule zvučne datoteke imaju slučajni šum na dnu dinamičkog područja pa digitalni signali zvuče kao analogni. Postoje razni tipovi dither šuma, a najbolja je *trokutna funkcija.*

Analiza zanemaruje da sustavi koji koriste dither **moraju koristiti usrednjavanje** kako bi ispravno funkcionirali. Pri mjerenjima S/Š najčešće se signal usrednjava 1 s, odnosno 44100 uzoraka. Pri tome se *ne napominje* da ljudsko uho usrednjava od 10 ms do 50 ms. Ljudskom sustavu sluha 1 s je ekstremno dugi interval. 10 ms odgovara 441 uzorku pri frekvenciji sempliranja CD, a 50 ms odgovara trajanju 2205 uzoraka. **To je razlog zašto više frekvencije sempliranja 96, 192, 385 kHz mogu zvučati bolje**; jednostavno, signal se više usrednjavanja, pa je modulacija učinkovitija.



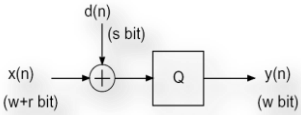
Sa stajališta ljudskog sluha učinkovitije usrednjavanje šuma daje manja izobličenja. Taj aspekt dithera otkriven je krajem osamdesetih, uvođenjem digitalnih stolova za miješanje na profesionalno tržište. Snižavanjem razine stereo signal se pretvarao u mono. Kako bi snizili cijenu digitalne konzole koristio se isti generator slučajnih brojeva za sve kanale. Kada su svi kanali dobili **nezavisni generator dither šuma** problem je nestao. Osim pri miksanju nezavisnih kanala, *dither* je potreban i pri *nesinkroniziranoj rekvantizaciji*, *digitalnoj ekvalizaciji* i pri implementaciji bilo kojeg efekta. Općenito vrijedi:

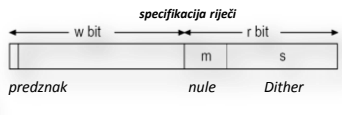
ako se signalu dodaje dither, on mora biti neovisan za svaki kanal.

Rekvantizacija

Pri pohrani zvučnih datoteka, promjene formata i algoritama digitalne obrade vrlo često je potrebno **kvantiziranu datoteku ponovno kvantizirati**. Taj postupak nazivamo **rekvantizacija**. Pri tome postupku vrlo male razine signala dovode do pogrešaka koje ovise o ulaznom signalu. Uvjeti modela kvantizacije tada više ne vrijede. Za smanjenje učinaka signala male razine potrebno je linearizirati krivulju kvantizera sumiranjem slučajne sekvence ***d(n)*** kvantiziranom signalu ***x(n)*** prije kvantizacijskog procesa. Tri su tipa slučajnog signala **dithera** koje možemo prikazati izrazima:

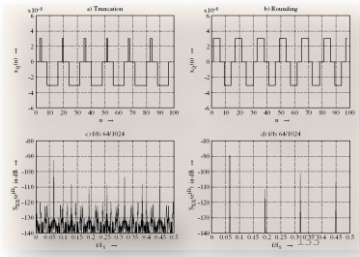
$$d_{\mathrm{RECT}}(n) = d_1(n),$$
$$d_{\mathrm{TRI}}(n) = d_1(n) + d_2(n),$$
$$d_{\mathrm{HP}}(n) = d_1(n) - d_1(n-1)$$





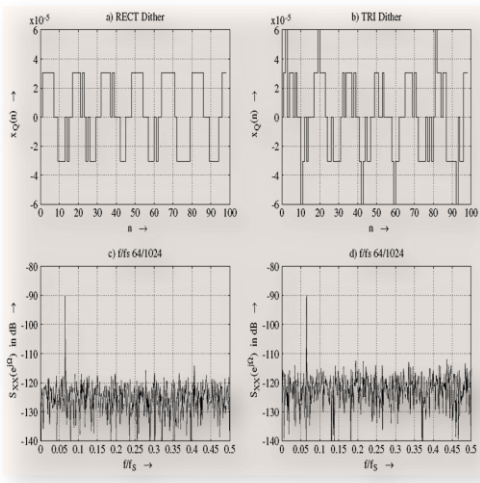
Pri tome se ne postiže statistička neovisnost greške ulaznog signala ali se djeluje na uvjete nastajanja greške. Slučajna sekvenca ***d(n)*** dobiva se **generatorom slučajnih brojeva** s **jednolikom funkcijom gustoće vjerojatnosti** (PDF). Postoji više tipova dither šuma: *RECT* **kvadratni**, *TRI* **trokutasti** i *HP* **trokutasti filtriran visokim propustom**. Za generiranje **trokutne PDF** slučajne sekvence potrebno je sumirati dvije neovisne jednolike slučajne sekvence $d_1(n)$ i $d_2(n)$. Za generiranje **trokutastog dithera filtriranog visokim propustom** vrijednosti dithera $d_1(n)$ dodaje se $-d_1(n-1)$, pa je potreban samo jedan generator slučajnih brojeva.

Područje amplituda jednolikog PDF dithera nalazi se između ±*Q*/2, a trokutastog unutar ±*Q*. Trokutasti PDF dither ima dvostruku ukupnu snagu šuma. Učinak ulazne amplitude kvantizera prikazan je na donjoj slici za 16-bitni kvantizer (Q=2^{-15}). Kvantizirani sinusoidalni signal amplitude 2^{-15} (1-bitna amplituda). Kako bi prikazali modulaciju šuma PDF jednolikog dithera, ulazna amplituda smanjena je na $A = 2^{-18}$, a frekvencija je određena na f/fS = 14/1024. To znači da je ulazna amplituda u kvantizer 0.25 bita.



Izlaz kvantizera bez dodanog dithera jednak je nuli.

Dodamo li RECT dither, kvantizirani signal prikazuje dijagram ***a***, a prikazan je i ulazni signal amplitude 0.25*Q*. Gustoća spektra snage kvantiziranog signala prikazana je na slici ***c***. U vremenskom području primjećujemo korelaciju amplituda ulaznog i izlaznog signala. Primjenom subjektivnih slušnih testova otkriveno je da se modulacija šumom javlja ako je ulazna amplituda jednoliko slabi i pada ispod amplitude koraka kvantizacije. Taj proces nastaje pri svim procesima slabljenja govornih i glazbenih signala. Kako bi se ova modulacija šumom smanjila pri niskim razinama signala, koristi se trokutasti PDF dither. Slika ***b*** prikazuje kvantizirani signal, a slika ***d*** prikazuje gustoću spektra snage. Može se primjetiti kako kvantizirani signal ima nepravilan oblik pa ne nastaje korelacija.



Pri subjektivnim slušnim testovima trokutasti PDF dither rezultira konstantnim pragom šuma pa i onda kada ulazna razina signala padne na nulu – digitalni zvučni signali **zvuče kao analogni**.

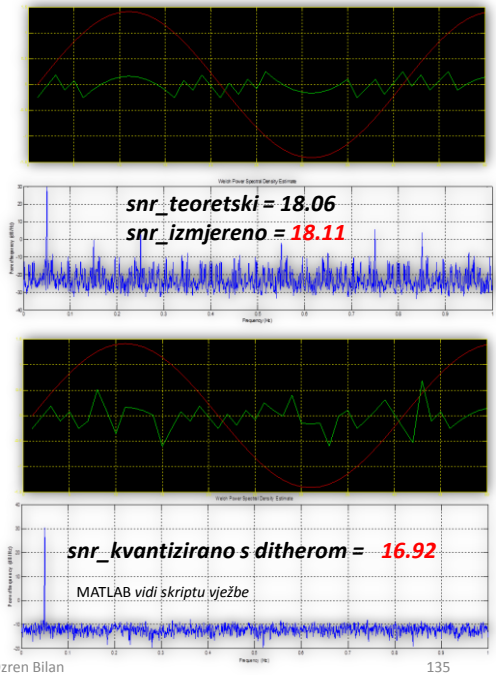
To ćemo pokazati u MATLAB-u:

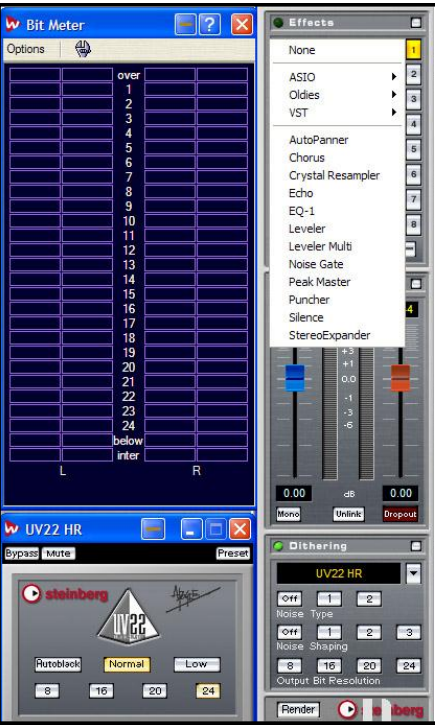
Generirat ćemo sinusni signal fundamentalne frekvencije f_0=200 Hz, sempliran *Fs*=8000 Hz, a varijancu ćemo odrediti na 1 ili 0 dB, tako što ćemo postaviti vršne vrijednosti na 2.

Kad **jednoliko kvantiziramo s 3 bita, na 8 razina** u području od [-2,+2], primjećujemo kako greška kvantizacije (*zeleno*) ne nalikuje na stvarni šum jer vidimo da se **u valnom obliku signala greške pojavljuje periodičnost i harmonici izvornog signala** pa je reproducirani zvuk vrlo neugodan.

U takvoj situaciji najbolje je postojeći šum kvantizacije *pobijeliti*. To se izvodi dodavanjem realnog šuma signalu. Dobivena **greška kvantizacije (*zeleno*) sada sliči slučajnom šumu**.

Gustoća spektra snage kvantiziranog signala nalikuje na bijeli šum dodan izvornom sinusnom signalu. Odnos signal šum je **malo** pogoršan ali **percipirani zvuk je daleko bolji**. **U tome je značaj dithera jer je omogućio da digitalni sustavi zvuče kao analogni**.



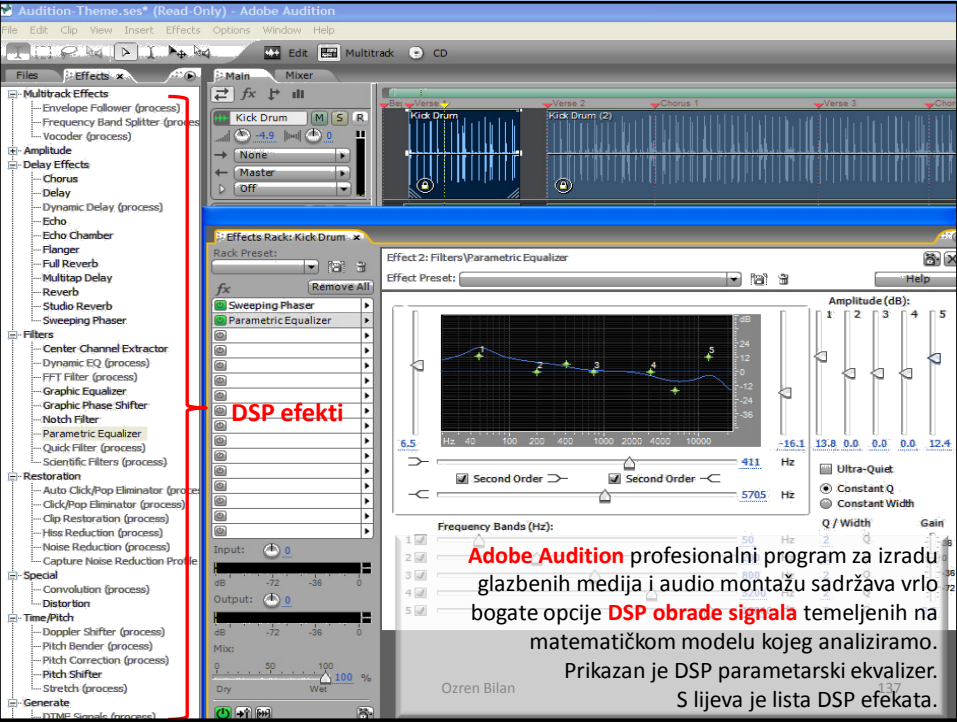


Dither šum

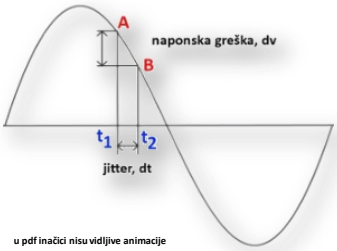
Steinberg Wave Lab 6 profesionalni program za autorizaciju CD i DVD-Audio medija, sadržava vrlo bogate opcije **DSP obrade signala** temeljenih na matematičkom modelu kojeg analiziramo. Pored toga sadržava i **tri opcije dodavanja dither šuma** pri rekvantizaciji glazbenih zapisa (**Intern, UV22, UV22HR**). Svaki od primijenjenih modela šuma ima posebne pogodnosti.

S lijeve strane prikazan je **bit metar** koji nam pokazuje stvarnu kvantizaciju signala, tj. stvarni broj korištenih bitova u trenutnom signalu. Često za datoteke, koje se specificiraju kao 24 bitne, mjerenjem možemo utvrditi da je riječ o 16 bitnim.

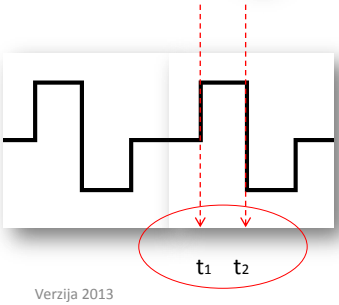
Proizvođači HD glazbe datotekama u rezoluciji 16 bita često umeću 8 nula i distribuiraju ih kao 24 bitne datoteke. Bit metar to jasno pokazuje.



Jitter



u pdf inačici nisu vidljive animacije



Treći problem koji može nastati je situacija u kojoj **diskretni uzorci signala promjene svoj vremenski položaj za neki vrlo mali iznos**. Stabilnost takta digitalnog audio sustava ima najveći utjecaj na kakvoću zvuka, a pojava nestabilnosti naziva se ***jitter***. Rješenje je vrlo stabilan generator takta ili *PLL*. Iznosi pogreške mogu biti vrlo mali ali i takvi ipak imaju utjecaj na rad sustava. Ako iznos *jittera* prijeđe neku granicu sustav neće moći točno dekodirati signalne bitove.

Vremenski jitter definira se kao odstupanje signala od svog idealnog vremenskog položaja. **Ukupni *jitter* (*TJ*)** tvore **deterministički *jitter* (*DJ*)** i **slučajni *jitter* (*RJ*)**. Pri tome se *DJ* sastoji od nekoliko komponenti koji uključuju elektromagnetske interferencije, preslušavanje, ograničenja frekvencijske širine pojasa itd. *RJ* posljedica je izvora šuma komponenti kao što je termički šum i *šum sačme*. Relativno je jednostavno izmjeriti svaku komponentu *jittera* ali nije lako izmjeriti i analizirati simultane višestruke komponente.

U *Matlabu* možemo simulirati sinusoidu s efektivnim iznosom ukupnog *jittera* po volji.

Učinak sempliranja na DAC

Na slici je DAC kojemu je narinut digitalni ulaz ***Vin(n)*** uz takt sempliranja ***CLK***. U idealnom slučaju takt sempliranja *CLK* ima period ***T***. U stvarnosti trajanje perioda se mijenja što nazivamo *jitter*. Označimo ga s ***εn*** pa će *n*-to vrijeme sempliranja CLK biti ***nTs+ εn*** umjesto ***nTs***. Zato jer je jitter ***εn*** mnogo manji od perioda ***Ts*** podrazumijevamo da je:

$$-\frac{T_s}{2} < \epsilon_n < \frac{T_s}{2}$$

Podrazumijevamo da je rezolucija dobra pa zanemarujemo učinke kvantizacije. Slika pokazuje idealni DAC izlaz bez jittera, a slika desno DAC izlaz s *jitterom*.

DAC izlazna snaga greške zbog *jittera* takta sempliranja određena je izrazom za ***Pe***.

εn predstavlja izlaznu grešku DAC zbog jittera. Ako ne postoji korelacija ulaznog signala ***Vin(t)*** i *jittera* sempliranja, a to je najčešći slučaj, ***εn*** i ***εn*** su neovisni pa je ***Pe***:

$$P_e := \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} e_n^2 |\epsilon_n| = E\{e_n^2 |\epsilon_n|\}$$

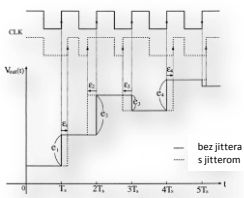
Ako je ulazni signal u DAC *V* (*n*) kosinusni valni oblik

$$V_{in}(n) = A\cos\left(2\pi \frac{f_{in}}{f_s} n\right)$$

snaga greške jittera određena je

$$P_e = 2A^2 \sin^2\left(\pi \frac{f_{in}}{f_s} n\right) E\{|\epsilon|\}$$

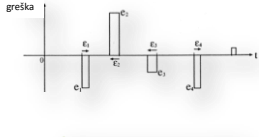
f je ulazna frekvencija, a ***fs*** je frekvencija sempliranja ***f=1/Ts***).

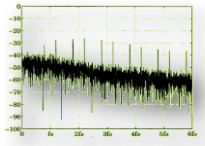


Simulacija spektra snage izlazne greške DAC zbog *jittera* generatora takta sempliranja s jednolikom raspodjelom pokazuje vršne vrijednosti

$$kf_s \pm f_{in} \quad (k = 1, 2, 3, \ldots)$$

MATLAB *vidi skriptu vježbe*





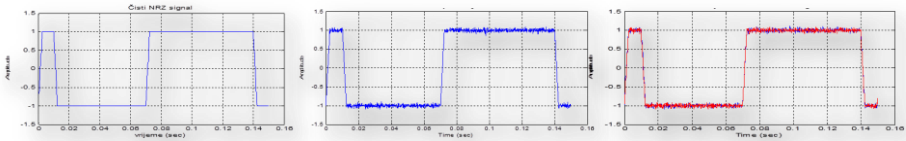
Simulacija spektra snage greške izlaznog signala DAC-a zbog jittera

Opisali smo **učinak takta sempliranja na DAC** i izveli smo izlaznu snagu greške zbog *jittera* takta sempliranja te simulirali karakteristiku spektra. Ispitali smo i učinak na odnos S/Š primjenom numeričke simulacije te smo zaključili:

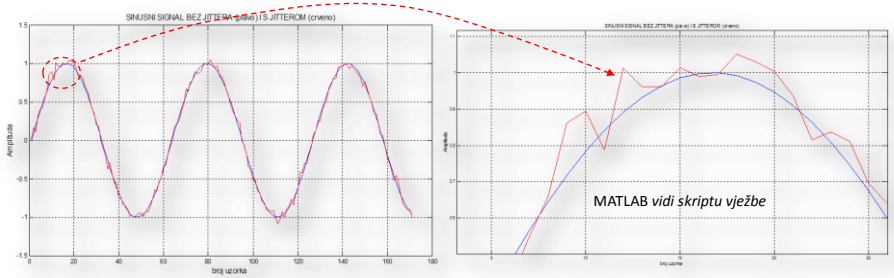
- Uzmemo li u obzir ukupnu **snagu šuma izvan i unutar frekvencijskog pojasa signala**, **odnos signal/šum DAC ostaje približno konstantan bez obzira na jitter** pri sempliranju. Ovaj dokaz i zapažanje znači da će 1-bit DSD dati **kvalitetniji signal bez filtera**.
- Međutim, **ako nakon DAC slijedi analogni niskopropusni filter, a uzimamo u obzir samo snagu šuma unutar frekvencijske širine pojasa signala**, **odnos signal/šum slabi porastom frekvencija signala**. Dakle, **jitter takta sempliranja postaje ozbiljan problem za vrlo brze DAC-ove**.

To objašnjava da će npr. 96kHz PCM DAC imati manju grešku *(bolji zvuk ?)* od 192kHz DAC-a iste stabilnosti takta.

Pretpostavimo da smo u *Matlabu* simulirali određeni čisti diskretni signal određenog vremena porasta i pada. Dobit ćemo grafički prikaz. U realnom sustavu signalu će se dodati ***šum.*** Dodaje se i ***jitter***, predstavljen je razlikom plavog i crvenog valnog oblika na slici dolje. Na slici izgleda da *jitter* nije toliko izražen. Povećamo li označeni dio slike izmjerit ćemo da je u simulaciji pri prolazu kroz nulu razlika točnog položaja signala i pogreške zbog *jittera* oko *80 μs*.



Slika prikazuje kako izgleda rekonstruirani sinusoidalni signal s tipičnim *jitterom,* što možemo poslušati

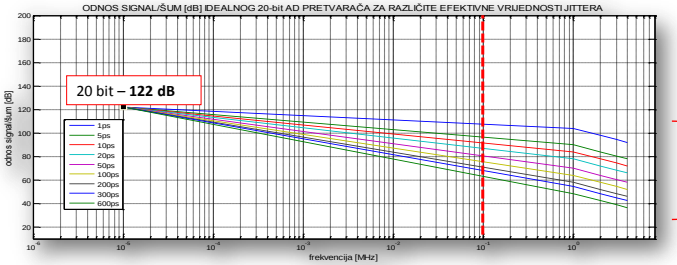


Izgled rekonstruiranog sinusoidalnog signala s jitterom - desno je povećani detalj

Matlab programom možemo simulirati i odrediti utjecaj ukupnog *jittera* npr. od *1ps* do *600ps* na **odnos signal/šum idealnog analogno digitalnog pretvarača** u ovisnosti o broju bitova. Tipične vrijednosti *jittera* koje se javljaju u kvalitetnim audio uređajima ne bi smjele prijeći vrijednosti od *10-15 ps*, a **napredni D/A konvertori** posebnim izvedbama postižu stabilnost *jittera* u području ***1 ps***.

U *Matlabu* možemo simulirati pojavu *jittera* na AD pretvaraču sa zadanim brojem bitova i odrediti utjecaj na teoretski odnos signal šum pretvarača. Teorija pokazuje da je odnos signal šum digitalnog sustava: *S/Š= 6.02 BROJ BITA + 1.76 [dB]*. Pogledajmo kako će *jitter* pogoršati odnose. Na apscisi je odnos signal šum [dB], a na ordinati frekvencija gdje okomita linija označava 100kHz.

Iz oblika dijagrama primjećujemo da **niskofrekvencijskim signalima u odnosu signal šum dominira greška kvantizacije**, a **za visokofrekvencijske signale dominira greška zbog *jittera***. Očito je da će **pretvarači s više bitova biti osjetljiviju na grešku *jittera*** jer je im je interval među uzorcima sve kraći. *Jitter* od 300ps na 10 kHz slabi odnos signal šum 20 bitnom pretvaraču skoro za 40 dB. Dijagram simuliramo za sve formate.



MATLAB *vidi skriptu vježbe*

Porastom jittera slabi odnos signal/šum

Ako je greška, kao u našem prethodnom teoretskom slučaju, prevelika, sustav praktički postaje neupotrebljiv jer mu je odnos signal šum na visokim frekvencijama postaje 0. To znači da će zbog *jittera*, iznos šuma biti približno jednak iznosu signala.

Dokazali smo da je pretvarač komponenta digitalnog sustava koja je najranjivija *jitterom*. Ako DAC generira takt cjelokupnog sustava ili ako rad svih komponenata sustava ovisi o vrlo stabilnom i vrlo točnom generatoru takta, DAC može u potpunosti ignorirati *jitter* izvora.

Audio sustavima temeljenim na računalu ili poslužitelju najveći problem je pravilan takt jer se **generator takta u računalu ne projektira za reprodukciju glazbe**.

Valni oblik digitalnog audio signala predstavljen je serijom uzoraka opisanih binarnim znamenkama. Svaki uzorak predstavlja točno određenu amplitudu u točno određenom vremenskom trenutku. Ako se bilo koji, od ta dva podatka, poremeti za neki vrlo mali iznos na izlazu ćemo dobiti izobličenje i šum. Zadržati točnu vrijednost amplitude najvećem broju uređaja ne predstavlja problem. Međutim, veliki je problem zadržati određenost ili pravodobnost vremenskog trenutka svakog uzorka.

DAC mora raditi na potpuno istom taktu kao i računalo izvora signala kako se međuspremnici podataka ne bi prepunili ili ispraznili. To je zahtjev koji se teže kvantificira i rješava. Računala temeljena na USB sučelju s vlastitim kodom predstavljaju povoljnija rješenja od SPDIF sučelja jer eliminiraju vremensku povezanost komponenti izvora i vanjskog DA pretvarača. Nove, kvalitetno izrađene i točno primijenjene, audio komponente u kombinaciji s točno konfiguriranim računalom imaju mogućnost davanja vrlo kvalitetnog zvuka koji može biti jednak ili bolji od postojećih postupaka reprodukcije.

Višetaktni *multirate* sustavi za obradu signala

Ciljevi

- Objasniti što je višetaktna obrada i kojim se postupcima i sklopovima realizira
- Opisati problem *aliasa* i rješenje primjenom preuzorkovanja (*oversampling*).
- Prikazati pojedinosti digitalnog aliasa i spektralnih slika.
- Prikazati postupak decimacije (*downsampling*) i interpolacije naduzorkovanjem (*upsampling*) pri promjeni efektivne frekvencije sempliranja digitalnog signala.
- Pokazati kako se interpolacija realizira primjenom naduzorkovanja (*upsampling*) i niskopropusnog filtera s prilagođenjem pojačanja.
- Prikazati postupak promjene frekvencije sempliranja kaskadom interpolacije i decimacije.
- Izvesti odnos frekvencije sempliranja i spektralne gustoće slučajnog šuma.
- Pokazati eliminiranje šuma u signalu primjenom preuzorkovanja (*over-sampling*).
- Opisati postupak *delta-sigma* kvantizacije i izvesti prijenosnu funkciju kvantizera za oblikovanje šuma prvog reda.
- Prikazati postupak *delta-sigma* kvantizacije i izdvajanja kvantiziranog signala niskopropusnim filtriranjem.

Ozren Bilan

Što je višetaktna obrada signala?

Višetaktno: opisuje postupak obrade koja primjenjuje više različitih frekvencija sempliranja. **Višetaktni sustavi** su **diskretni sustavi** koji koriste više od jedne frekvencije sempliranja, a temeljni postupci koji se primjenjuju su **naduzorkovanje** *upsampling* i **poduzorkovanje** *downsampling.*

Gdje se višetaktna obrada primjenjuje?

Pri pretvorbi frekvencije sempliranja zvučnih signala i pri algoritmima kodiranja i sažimanja zvučnih datoteka. Obrada zvučnih signala često zahtjeva **pretvorbu frekvencije sempliranja**.

Pretvorba frekvencije sempliranja naziva se **resempliranje** pri čemu se primjenjuju dva osnovna sklopa višetaktnih sustava:

- sklop za **naduzorkovanje** *upsampler*
- sklop za **poduzorkovanje** *downsampler*.

Opisat ćemo **naduzorkovanje** *upsampling* i **poduzorkovanje** *downsampling* u vremenskom i frekvencijskom području.

Pored toga uvode se sklopovi

- ***interpolator*** pri naduzorkovanju
- ***decimator*** pri poduzorkovanju

Ti sustavi sastoje se od filtera u kombinaciji sklopom za **naduzorkovanje** *upsampling* ili **poduzorkovanje** *downsampling*.

Pojmovi višetaktne obrade ne smiju se miješati sa pojmovima **sempliranja analognog signala** frekvencijom višom od Nyquistove – preuzorkovanje i nižom od Nyquistove – poduzorkovanje.

Temeljne operacije višetaktne obrade

Temeljne operacije višetaktne obrade za promjenu frekvencije sempliranja su:

- **naduzorkovanje** povećava frekvenciju sempliranja za cjelobrojni faktor *L*
- **poduzorkovanje** smanjuje frekvencije sempliranja za cjelobrojni faktor M

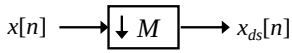
Opis **naduzorkovanja** *upsampling* u vremenskom području

Operacijom naduzorkovanja, označeno s cjelobrojnim faktorom *L*, generira se izlazni niz čija je frekvencija sempliranja ***L* puta viša** od frekvencije sempliranja ulaznog signala *x*[*n*]. Dijagram je

operacija naduzorkovanja realizira se **umetanjem** *L-1* jednoliko razmaknutih nultih uzoraka između dva susjedna uzorka *x*[*n*]. U praksi se, pri postupku naduzorkovanja, uzorci vrijednosti nula zamjenjuju s odgovarajućim uzorcima različitim od nule; ***interpolacijom***, a što se realizira postupkom filtriranja. Primjer; niz *1 1 1*, naduzorkujemo četverostruko (*L=4*) umetanjem *L-1* nule: *100010001000*.

Opis **poduzorkovanja** *downsampling* u vremenskom području

Operacijom poduzorkovanja, označenom s cjelobrojnim faktorom *M*, generira se izlazni niz čija je frekvencija sempliranja ***M* puta niža** od frekvencije sempliranja ulaznog signala x[n]. Dijagram je



Operacija *poduzorkovanja* realizira se zadržavanjem svakog *M*-tog uzorka *x*[*n*] te eliminiramo preostalih *M*-1 uzoraka između njih, postupkom ***decimacije***. Primjer; niz *1 0 1 0 1*, decimiramo: *1 1 1*.

Problem *aliasa*

- Ako signal ima **frekvencijsku širinu pojasa** *B[Hz]*, tada je prema *Nyquistu* minimalna frekvencija sempliranja *2B[Hz]*.
- U cilju sprječavanja aliasa, prije sempliranja, signal moramo filtrirati na frekvencijsku širinu pojasa $f_s/2$.
- Ako signal sadržava frekvencijske komponente ili šum iznad *B[Hz]*, pri sempliranju *Nyquistovom* frekvencijom *2B[Hz]*, analogni niskopropusni (***antialias***) filter mora imati **nepraktično veliku strminu**.

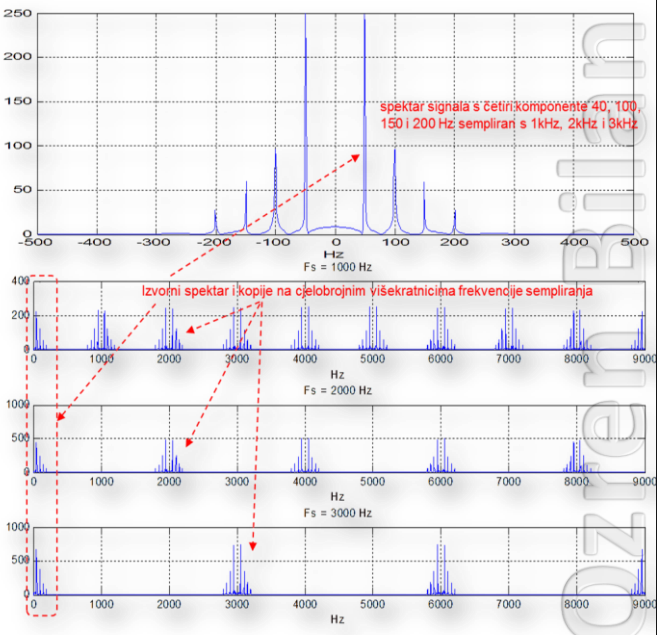
Rješenje problema: **višetaktna obrada**

Postupak:

- Signal **preuzorkujemo** mnogostruko višom frekvencijom od 2*B[Hz]* i pojasno ograničavamo sa strmim **digitalnim** niskopropusnim filterom
- Digitalno **decimiramo** signal na efektivnu frekvenciju sempliranja 2*B[Hz]* nakon digitalnog filtriranja
- To nam omogućava implementaciju vrlo jednostavnog **analognog** anti-alias filtera.

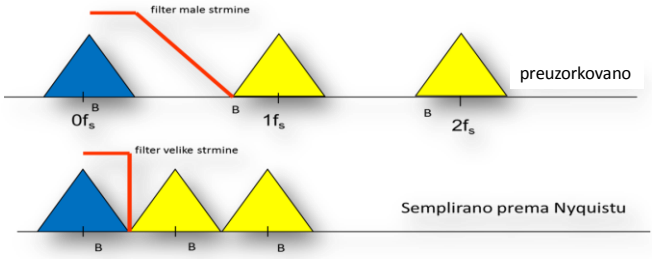
Problem *aliasa*

- Signal **preuzorkujemo** mnogostruko višom frekvencijom od 2*B[Hz]* i pojasno ograničavamo sa strmim digitalnim niskopropusnim filterom
- Digitalno **decimiramo** signal na efektivnu frekvenciju sempliranja 2*B[Hz]* nakon digitalnog filtriranja
- To nam omogućava implementaciju vrlo jednostavnog analognog *anti-alias* filtera.

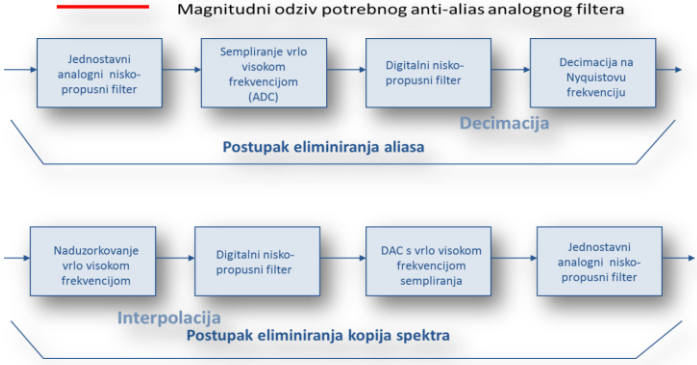


Ozren Bilan

Kako preuzorkovanje pomaže anti-alias ?



Digitalno elimininranje aliasa i kopija spektra



Poduzorkovanje i decimacija

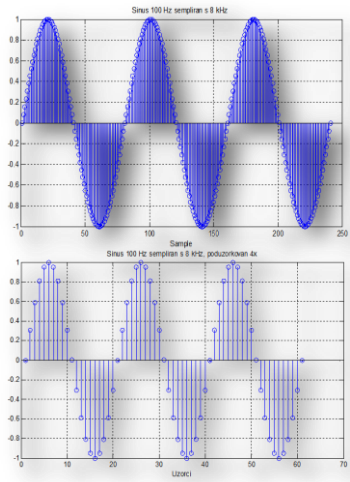
Uzorke eliminiramo postupkom *downsamplinga* - **decimacija.**

Decimacija faktorom ***N***, snižava za isti faktor efektivnu frekvenciju sempliranja, pa je nova frekvencija sempliranja jednaka f_s/N

U cilju sprječavanja aliasa, signal koji se decimira, prije eliminiranja uzoraka, ograničava na frekvenciju $f_s/2N$, niskopropusnim filterom

Niskopropusno filtriranje praćeno *downsamplingom* nazivamo **decimacija**

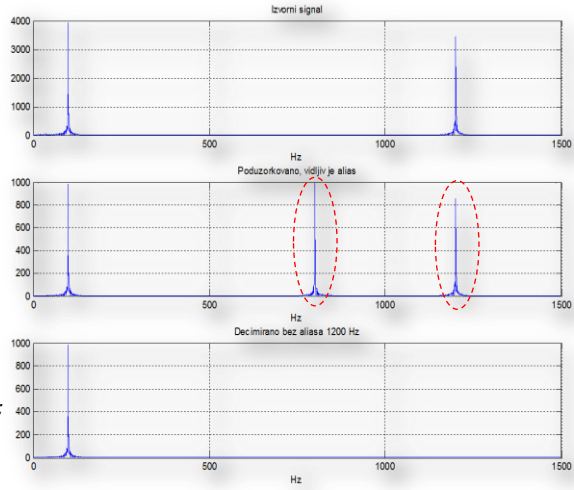
```
[ysin,tsin]=analog(100,1,30,8000);
stem(ysin)
grid
title('Sinus 100 Hz sampliran s 8
    kHz ')
xlabel('Sample')
dysin=downsample(ysin,4);
figure,
stem(dysin);
grid
title('Sinus 100 Hz sampliran s 8
    kHz, decimiran 4x')
xlabel('Uzorci')
```



Decimacija

Pri decimiranju, niskopropusno filtriranje prethodi *downsamplingu* kako bi se izbjegao alias.
Primjer: Decimiranje faktorom 4 signala 100 i 1200 Hz MATLAB naredbom ***decimate***

```
x=analog([100,1200],[1,1],1000,8000);
xd4=downsample(x,4);
% sada je frek. sempliranja 2000 Hz pa
    ne možeme pravilno semplirati
    komponentu od 1200Hz.
xdecim4=decimate(x,4);
% Niskopropusni filter propušta do
    0.8*(8000/8) = 800 Hz i neće
    propustiti komponentu 1200 Hz
subplot(3,1,1),
dtft_demof(x,0,1500,2024,8000);
grid
title('Izvorni signal')
subplot(3,1,2),
dtft_demof(xd4,0,1500,2024,2000);
grid
title('Downsampled, vidljiv je alias')
subplot(3,1,3),
dtft_demof(xdecim4,0,1500,2024,2000);
grid
title('Decimirano bez aliasa 1200 Hz')
```



Naduzorkovanje i interpolacija

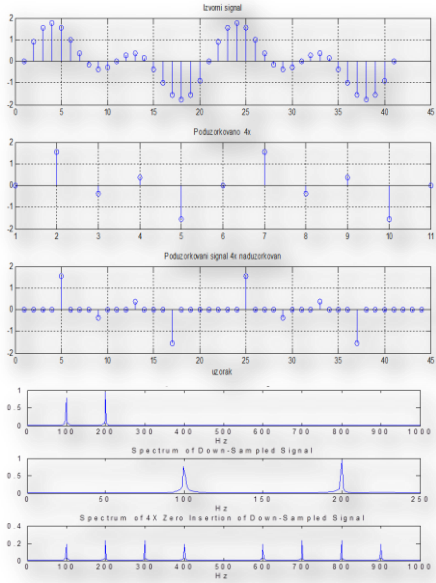
Postupak umetanja nula između postojećih uzoraka nazivamo ***naduzorkovanje***

Naduzorkovanje faktorom ***N*** povisuje frekvenciju sempliranja na Nf_s

Procjena (*estimacija*) vrijednosti između postojećih uzoraka naziva se **interpolacija**

Interpolacija se izvodi **niskopropusnim filtriranjem** naduzorkovanog signala koje prati **ispravak amplitude** kako bi se restorirala **snaga signala**

```
[x,ts]=analog([100,200],[1,1],20,2000);
% generiranje kratkog signala
xd4=downsample(x,4);
xu4=upsample(xd4,4);

subplot(3,1,1),
stem(x),
grid
title('Izvorni signal')
subplot(3,1,2),
stem(xd4),
grid
title('Downsampled 4x')
subplot(3,1,3),
stem(xu4),
grid
title('downsampled signal 4x naduzorkovan')
xlabel('uzorak')
```



Postupak interpolacije i rekonstrukcije poduzorkovanog signala interpolacijskim filtriranjem

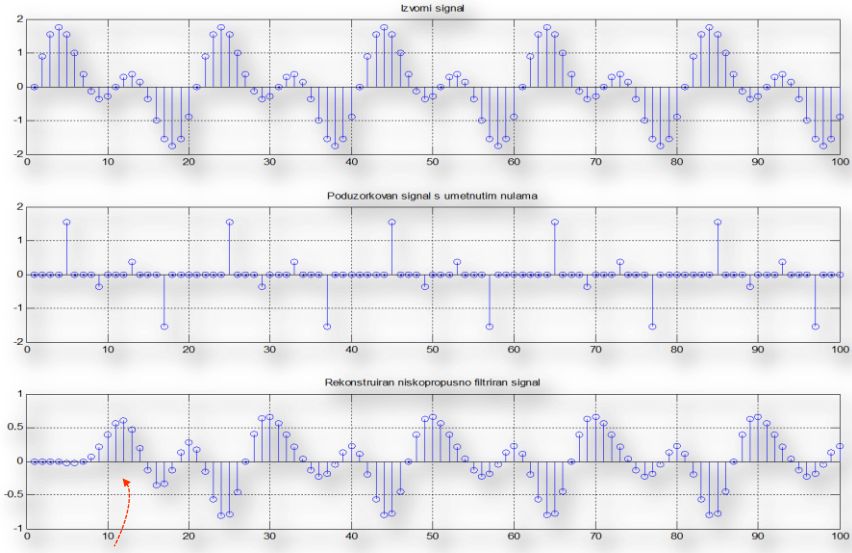
Spektar naduzorkovanog signala daje rješenje problema

- Naduzorkovani signal filtriramo niskopropusnim filterom kako bi eliminirali sve frekvencijske komponente iznad $f_s/2N$, gdje je f_s efektivna frekvencija sempliranja nakon naduzorkovanja, a *N* je faktor naduzorkovanja.
- Rezultat množimo s *N* kako bi rekonstruirali točnu snagu naduzorkovanog signala

Tipično se koristi niskopropusni filter *8N* reda. Postupak se primjenjuje MATLAB naredbom ***interp***, korištenjem posebnog **simetričnog FIR filtera** koji kompenzira grupno kašnjenje

```
% konstruiramo signal
x=analog([100,200],[1,1],1000,2000);
xd4 =downsample(x,4); % poduzorkujemo
xu4 =upsample(xd4,4); %naduzorkujemo
n=0:32;%uzimamo 33 uzorka

% Projektiramo niskopropusni filter reda 8N
fs=2000;% frekvencija sempliranja
N=4;
odreznaf=fs/(2*N);% odrezna frekvencija
omega=2*pi*odreznaf/fs;%digitalna frekvencija
% imp. odziv konstruiramo niskopropusni filter
h=(omega/pi)*sinc(omega*(n-6)/pi).*hamming(33)';
x_rekonstruiran=4*filter(h,1,xu4);
% niskopropusno filtriranje i množenje s N=4

subplot(3,1,1);
stem(x(1:100)),
grid
title('Izvorni signal')

subplot(3,1,2);
stem(xu4(1:100)),
grid
title('Poduzorkovan signal s umetnutim nulama')

subplot(3,1,3);
stem(x_rekonstruiran(1:100)),
grid
title('Rekonstruiran niskopropusno filtriran signal')
```

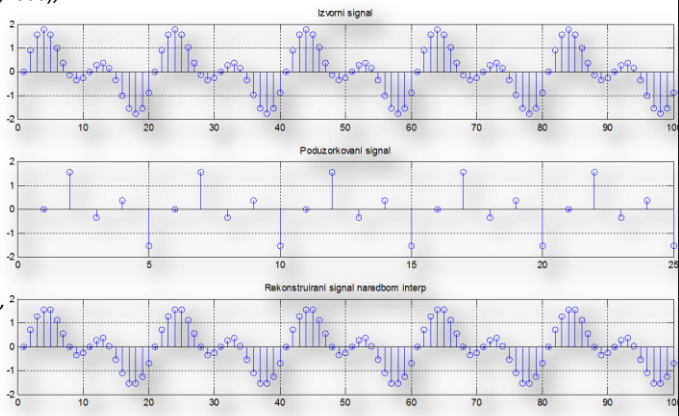
Signal nakon interpolacijske rekonstrukcije



Projektirani niskopropusni filter unosi grupno kašnjenje od oko 16 uzoraka

Interpolacija naredbom *interp* i prikaz rezultata

```
x=analog([100,200],[1,1],1000,2000);
% konstruiramo signal
xd4=downsample(x,4);
xu4=upsample(xd4,4);

x_interp=interp(xd4,4);

subplot(3,1,1),
stem(x(1:100)),
grid
title('Izvorni signal')

subplot(3,1,2),stem(xd4(1:25)),
grid
title('Poduzorkovani signal')

subplot(3,1,3),
stem(x_interp(1:100)),
grid
title('Rekonstruirani signal naredbom interp')
```



Promjena frekvencije sempliranja – *resempliranje*

Ponekad je potrebno promijeniti frekvenciju sempliranja za faktor L/M gdje je L faktor **interpolacije**, a M je faktor **decimacije**

Promjena frekvencije sempliranja na više ili na niže uvijek se izvodi redom: prvo interpolacija faktorom L pa decimacija faktorom M, kako se ne bi izgubile frekvencijske komponente važne za glazbeni signal.

Primjer: Mijenjamo glazbeni signal s frekvencije sempliranja CD 44.1 kHz na frekvenciju sempliranja uređaja za digitalno snimanje 48 kHz. Potrebno je izvršiti interpolaciju faktorom 160, a zatim decimirati faktorom 147 jer je 160/147=48/44,1

$$\frac{L}{M}=\frac{48}{44.1}=\frac{480}{441}=\frac{160}{147}$$

1. primjer *resempliranja interp* i *decimate*

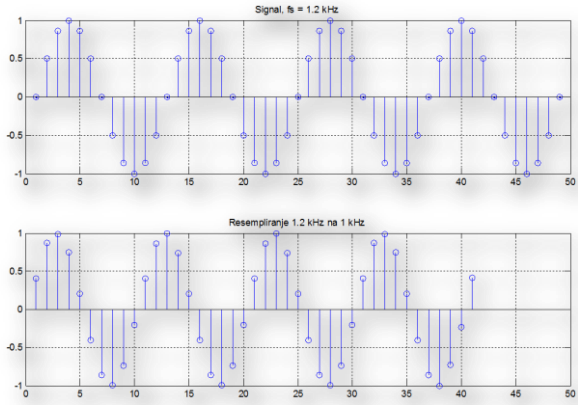
Pretvorit ćemo sinusoidalni signal 100 Hz sempliran s 1200 Hz u signal sempliran 1000 Hz.

L/M=1000/1200 = 5/6

x=analog(100,1,40,1200);
*xL=**interp**(x,5);*
*% **interp** izvodi interpolaciju*
*xLM=**decimate**(xL,6);*
*% **decimate** decimira signal*

subplot(2,1,1),
stem(x),grid
title('Signal, fs = 1.2 kHz')

subplot(2,1,2),
stem(xLM),grid
title('Resempliranje 1.2 kHz na 1 kHz')
axis([0,50,-1,1])



2. primjer resempliranja upfirdn MEX kaskada naduzorkovanja, FIR filtriranja i poduzorkovanja

Ako je potrebno promijeniti frekvenciju sempliranja datoteke s one koju koristi **DAT HD signal od 48 kHz** na frekvenciju sempliranja **CD od 44,1 kHz** pa se *fs* mijenja za faktor od 147/160 jer je 48 x 147 /160= 44,1. Na zadanu zvučnu datoteku *x* primjenjujemo naredbu koja predstavlja kaskadu tri operacije ***y =upfirdn(xin,h,p,q)*** :

1) Naduzorkuje se signal ***x*** cjelobrojnim faktorom ***p*** umetanjem nula.
2) FIR filterom filtriramo naduzorkovani signal sa sekvencom impulsnog odziva zadanog vektorom matrice ***h*** u cilju poboljšanja odziva
3) Odbacivanjem suvišnih uzoraka poduzorkujemo rezultat cjelobrojnim faktorom ***q***

upfirdn se primjenjuju se kao MEX-datoteka (binarne *MEX-datoteke su* dinamički povezani potprogrami koje MATLAB interpreter učitava i izvodi. MEX znači *MATLAB executable*) kako bi se postigla maksimalna brzina. FIR filter je niskopropusnog tipa koji se projektira funkcijom ***firpm*** ili ***fir1***.

yout* = upfirdn(*xin*,*h*)** filtrira ulazni signal ***x FIR filterom impulsnog odziva ***h***. Ako je ***x*** prikazan u retku ili kao stupčasti vektor, onda predstavlja jedan signal. Ako je ***x*** matrica, tada se svaki stupac nezavisno filtrira. Ako je ***h*** predstavljen retkom ili stupčastim vektorom, onda predstavlja jedan FIR filter. Ako je ***h*** matrica, tada svaki stupac predstavlja sekvencu posebnog FIR impulsnog odziva. Ako je ***y*** prikazan u retku ili kao stupčasti vektor, onda predstavlja jedan signal. Ova sintaksa ne izvodi ni naduzorkovanje ni poduzorkovanje.

yout = upfirdn(xin,h,p) specificira cjelobrojni faktor naduzorkovanja ***p***, a podešena početna vrijednost je ***p=1***.

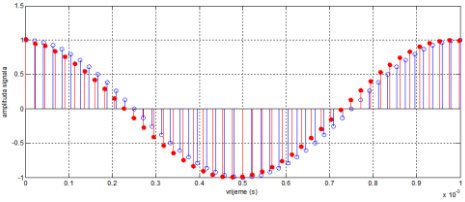
yout = upfirdn(xin,h,p,q) određuje cjelobrojnu vrijednost faktora poduzorkovanja ***q***, gdje je podešena početna vrijednost jednaka 1. Dužina, tj. broj uzoraka izlazne datoteke ***yout*** *jednaka je*

*ceil(((length(xin)-1)*p+length(h)))/q)*

Budući da *upfirdn* izvodi konvoluciju i promjenu frekvencije sempliranja izlazni signal različite je dužine od ulaznog. Približan broj redaka matrice je *p/q puta broj redaka* ulazne datoteke.

Računalo mora imati instaliran MEX compiler *(upiši: MEX –setup i prati upute)*

```
L = 147; M = 160; % faktor Interpolacije/decimacije
N = 24*L;
h = fir1(N-1,1/M,kaiser(N,7.8562)); % filtriramo Kaiser filtrom
h = L*h; %  interpolacija propusnog područja
Fs = 48e3; % Izvorna frekvencija sempliranja 48kHz DAT
n = 0:10239; % 10240 uzorka, trajanje 0.213 s
x = cos(2*pi*1e3/Fs*n); % generiramo signal DAT, sinusoida 1kHz
y = upfirdn(x,h,L,M); % 9430 uzoraka, trajanje 0.213 s
% Prekrivamo DAT signal plavo fs=48kHz s resempliranim
% CD signalom fs=44.1kHz crveno
stem(n(1:49)/Fs,x(1:49)); hold on
stem(n(1:45)/(Fs*L/M),y(13:57),'r','filled');
grid;xlabel('vrijeme (s)');ylabel('amplituda signala');
```



Pretvorba stereo *wav* datoteke s 44100 na 48000Hz :
Pohranimo li gornje wav datoteku u Matlabu, y možemo vratiti sa 44.1 na 48 kHz (u početni oblik) MATLAB-om ili npr. u **sox**-u naredbom:

```
sox input.wav input.f32
./3-options-input-fn 44100 48000 2 <input.f32 >output.f32
sox -c 2 -r 48000 output.f32 output.wav
```


3. primjer resempliranje HD datoteke 24b 96kHz i 192kHz na 48kHz

Resempliranje HD zvučne datoteke sa *96 i 192 kHz 24 bit na 48 kHz*.

FIRRateConverter naduzorkuje, filtrira i poduzorkuje ulazni signal

HFIRRC = dsp.FIRRateConverter vraća sistemski objekt, ***HFIRRC***, koji resemplira ulazni signal naduzorkovanjem cjelobrojnim faktorom L, primjenjuje FIR filter i poduzorkuje drugim cjelobrojnim faktorom M.

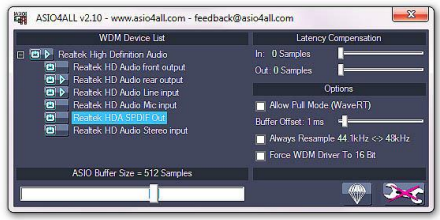
U mapi MATLAB mora biti pohranjena HD datoteka *hd_24_96.wav i hd_24_192.wav*.

Interpoliramo faktorom 1 i decimiramo faktorom 2 pa 4. Ugrađenim FIR filterom filtriramo signal.

Za reprodukciju potrebno je na računalu imati priključen sustav reprodukcije.

```
% Resempliranje HD zvučne datoteke sa 96 kHz 24 bit na 48 kHz.
hmfr = dsp.AudioFileReader('hd_24_96.wav', 'OutputDataType', 'single', 'SamplesPerFrame', 300);
   hap = dsp.AudioPlayer(48000);
% Stvori  FIRRateConverter System objekt
% int. faktor = 1, decimacijski faktor = 4. FIR filter koeficijenti određuju niskopropusni filter
   hfirrc = dsp.FIRRateConverter(1,2);
   while ~isDone(hmfr)
    audio1 = step(hmfr);
    audio2 = step(hfirrc, audio1);
    step(hap, audio2);
   end
   release(hmfr); % zatvori ulaznu datoteku
   release(hap); % zatvori izlazni uređaj za reprodukciju zvuka
```

```
% Resempliranje HD zvučne datoteke sa 192 kHz 24 bit na 48 kHz.
hmfr = dsp.AudioFileReader('hd_24_192.wav', 'OutputDataType', 'single', 'SamplesPerFrame', 300);
   hap = dsp.AudioPlayer(48000);
% Stvori  FIRRateConverter System objekt
% int. faktor = 1, decimacijski faktor = 4.FIR filter koeficijenti određuju niskopropusni filter
   hfirrc = dsp.FIRRateConverter(1,4);
    while ~isDone(hmfr)
     audio1 = step(hmfr);
     audio2 = step(hfirrc, audio1);
     step(hap, audio2);
   end
   release(hmfr); % zatvori ulaznu datoteku
   release(hap); % zatvori izlazni uređaj za reprodukciju zvuka
```



Gustoća snage u frekvencijskom području

$$P = \frac{1}{2\pi N}\int_{-\pi}^{\pi} |X_N(\Omega)|^2 \, d\Omega$$

$$d\Omega = \frac{2\pi}{f_s} df$$

gdje je

$$f = f_s/2 \quad za \quad \Omega = \pi$$

pa je

$$P = \left(\frac{1}{2\pi N}\right)\left(\frac{2\pi}{f_s}\right)\int_{-f_s/2}^{f_s/2} |X_N(f)|^2 df$$

ili

$$P = \int_{-f_s/2}^{f_s/2} \frac{|X(f)|^2}{N f_s} df$$

$$p(f) = \frac{|X(f)|^2}{N f_s}$$

Naduzorkovanje i slučajni šum

Snaga Gaussovog šuma jednoliko je distribuirana u frekvencijskom području

Posljedica:

gustoća snage šuma smanjuje se za faktor 2 ili (*3 dB*) podvostručenjem frekvencije sempliranja.

Proračun *srednje gustoće snage šuma* pri različitim frekvencijama sempliranja

```
[zero_sig,tt]=analog(1,0,1000,50000);        % konstruiramo psuedo-analogni nulti signal
analog_noise=zero_sig+randn(size(zero_sig)); % konstruiramo pseudo-analogni signal šuma
n1000=sample(tt,analog_noise,1000);          % sempliramo signal šuma frekvencijom sempliranja od 1 kHz, 2 kHz i 4 kHz
n2000=sample(tt,analog_noise,2000);
n4000=sample(tt,analog_noise,4000);
[N1,f1]=dtft_demof(n1000,0,500,512,1000);    % računamo  DTFT za svaki digitalni signal šuma
[N2,f2]=dtft_demof(n2000,0,1000,512,2000);
[N4,f4]=dtft_demof(n4000,0,2000,512,4000);
pd1=(abs(N1)).^2/(length(n1000)*1000);       % računamo gustoću spektra snage za svaki digitalni signal šuma
pd2=(abs(N2)).^2/(length(n2000)*2000);
pd4=(abs(N4)).^2/(length(n4000)*4000);
mean(pd1)                                    % računamo srednju gustoću spektra snage za svaki digitalni signal šuma
mean(pd2)
mean(pd4)
```

```
ans =
   9.9750e-004

ans =
   4.8431e-004

ans =
   2.5061e-004
```

Uočavamo:

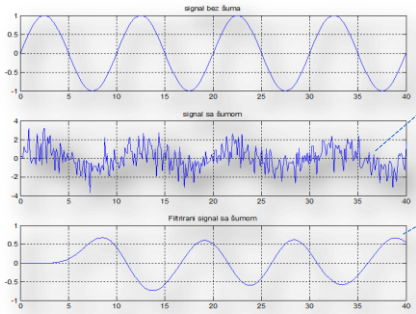
srednja gustoća snage se dvostruko smanjuje za svako podvostručenje frekvencije sempliranja

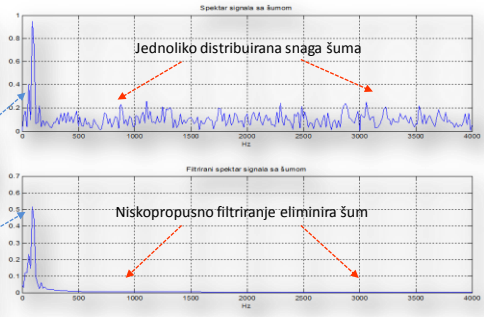
Eliminiranje šuma iz signala naduzorkovanjem i niskopropusnim filtriranjem

- Naduzorkujemo signal taktom koji je mnogostruko viši od frekvencijske širine pojasa korisnog signala *B[Hz]*
- Naduzorkovanje smanjuje snagu šuma u pojasu korisnog signala *B[Hz]* za ***3 dB pri svakom podvostručenju frekvencije sempliranja***
- Filtriramo signal niskopropusnim filterom odrezne frekvencije u blizini *B[Hz]*.
- Tako eliminiramo snagu šuma na frekvencijama iznad *B[Hz]*.

Primjer eliminiranja šuma naduzorkovanjem i niskopropusnim filtriranjem

```
[asig,tt]=analog(100,1,40,40000);
asign=asig+randn(size(asig));
[d8,t8]=sample(tt,asig,8000);
[d8n,t8]=sample(tt,asign,8000);
n=0:100;
omega=2*pi*125/8000;
hw=(omega/pi)*sinc(omega*(n-
    50)/pi).*blackman(101)';
d8nf=filter(hw,1,d8n);
subplot(3,1,1),plot(t8,d8),grid
title('signal bez šuma')
subplot(3,1,2),plot(t8,d8n), grid
title('signal sa šumom')
subplot(3,1,3),plot(t8,d8nf), grid
title('Filtrirani signal sa šumom')
pause
subplot(2,1,1),fft_plot(d8n,8000); grid
title('Spektar signala sa šumom')
subplot(2,1,2),fft_plot(d8nf,8000); grid
title('Filtrirani spektar signala sa šumom')
```





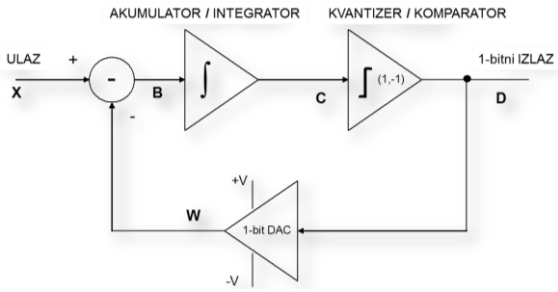
Delta-Sigma (ΔΣ) kvantizacija

Semplirani i kvantizirani signal može se digitalno prikazati na više različitih načina. Najednostavniji je LPCM. To je najuobičajeniji način prikaza digitalnog signal kao sekvence binarno kodiranih vrijednosti amplitude.

Delta-Sigma modulacija opisuje potpuno različit način pristupu kodiranja signala. Brzina obrade DSP sustava mjeri se u bit/s. Brzina je Nf_s, gdje je N broj kvantizacijski bitova AD pretvarača, a f_s je frekvencija sempliranja.

Kod Delta-Sigma sustava i zadanoj brzini sustava, maksimalna f_s zahtijeva jednobitnu kvantizaciju koja se na prvi pogled čini nemogućom jer nastaje vrlo visoki šum kvantizacije kao posljedica činjenice da svaki bit doprinosi odnosu S/Š 6dB.

Međutim, ***ΔΣ*** kvantizacija kvantizira sa samo 1 bitom, a istovremeno upravlja kvantizacijskim šumom.



DSM pretežno se koristio kao *među-kod* pri pretvorbi podataka. Međutim, delta sigma se razvila u glavni način kodiranja komercijalnih sustava poput Super Audio CD. ***SACD*** koristi oblik DS modulatora koji se naziva Direct Stream Digital (***DSD***).

Tako DSM s vrlo visokim naduzorkovanjem daje na izlazu struju bitova. Struja bitova se po potrebi sažima i neposredno pohranjuje na medij bez pretvorbe u LPCM kao kod CD. Motivacija za korištenje ovakve sheme kodiranja je povišenje kvalitete zvučnog signala bez potrebe za pretvorbom formata.

Što je *SACD*, a što je *DSD*?

SACD je **ZAŠTIĆENI FORMAT DISKA** kojeg je razvio *Sony* krajem 90-tih, a koristi DSD (Direct Stream Digital) format zapisa. SACD diskovi mogu se reproducirati **samo na reproduktorima sa SACD logom**. **Ne postoji uređaj (pržilica) za izradu SACD diskova** pa se on ne može napraviti kod kuće na računalu. Jedini način izrade SACD diskova postoji u nekoliko velikih postrojenja u svijetu.

DSD (Direct Stream Digital) je **OTVORENI SLOBODNI FORMAT KOJI NE ZAHTJEVA LICENCU PRI PRODAJI ILI REPRODUKCIJI DATOTEKA**. Dakle, DSD je poput magnetofonskih traka ili PCM digitalnog formata te je svakom dostupan. U DSD formatu može se snimati pa su dostupni čitavi sustavi za kućne studije i profesionalnu primjenu. Neki studiji za snimanje zastupaju mišljenje da je DSD format bolji od LPCM formata pa nativno snimaju i obrađuju zvučne datoteke u DSD formatu. Takve datoteke mogu će je reproducirati sa web ili kućnih muzičkih poslužitelja i DSD DAC-ova, a moguće ih je i snimiti primjenom kućnog računala na sve oblike DVD medija.

Dakle, DSD Diskovi su samo drugi način distribucije DSD zvučnih signala. Ako se kod kuće izradi DSD Disk najčešće ga nije moguće reproducirati preko SACD reproduktora, osim ako na reproduktoru **tvornički namjerno nije izostavljen sustav zaštite**. Preko računala DSD disk se može reproducirati programima kojim se isti izrađuju, a neka prijenosna računala već tvornički imaju ugrađen cijeli sustav za snimanje, reprodukciju i izradu DSD formata na DVD.

Najkraće,

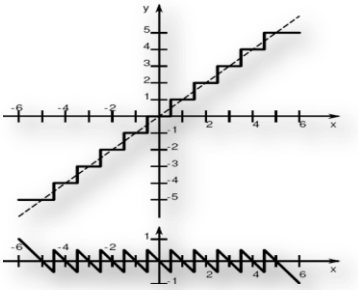
Na *SACD* mediju je upisan *DSD* format sa zaštitom protiv kopiranja.

***DSD* je otvoreni format dostupan svakom, može se pohraniti na DVD medij, nema zaštite.**

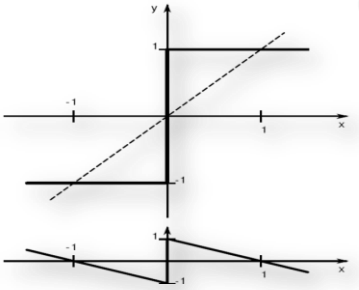
DSD* koristi *Delta-Sigma Modulator

Zbog vrlo bliskog odnosa sa A/D pretvorbom, vrlo je korisno analizirati kvantizaciju u smislu **diskretiziranja kontinuiranog ulaznog signala**. Na delta sigma modulator primjenjuju se potpuno iste zakonitosti bez obzira da li je kvantizacija signala u više ili manje koraka ili s jednoliko ili nejednoliko udaljenim razinama.

Najjednostavnijim riječima, kvantizacija je **proces aproksimiranja kontinuirane amplitude primjenom sekvence od dva ili više koraka koji su predstavljeni binarnim vrijednostima**.



11 koračni kvantizer s q=1 (Δ=1)



kvantizer s dva koraka gdje je Δ=2

Slika prikazuje dva različita načina kojim se kvantiziraju analogni signali. Kvantizer lijevo primjenjuje se kao analogni komparator ili digitalno kao ekstraktor MSB ili bita predznaka. Greška kvantizacije, bilo kojega kvantizera, uz pretpostavku **nekoreliranog modela bijelog šuma pogreške**, gdje je Δ korak kvantiziranja, iznosi:

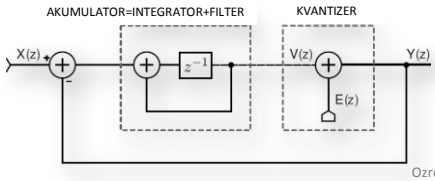
$$e_{rms}^2 = \frac{\Delta^2}{12}$$

Nyquist-Shannon teorem sempliranja tvrdi da se pojasno ograničeni signal do frekvencije f_B može točno predstaviti sekvencom jednoliko udaljenih vrijednosti ako ga sempliramo s frekvencijom $fs=2f_B$. Za dobro predstavljanje signala, uzorci podataka moraju biti vrlo točno kvantizirani po amplitudi. Zahtjev pretvarača se teško postiže ako se koriste vrlo niski naponi napajanja jer se postavljaju vrlo veliki zahtjevi za analogne komponente sklopa.

Preuzorkovanje povećava broj uzoraka mnogostruko iznad *Nyquistove* frekvencije. Faktor preuzorkovanja OSR je omjer stvarne frekvencije sempliranja i *Nyquistove* frekvencije sempliranja, što možemo sumirati na slijedeći način: $OSR = f_N/2f_B$, gdje je $f_N=Nfs$

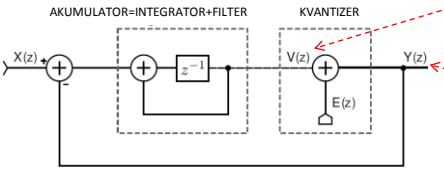
Svako podvostručenje odnosa preuzorkovanja (OSR) smanjuje snagu šuma *3dB*. Iako preuzorkovanje nije vrlo učinkovito, nužno je za koncept oblikovanja šuma. Najjednostavnijim rječnikom; **oblikovanje šuma je postupak filtriranja šuma kvantizacije u spektar u kojem ima mnogo više visokofrekvencijskih komponenti nego niskofrekvencijskih, na kojima se nalazi korisni signal. Dakle, oblikovanje šuma u konačnici, vrši pomak snage šuma greške kvantizacije iz frekvencijske širine pojasa korisnog signala na mnogo više frekvencije.**

To se postiže primjenom različitih filterskih funkcija na ulazni signal i kvantizacijom signala greške. Slika prikazuje primijenjen linearni ekvivalenti DSM 1-reda. Kvantizer je zamijenjen sumom signala greške za koju pretpostavljamo da je neovisna o ulaznom signalu. Na opisanom modelu odredit ćemo prijenosne funkcije koje opisuju signal i šum.



Bit **Delta-Sigma** AD pretvarača:
Primjenjuje negativnu povratnu vezu pri akumulaciji valnog oblika. Ako se ulazni valni oblik, akumuliran iznad jednog perioda uzimanja uzorka poveća iznad akumulirane vrijednosti, izlaz pretvarača je digitalna 1. Ako valni oblik oslabi u odnosu na akumuliranu vrijednost, izlaz pretvarača je digitalna 0.

Iz sheme slijedi:



$$V(z) = z^{-1}(V(z) + X(z) - Y(z))$$
$$= \frac{z^{-1}}{1 - z^{-1}}(X(z) - Y(z))$$

$$Y(z) = E(z) + V(z)$$
$$= E(z) + \frac{z^{-1}}{1 - z^{-1}}(X(z) - Y(z))$$
$$(1 - z^{-1})Y(z) = (1 - z^{-1})E(z) + z^{-1}X(z) - z^{-1}Y(z)$$
$$Y(z) = (1 - z^{-1})E(z) + z^{-1}X(z)$$

$$NTF(z) = 1 - z^{-1}$$
$$STF(z) = z^{-1}$$

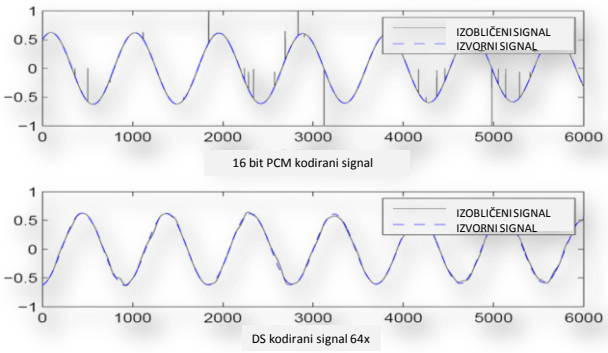
Izlaz je predstavljen prijenosnom funkcijom šuma *NTF* s vrlo jednostavnim oblikom filtera i prijenosnom funkcijom signala *STF*, predstavljenom kašnjenjem ulaznog signala. DSM se može napraviti i bez kašnjenja tako da premjestimo kašnjenje iz integratora u dodatno kašnjenje u petlju povratne veze. Prijenosna funkcija šuma može biti prvog ili višeg reda. **Primjenom vrlo visoke frekvencije sempliranja** (*preuzorkovanje analognog signala*) možemo prikazati kako srednja vrijednost 1-bitnog izlaza aproksimira ulazni signal. Međutim, kako bi dobili dobar uvid u **kvalitetu** signala potrebno je izvršiti analizu u **frekvencijskom području**. U praktičnoj primjeni oblikovanje šuma modulatorom prvog reda nije dovoljno za visoku kvalitetu signala pa se radi kaskada filtera oblikovanja šuma prvog reda. Pri tome nastaju problemi primjene DSM. Najveći problem je **kompromis između kvalitete signala i brzine takta** što dovodi do neprihvatljive potrošnje, povećanja radne temperature i ograničenja frekvencijske širine pojasa uređaja. To su razlozi zbog kojih se *delta-sigma kodiranje* najviše koristi kao *međukod* pri pretvorbi iako postoji *nativna* primjena DSM u obliku DSD. Za digitalnu obradu signala najviše se koristi LPCM.

$$Y(z) = z^{-1}X(z) + (1 - z^{-1})E(z)$$

Temelj DSM u tome da se signal koji se struji iz jednolikog 1-bitnog delta sigma analogno digitalnog pretvarača **zadržava u jednobitnom formatu, a zatim se primjenjuje DAC**. Ovaj postupak rezultira izlaznim strujanjem jednoliko semplirane podatkovne strukture u kojoj je svakom bitu pridijeljena ista težina, a vrijednost signala je predstavljena vremenskim slijedom grupe bitova. Pri tome svaki bit ima istu težinsku vrijednost u odnosu na amplitudu signala. To svojstvo čini Delta Sigma kodiranje vrlo otpornim na greške, jer pogreška nekoliko bitova ima mnogo manji utjecaj na konačnu vrijednost signala nego pri LPCM kodiranju.

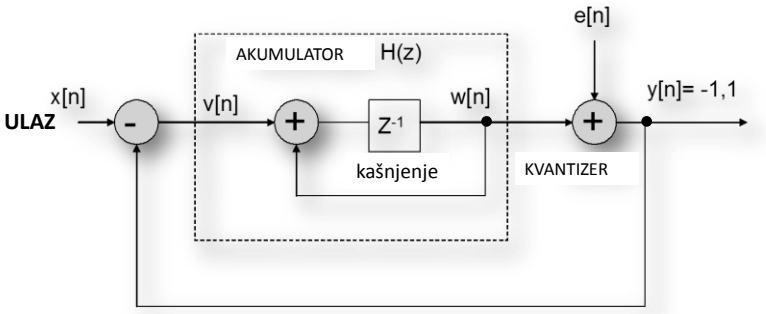
Slika prikazuje rezultat jednostavnog testa kako mali iznos pogrešnih bitova utječe na PCM i Delta Sigma kodirani signal. Delta Sigma signal je preuzorkovan *64x*, moduliran s modulatorom drugog reda, a rekonstruiran FIR-filterom višeg reda. Pogrešni bitovi su slučajno invertirani s istom vjerojatnosti 0,02% za PCM i Delta Sigma signale.

Delta Sigma kodiranje pokazuje izvrstan način primjene aritmetičkih postupaka otpornih na greške. PCM sustavi u praksi koriste složenije sustave zaštite od greške kao što su jednostavna ciklička provjera redundancije, *Reed Solomon Code* i *dvostruki Reed Solomon Code*.



PCM (gore) i DS (dolje) kodirani signal u kojem je 0,02% bita slučajno invertirano, a prikazani su signali sa i bez pogreške

Diskretni vremenski model *Delta-Sigma* kvantizera



$$Y(z) = z^{-1}X(z) + (1 - z^{-1})E(z)$$

IZLAZ = **ULAZ** x kašnjenje + šum x visokopropusni filter

1-bitna kvantizacija s oblikovanjem šuma

Ako je kvantizer rezolucije *B*-bita njegovo ukupno radno područje bez preopterećenja bit će $2^B \cdot \Delta$. **Najviša razina sinusoidalnog signala koji neće izazvati preopterećenje** bit će onda $x[n]=2^{B-1} \cdot \Delta \cdot \sin(\omega n)$, a snaga (varijanca) će biti $\sigma_x^2 = 2^{2B} \cdot \Delta^2/8$. Vršni odnos *SQNR* (signal/šum kvantizacije) za sinusoidalni signal na ulazu je:

$$SQNR_{\max} = 10 \cdot \log_{10}\left(\frac{\frac{2^{2B} \cdot \Delta^2}{8}}{\frac{\Delta^2}{12}}\right) = 6.02B + 1.76 \ \text{[dB]}$$

Izrez predstavlja pravilo **6dB po bitu** koje se koristi i za proračun **ekvivalentnog broja bita** *ENOB* **DSM**. Nužna širina bita za ulaz DSM povezan je sa odnosom SNR koji se može postići. Očekuje se vrlo niski odnos SNR modulatora. Kako bi u praksi točno izmjerili DSM ulazni signal treba imati nešto višu rezoluciju od izlaznog signala. Tako se jamči da rezolucija neće biti ograničavajući faktor. Standardni način izražavanja rezolucije je proračun **efektivnog broja bitova** *ENOB* koji odgovara razini SNR. ***ENOB*** se definira kao nužan broj bita za rekonstrukciju PCM kodirane sinusoide za zadanim odnosom SNR.

$$ENOB = \frac{SNR - 1.76}{6.02}$$

Za očekivani SNR od 40dB gornja granica *ENOB = 6,4* tj. mora se koristit najmanje *7 bita*, kako bi osigurali da modulator ne bude ograničen s niskom ulaznom rezolucijom. Za *SNR* od oko *120dB*, kojeg deklarira *SACD DSD*, granica je *ENOB* od oko *19-20 bita*.

Ako je ulazni signal s glatkim spektrom snage, uzorci pogrešaka bit će približno *ortogonalni*. Tada je **po definiciji** *autokorelacijska funkcija pogreške*:

$$r_{ee}(k) \overset{def}{=} E\left(e[n]\cdot e[n-k]\right) \approx \begin{cases} \sigma_e^2 & , k=0 \\ 0 & , \text{inače} \end{cases}$$

Wiener-Khinchin *teorem* pokazuje da će gustoća spektra snage pogreške biti:

$$S_e(\omega) = \frac{1}{2\pi}\cdot \Im\{r_{ee}(k)\} = \frac{\sigma_e^2}{2\pi}$$

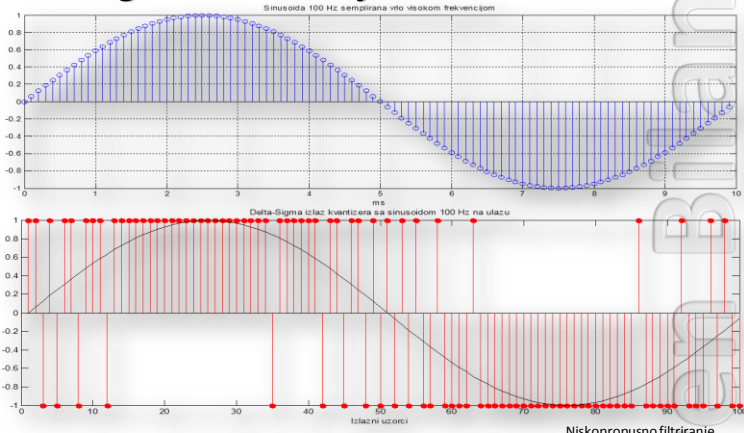
Prvi su nastali LPCM pretvarači koje su naslijedili (*1-bit*) DSM, a naknadno su razvijeni s filterima s preklapanjem kapaciteta. Kasnih 90-tih, višebitni DSM preuzeo je primat nad 1-bitnim. DAC u strujnom modu je ranih 2000-tih zamijenjen sa filterima s preklapanjem kapaciteta. Karakteristike sustava se stalno poboljšavaju kao i učinkovitost kvantificirana faktorom *FOM*

$$FOM = \frac{2^{ENOB}\cdot 2f_b}{P}$$

Godina	Topologija	SNDR fs	Snaga	fB	ENOB	FOM ($\times 10^9$)
(1986)	16-bit LPCM current-divider	95dB	400mW	20kHz (A)	15.5	4.63
(1987)	1-bit DSM, CT CMOS buffer	90dB	150mW	20kHz	14.7	7.10
(1991)	1-bit DSM, SC DAC	102dB	375mW	20kHz (A)	16.7	11.4
(2000)	Multi-bit DSM, SC DAC	102dB	155mW	20kHz (A)	16.7	27.5
(2000)	Multi-bit DSM, SC DAC	100dB	100mW	20kHz (A)	16.3	32.3
(2000)	Multi-bit DSM, I-DAC	108dB	111mW[9]	20kHz (A)	17.6	71.6
(2001)	Multi-bit DSM, I-DAC	112dB	125mW[10]	20kHz (A)	18.3	103
(2003)	Multi-bit DSM, PWM hyb.I-DAC	>110dB[11]	75mW	20kHz (A)	>18.0	>140

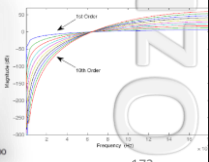
Delta-Sigma kvantizacija sinusoide

```
[x,ts]=analog(100,1,500,10000);
stem(ts(1:100),x(1:100))
grid
title('Sinusoida 100 Hz semplirana vrlo visokom frekvencijom')
xlabel('ms')
y=deltasigma2(x);
% funkcija algoritma delta-sigma kvantizera
figure,stem(y(1:100),'r','filled');grid
hold
plot(x(1:100),'k')
grid
title('Delta-Sigma izlaz kvantizera sa sinusoidom 100 Hz na ulazu')
xlabel('Izlazni uzorci')
hold off
```

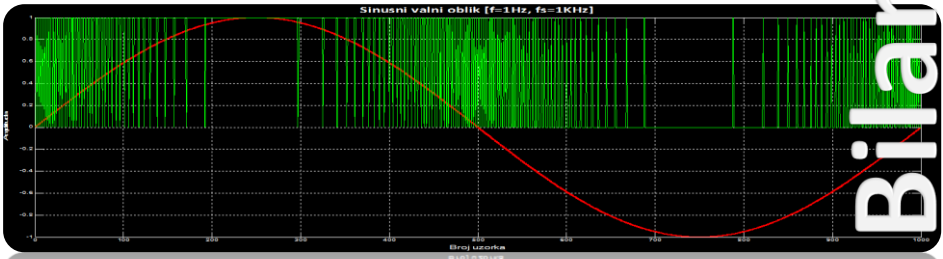


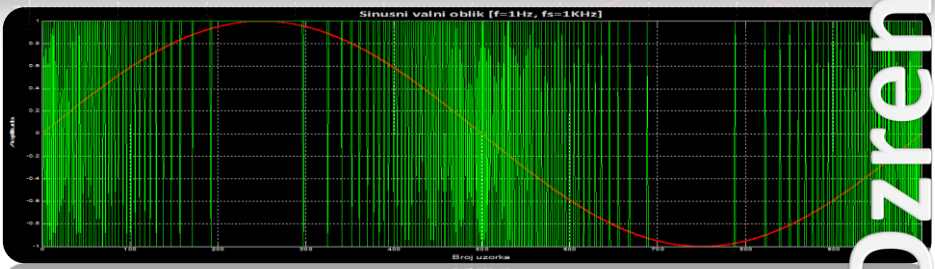

```
>> n=0:500;
omega=2*pi*200/10000;
[x,ts]=analog(100,1,500,10000);
y=deltasigma2(x);
fft_plot(y,10000);
title('Izlazni spektar nefiltriranog Delta-Sigma kvantizera')
```

Niskopropusno filtriranje eliminirat će šum kvantizacije



Unipolarni i bipolarni prikaz signala

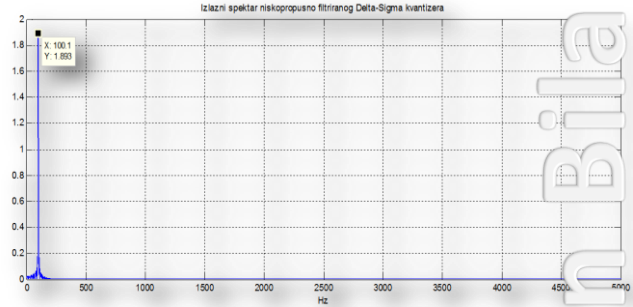


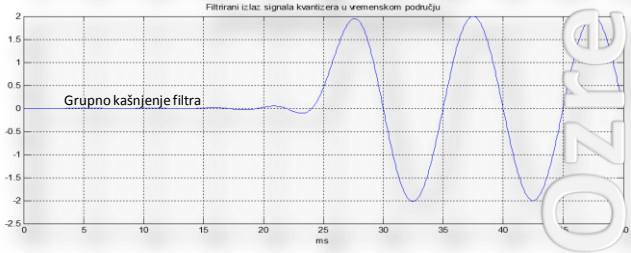


Ozren Bilan

Niskopropusno filtriranje *ΔΣ* sinusoide

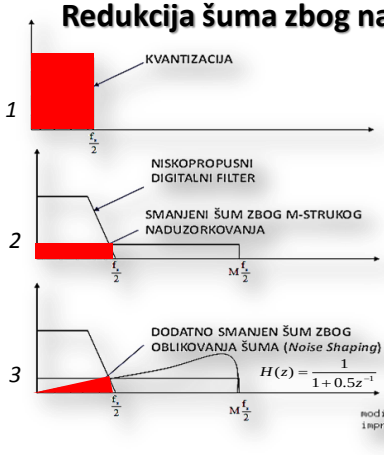
```
n=0:500;
omega=2*pi*200/10000;
h=(omega/pi)*sinc(omega*(n-250)/pi);
hb200=h.*blackman(501)';
[x,ts]=analog(100,2,500,10000);
y=deltasigma2(x);
% algoritam D-S kvantizera
y_lpf=filter(hb200,1,y);
fft_plot(y_lpf,10000);
title('Izlazni spektar niskopropusno filtriranog Delta-Sigma kvantizera')
figure,plot(ts(1:500),y_lpf(1:500))
title('filtrirani izlaz signala kvantizera u vremenskom području')
xlabel('ms')
```





Ozren Bilan

Redukcija šuma zbog naduzorkovanja i oblikovanja šuma

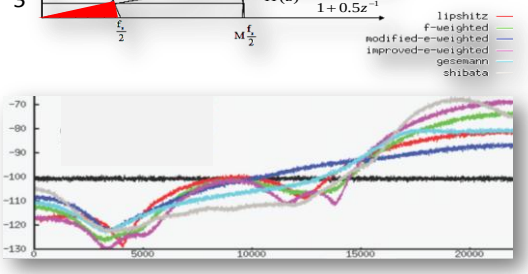


Usporedba funkcija **gustoće spektra snage kvantizacijskog šuma** uz isti broj bitova:

obični DAC;
efektivna varijanca šuma = površina 1

naduzorkujući DAC;
efektivna varijanca šuma = površina 2

naduzorkujući DAC s oblikovanjem šuma;
efektivna varijanca šuma = površina 3.



Postoje različiti algoritmi oblikovanja šuma koji se primjenjuju programski ovisno o tipu programa

Frekvencijski odzivi različitih algoritama oblikovanja šuma u *Sox resampleru*

format: [-a][-S|-s|-f filter] (none) TPDF
-a Automatski uključuje i isključuje dither po potrebi
-S sloped TPDF (bez oblikovanja šuma)
-s oblikovanje šuma (shibata filter)
–f ime filter se postavlja na jedan od slijedećih: lipshitz, f -weighted, modified-e weighted, improved-e-weighted, gesemann, shibata, low-shibata, high-shibata.

Zaključak

- Spektralni sadržaj digitalnog signala se replicira na cjelobrojnim višekratnicima frekvencije sempliranja.
- **Preuzorkujemo** li datoteku frekvencijom sempliranja mnogo višom od *Nyquistove*, frekvencijski sadržaj replika spektra proširit će se na mnogo šire područje.
- Ako preuzorkujemo pa digitalno niskopropusno filtriramo signal znatno se olakšavaju zahtjevi analognog anti-alias filtera.
- Prekomjerni uzorci preuzorkovanog pojasno ograničenog signala eliminiraju se poduzorkovanjem ili decimacijom što snižava efektivnu frekvenciju sempliranja.
- Signal se rekonstruira na višu frekvenciju sempliranja postupkom **naduzorkovanja** i **interpolacije**.
- Budući da je *Gaussov šum* u frekvencijskom području jednoliko raspodijeljen kombinacija postupaka naduzorkovanja i filtriranja učinkovito eliminira šum iz signala u području korisne frekvencijske širine pojasa.
- Preuzorkovanje maksimizira ***delta-sigma kvantizacija,*** što je 1-bitna kvantizacija.
- Delta-sigma kvantizer ima svojstvo **oblikovanja šuma** koje eliminira šum kvantizacije niskopropusnim filtriranjem.

PITANJA

1. Zašto je DSP multidisciplinarna?
2. Što su signali i sustavi; kakvi mogu biti?
3. Objasni svojstva linearnih vremenski nepromjenjivih sustava
4. Što određuje statička linearnost sustava?
5. Objasni princip superpozicije
6. Što je homomorfna obrada?
7. Definiraj osnovne vremenski kontinuirane signale?
8. Što je šum i kako ga opisujemo?
9. Što je srednja vrijednost i varijanca?
10. Kakva je Gaussova normalna distribucija?
11. Objasni postupak AD pretvorbe i alias
12. Objasni pojam šuma kvantizacije i dither, tipovi dithera?
13. Što je diskretna sekvenca?
14. Definiraj osnovne diskretne signale
15. Zašto pomak i refleksija nisu komutativni?
16. Objasni digitalnu frekvenciju i periodičnost diskretne sinusoide
17. Objasni jednadžbu diferencija na primjeru vremenski diskretnog sustava
18. Nacrtaj i objasni djelovanje temeljnih gradbenih blokova diskretnog sustava
19. Objasni općenito tipove diskretnih sustava
20. Objasni princip kauzalnosti
21. Objasni vremenske promjenjive i nepromjenjive sustave
22. Objasni razliku linearnih i nelinearnih sustava
23. Nyquistov A/D pretvarač radi sa 8 bitnom kvantizacijom, a dinamičko područje kvantizatora je od -2 do 2V.
    a) odrediti ukupan broj razina kvantizatora i interval kvantizacije.
    b) Kolika je maksimalna pogreška kvantizacije?
    c) Kolika je snaga šuma kvantizacije (mjerena na R=1 ohm)
    d) Kolika je snaga signala (sinusni) koji pobuđuju cijelo područje kvantizatora?
    e) Odrediti odnos signal šum uz sinusni signal koji pobuđuje cijelo i pola područja kvantizatora.
    f) Koliki je tok podataka ako se uzorkovanje izvodi sa 44,1 kHz?
24. Višetaktna obrada:
    preuzorkovanje,
    naduzorkovanje,
    decimacija,
    interpolacija.
25. Resempliranje
25. Delta sigma pretvarač
26. Naduzorkovanje i oblikovanje šuma

Ozren Bilan