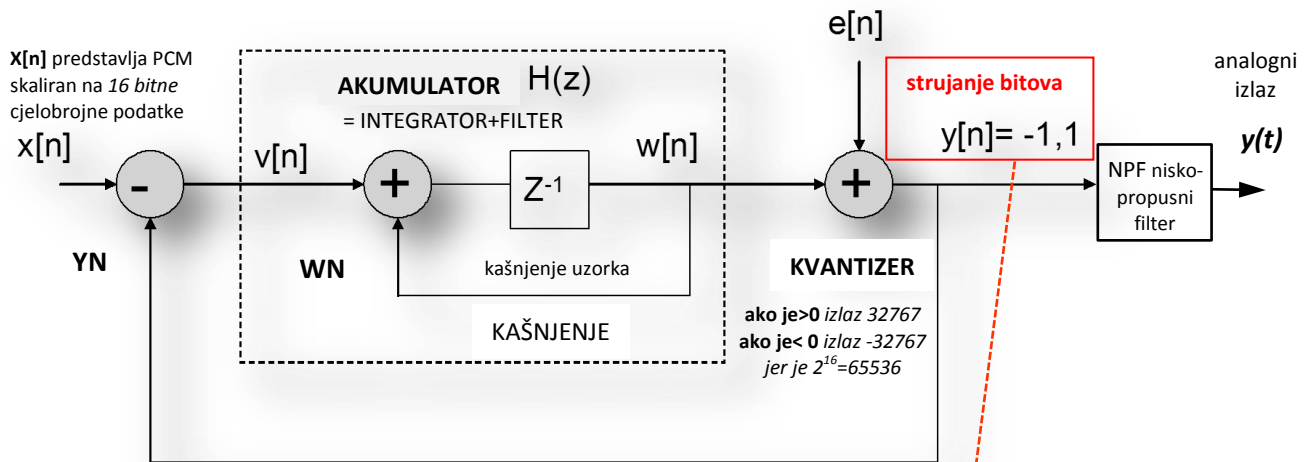


### Jednostavno objašnjenje Delta-Sigma Modulacije s MATLAB primjerom

Algoritam *Delta-Sigma Modulatora* temeljen je na postavci potpune aproksimacije analognog valnog oblika korištenjem samo jednog bita. Postavka je vrlo slična načinu rada impulsno širinske modulacije PWM koju je koristio *Laserdisc video*. Međutim, DSM postiže mnogo bolju rezoluciju. DSM višeg reda zahtijevaju sklopove za množenje i zbrajanje pa su im sklopovi skuplji od jednostavnih PWM.

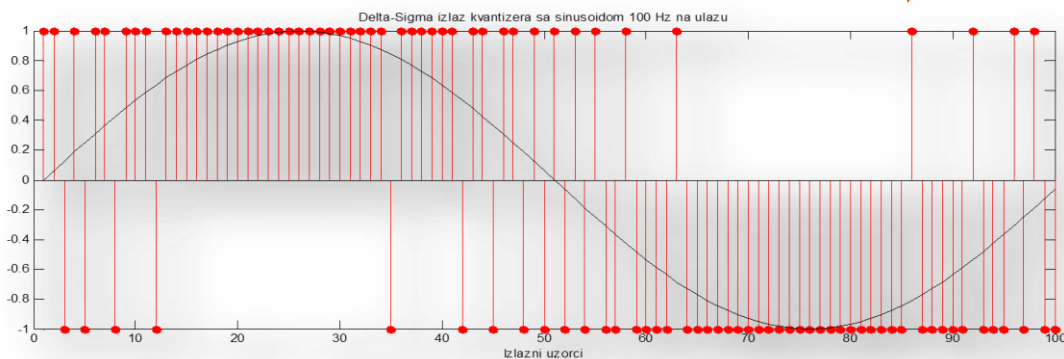
Delta-sigma signal izmjenjuje se između razina koja predstavljaju isključeno i uključeno stanje, a predstavljen je binarnim signalom koji se naziva *strujanje bitova (bitstream)*. Pri tome je usrednjena vrijednost strujanja bitova, unutar nekog intervala  $t$ , proporcionalna izvornom signalu. Zbog toga, analogni signal možemo dobiti jednostavnim niskopropusnim filtriranjem. DSM prvog reda možemo prikazati slijedećom shemom.



Iz sheme 1-bitnu kvantizaciju s oblikovanjem šuma možemo opisati izrazom:

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

IZLAZ = ULAZ x kašnjenje + šum x visokopropusni filter



Na delta sigma modulator, u smislu **diskretiziranja kontinuiranog ulaznog signala**, primjenjuju se potpuno iste zakonitosti bez obzira je li 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**. Integratorom se izvodi oblikovanje šuma, a povratnom vezom se aproksimira ulazni signal. Pri tome povećanje frekvencije uzorkovanja predstavlja temeljni postupak kojim izlazni signal  $y[n]$  aproksimira ulazni signal  $x[n]$ .

Općenito, **CD kvalitetu** predstavlja kvantiziranje na **16 bita** uz frekvenciju uzorkovanja od **44100 Hz**. Svaki bit pridonosi omjeru signal šum sa 6 dB, pa teoretski CD kvaliteta postiže odnos signal šum od oko 96 dB.

Tipični SDM 4. reda postiže isti omjer signal šum korištenjem samo 1 bita ako se povisi frekvencija uzorkovanja 45 puta. U praksi DSD signal koristi 64 puta višu frekvenciju uzorkovanja od 44100 Hz, tj. 2822400 Hz. Stvarno postignuti odnos signal šum je oko 120 dB. Pored toga izlazni signal  $y[n]$  mora se propustiti kroz niskopropusni filter. Svako podvostručenje frekvencije uzorkovanja smanjuje snagu šuma  $3dB$ .

Iako povećanje frekvencije uzorkovanja nije vrlo učinkovito, nužno je za koncept oblikovanja šuma. Najjednostavnijim rječnikom; **oblikovanje šuma je postupak filtriranja šuma kvantizacije u spektru u kojem ima mnogo više visokofrekvencijskih komponenti nego niskofrekvencijskih, na kojima se nalazi koristan signal. Dakle, oblikovanje šuma u konačnici, vrši pomak snage šuma greške kvantizacije iz frekvencijske širine pojasa korisnog signala na mnogo više frekvencije.**

DSM vrlo je koristan za digitalnu obradu **zvučnih signala** jer ima dvije vrlo povoljne karakteristike:

- dinamičko područje SDM signala ograničeno je **samo** faktorom naduzorkovanja.
- SDM funkcionira **i sa digitalnim (sempliranim) PCM** strujanjem **i sa analognim** podacima.

Dakle, postojeće analogne i digitalne datoteke mogu se bez problema konvertirati u DSD oblik.

U najjednostavnijem obliku možemo ga shvatiti kao sklop koji bez greške određuje hoće li izlazni signal  $y[n]$  biti +1 ili -1. Pozitivni ili negativni jedinični signali se dovode u akumulator mnogostruko brže nego što dolaze ulazni uzorci. Tu brzinu određuje faktor naduzorkovanja. *Delta-Sigma Modulatori* višeg reda mogu imati probleme stabilnosti.

Najkraće bit je u slijedećem - *Delta-Sigma pretvarač* 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 slabi u odnosu na akumuliranu vrijednost, izlaz pretvarača je digitalna 0.

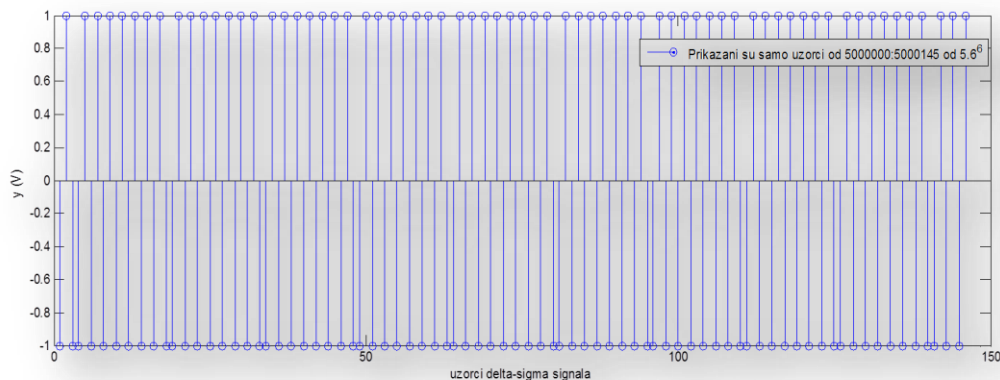
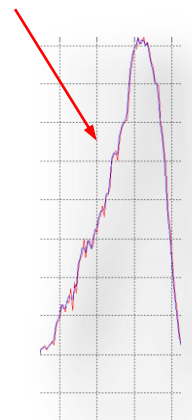
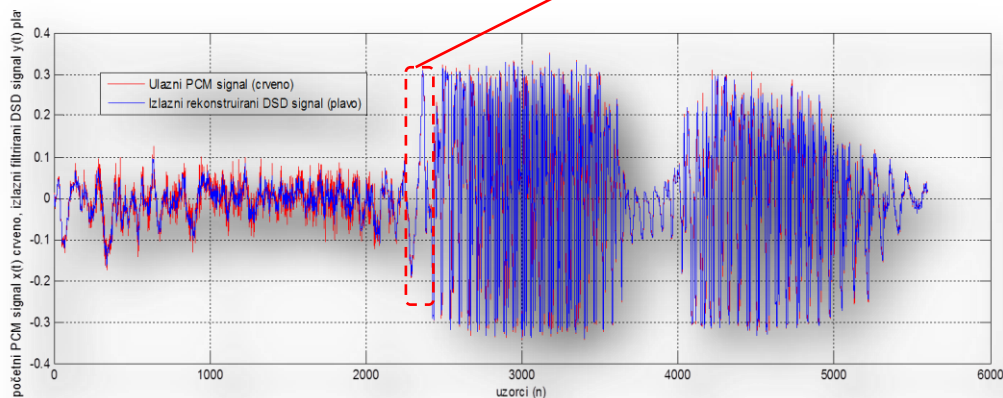
U Matlabu, prema gornjoj shemi, možemo napisati jednostavni program kojim ulaznu PCM datoteku, koju smo sami snimili, možemo pretvoriti u **jednobitno strujanje**, a zatim je vratiti u analogni oblik. Na izlazu računala *bit perfect* strujanjem s ASIO pogonskim programom, preko vanjskog asinkronog DAC-a možemo čuti ulaznu i izlaznu datoteku u rezoluciji po volji. Program prikazuje valne oblike ulazne i izlazne datoteke, te delta-sigma jednobitno strujanje. Budući da ćemo koristiti vrlo veliko povećanje frekvencije sempliranja, strujanje nećemo moći prikazati u cjelini odjednom, pa ćemo prikazati samo mali dio strujanja.

MATLAB:

```
%učitavamo wav datoteku po volji koju smo snimili u Matlabu ili neku drugu
[x, fs] = wavread('sad1.wav'); % datoteka predstavlja izgovor riječi sada
% grafički prikazujemo valni oblik crvenom bojom i zadržavamo dijagram na slici
figure
subplot(211) % na slici će biti dva dijagrama jedan iznad drugog, a na prvom će biti dva valna oblika
plot(1:length(x),x,'r');
grid;
hold on
%naduzorkujemo ulazni signal 1000 puta
Naduzorkovanje=1000
% broj uzoraka neka je jednak je dužini datoteke, a to su svi uzorci
% naredba length(X) vraća dužinu vektora x
BrojUzoraka=length(x) % zadajemo početno stanje varijabli YN i WN
YN = 0;
WN = 0;
% skaliramo podatke na 16 bitne cjelobrojne vrijednosti
% naredba round(x) zaokružava elemente vektora x na najbližu cjelobrojnu vrijednost
x=round(x*32766);
% izvodimo petlju for, n puta, od prvog do posljednjeg uzorka
for n=1:BrojUzoraka
z1(1)= WN;
xn=x(n);
for k=1:Naduzorkovanje % svaki uzorak se naduzorkuje
% na slici u tekstu vidimo definicije signala WN i YN
z1(k) = WN;
WN = z1(k) + xn - YN;
% zato jer je jer je 2^16 = 65536 uzoraka, koje od 0 dijelimo na pozitivne i negativne
YN = (WN > 0) * 32766 - (WN <= 0) * 32766;
y(k+(n-1)*Naduzorkovanje) = (WN > 0) - (WN <= 0);
end
end
```

```
subplot(212) % crtamo strujanje
% prikazujemo samo 145 uzoraka delta sigma strujanja od ukupno 5600000
stem(y(5000000:5000145))
xlabel('uzorci delta-sigma signala');
ylabel('strujanje bitova y(V) ');
legend('Prikazani su samo uzorci od 5000000:5000145 od 5.6^6')
% niskopropusno filtriramo kako bi rekonstruirali izvorni signal
% možemo mijenjati primijenjene filtre za idealnu rekonstrukciju
b=fir1(121,1/(Naduzorkovanje*2));
y=filter(b,1,y); % šum prebacujemo izvan propusnog pojasa
y=decimate(y, Naduzorkovanje); % zadržavamo samo decimirane uzorke kako bi se vratili na početnu frekvenciju sempliranja
% na istom dijagramu prikazujemo rekonstruirani signal plavom bojom
subplot(211) % vraćamo se na prvi dijagram kako bi prikazali rekonstruirano DS strujanje
plot(1:length(y),y );
xlabel('uzorci (n)');
ylabel('početni PCM signal x(t) crveno, izlazni filtrirani DSD signal y(t) plavo');
legend('Ulazni PCM signal (crveno)', 'Izlazni rekonstruirani DSD signal (plavo)')
% slušamo početnu PCM datoteku x tako da iz Matlaba ASIO driverom prosljeđujemo na asinkroni DAC
[x, fs] = wavread('sad1.wav'); % naredbom wavwrite datoteke x i y možemo pohraniti na disk
sound(x,fs);
pause
% nakon pauze reproduciramo datoteku y, pritiskom na Enter
sound(y,fs);
% obrada traje oko 4 s na dualnom procesoru 3,2GHz
```

Iz *povećanog* detalja **vidimo**: rekonstruirani DS signal idealno prati ulazni signal. To možemo **čuti** pri MATLAB reprodukciji bit perfect strujanja ASIO driverom preko asinkronog DAC i kvalitetnog sustava



**Napomena:**

Prikazani Matlab kod nije optimalan i može se poboljšati, a namjena mu je upoznavanje s programiranjem u Matlabu i obradom signala. Detaljniju analizu složenijim programima, detaljnijim prikazom oblika DSD signala i proračunom FFT spektra vidi u pdf dokumentima:

- *Kontinuirani i diskretni signali i DSP - Analogno digitalna i digitalno analogna pretvorba i višetaktni i naduzorkujući sustavi*
- *Zvučni signali visoke HD i ultra visoke UHRA razlučivosti*
- *Laboratorijske vježbe MATLAB: SPIK03 i SPIE09*