IIR-suodinten suunnittelu

- Perinteinen menetelmä IIR-suodinten suunnittelussa on suunnitella ensin vastaava analoginen suodin ja muuntaa se sitten digitaaliseen muotoon.
- Muunnoksessa käytetään ns. bilineaarimuunnosta tai impulssi-invarianttitekniikkaa.
- Tällä kurssilla ei perehdytä suunnittelumenetelmiin, vaan ne jätetään myöhemmille lineaarisen suodatuksen kursseille.
- Seuraavassa esitellään IIR-suodinten neljä eri tyyppiä ja tutkitaan kuinka suunnittelu tehdään Matlabin valmiilla rutiineilla.

IIR-suodinten suunnittelu

- Matlab-ohjelmiston ja sen signaalinkäsittelyn toolboxin avulla on mahdollista suunnitella IIR-suotimia samaan tyyliin kuin FIR-suotimiakin.
- Matlabille annetaan siis suotimelle asetettavat vaatimukset vektorimuodossa, ja Matlab palauttaa sitä vastaavan IIR-suotimen kertoimet kahtena vektorina.
- Nämä vektorit sisältävät kertoimet herätteen ja vasteen viivästetyille termeille.
- IIR-suotimet jaetaan neljään luokkaan niiden päästö- ja estokaistan värähtelyominaisuuksien mukaan:
  - Butterworth–suotimet,
  - Tyypin I Chebyshev–suotimet,
  - Tyypin II Chebyshev–suotimet,
  - Elliptiset suotimet.

päästökaista estokaista

alipäästö-
suodin

Butterworth–suotimet

- Butterworth-tyyppisille IIR-suotimille on tyypillistä, että päästökaistan alku ja estokaistan loppu (alipäästösuotimen tapauksessa) ovat molemmat mahdollisimman tasaisia (maximally flat stopband; maximally flat passband).
- Butterworth-tyyppisen IIR-suotimen suunnittelussa tarvitaan kahta komentoa, joista ensimmäinen laskee tarvittavien kertoimien määrän ja skaalaa argumenttina saamansa päästö– ja estokaistat vastaavaksi analogisen prototyypin argumentiksi.
- Käytännössä suunnittelussa ei siis tarvitse muistaa lainkaan analogisen prototyypin käyttöä; Matlab hoitaa sen käytön.

Butterworth–suotimet

- Alipäästösuodinta suunniteltaessa kertoimien määrää voidaan arvioida komennolla

```
[N, Wn] = buttord(2*Wp, 2*Ws, Rp, Rs);
```

  missä `N` on ulostulona saatava aste ja `Wn` on analogisen prototyypin rajataajuus.
- Kahden ensimmäisen syöteparametrin kerroin tarvitaan, koska Matlab skaalaa taajuudet välille $[0, 1]$ ja tällä kurssilla ne skaalataan välille $[0, 0.5]$.
- Tarvittavat argumentit ovat
  - `Wp` on päästökaistan rajataajuus
  - `Ws` on estokaistan rajataajuus
  - `Rp` on suurin sallittu värähtely päästökaistalla desibeleinä
  - `Rs` on estokaistan minimivaimennus

Butterworth–suotimet

- Suunnittelu tapahtuu tämän jälkeen komennolla

```
[b,a] = butter (N, Wn);
```

- Samaa komentoa käytetään myös ylipäästösuodinten suunnittelussa.
- Tällöin lisätään komennon viimeiseksi argumentiksi merkkijono ’high’. Esimerkiksi:

```
[b,a] = butter (N, Wn, 'high');
```

- Jos halutaan kaistanpäästö- tai kaistanestosuotimia, pitää argumenttien Ws ja Wp sijasta antaa vektorit [W1, W2] ja [W3, W4], jotka ilmoittavat amplitudivasteiden reunoja vastaavat siirtymäkaistat.
- Lisäämällä viimeiseksi argumentiksi merkkijono ’stop’, saadaan kaistanestosuodin; muutoin tuloksena on kaistanpäästösuodin.

Butterworth–suotimet

- Kaikissa tapauksissa tuloksena saadaan kaksi vektoria, jotka sisältävät siirtofunktion kertoimet: $b = (b_0, b_1, \ldots, b_N)$ ja $a = (1, a_1, \ldots, a_N)$.
- Huomaa, että tässä luvussa käytämme poikkeuksellisesti Matlabin merkintätapaa siirtofunktion kertoimille.
- Näistä ensimmäisessä on osoittajan ja jälkimmäisessä nimittäjän kertoimet:

$$\frac{Y(z)}{X(z)} = H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}}.$$

Butterworth–suotimet

- Kuten aiemmin on opittu, tästä saadaan varsinaisen suotimen kertoimet kertomalla yhtälön molemmat puolet termillä $X(z) \cdot \left(1 + \sum_{k=1}^{N} a_k z^{-k}\right)$

$$Y(z) \cdot \left(1 + \sum_{k=1}^{N} a_k z^{-k}\right) = X(z) \sum_{k=0}^{N} b_k z^{-k}$$

ja edelleen käänteisen z-muunnoksen avulla

$$y(n) + \sum_{k=1}^{N} a_k y(n-k) = \sum_{k=0}^{N} b_k x(n-k).$$

Butterworth–suotimet

- Tämä voidaan ilmaista muodossa

$$y(n) = \sum_{k=0}^{N} b_k x(n-k) - \sum_{k=1}^{N} a_k y(n-k).$$

- Suunnittelun jälkeen tämä kaava voidaankin jo sellaisenaan toteuttaa.
- Myös muut kolme IIR suunnittelukomentoa palauttavat kaksi vektoria, joilla on sama tulkinta.

Butterworth–suotimet

Esimerkki: Suunnitellaan alipäästösuodin, jonka päästökaista on normalisoiduissa taajuuksissa välillä $[0, 0.11]$ ja estokaista välillä $[0.145, 0.5]$; päästökaistan maksimivärähtely on 0.01 dB ja estokaistan minimivaimennus on 47 dB.

```
[N, Wn] = buttord (0.22, 0.29, 0.01, 47)
```

- Komento palauttaa arvot:

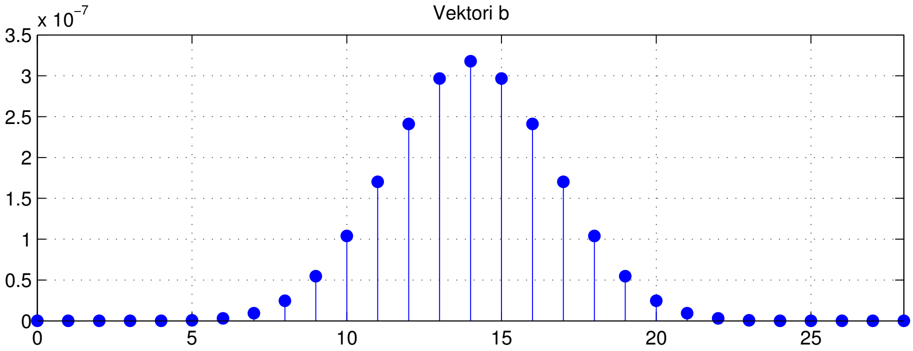
```
N = 28
Wn = 0.2443
```

- Varsinainen suunnittelu tapahtuu komennolla

```
[b,a] = butter(N,Wn);
```

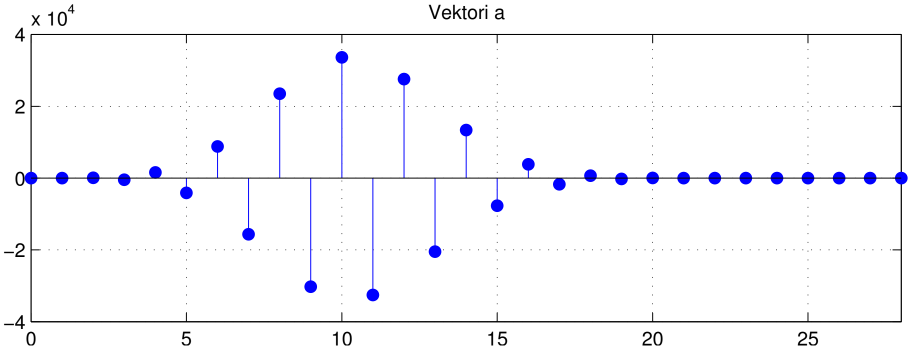
Butterworth–suotimet

- Näin saadaan kaksi vektoria; b jonka kuvaaja on seuraavassa



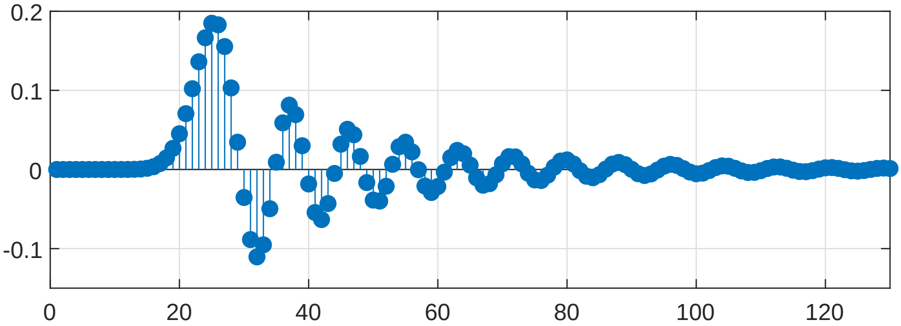
Butterworth–suotimet

ja vektori a:



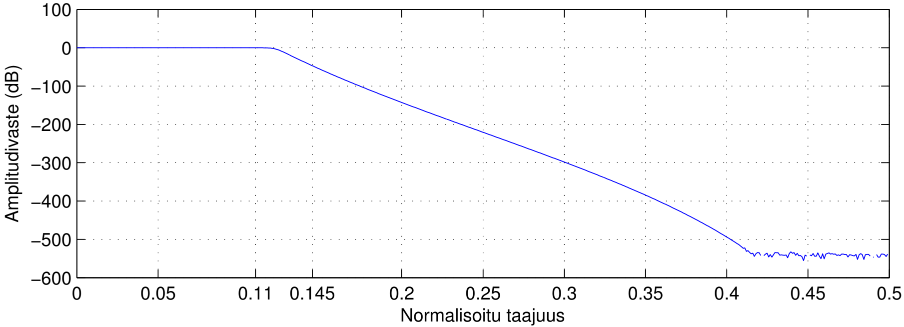
Butterworth–suotimet

- Suunnitellun suotimen impulssivasteen alku saadaan komennolla `impz(b,a)`.



Butterworth–suotimet

- Amplitudivaste (`freqz(b,a)`) on seuraavassa kuvassa.

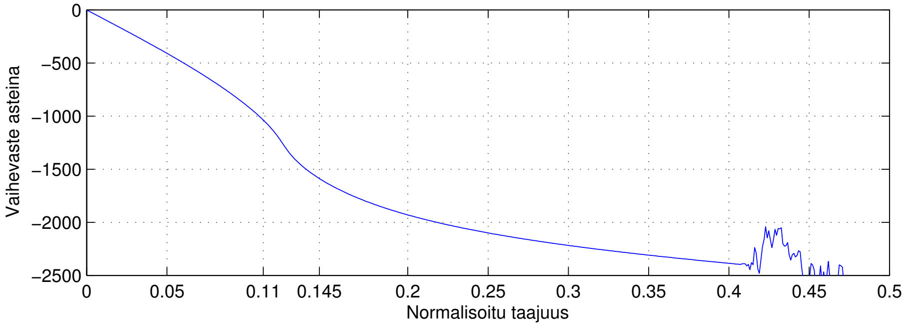


Butterworth–suotimet

- Huomaa Butterworth-suotimelle tyypillinen huikea vaimennus estokaistan loppupäässä.
- Vaimennus lähestyy arvoa $-\infty$ dB lähestyttäessä normalisoitua taajuutta 0.5.
- Selvästikin Matlabin laskentatarkkuus loppuu, kun $|\mathrm{H}(e^{\mathrm{i}\omega})|$ on alle -500 dB (lineaarisella asteikolla luvut ovat tällöin luokkaa 10^{-25}).

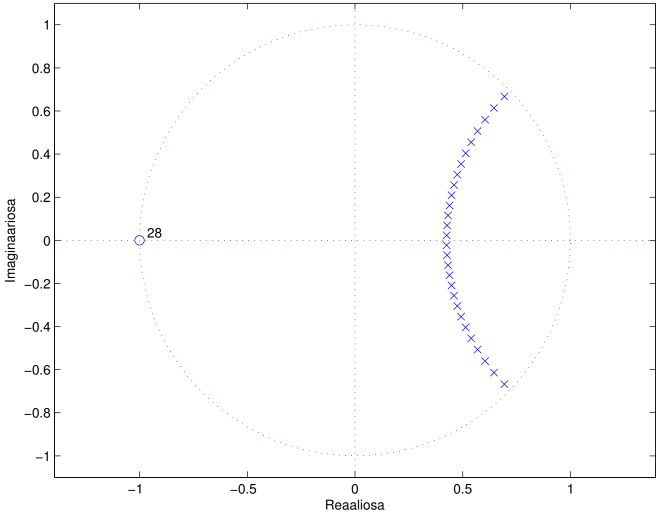
Butterworth–suotimet

- Vaihevaste on melko lineaarinen päästökaistalla eikä näin ollen haitanne useimmissa sovelluksissa. Sen kuvaaja on alla.



Butterworth–suotimet

- Lopuksi navat ja nollat:



Butterworth–suotimet

- Kuvasta nähdään kaikkien suotimen nollien sijaitsevan pisteessä $z = -1$, joka vastaa Nyquistin taajuutta.
- Näin ollen suotimen amplitudivaste tällä taajuudella on lineaarisella asteikolla nolla, eli desibeleinä amplitudivaste lähestyy arvoa $-\infty$.
- Amplitudi- ja vaihevasteen kuvaajista nähdäänkin, että Matlabin laskentatarkkuus loppuu lähellä Nyquistin taajuutta.
- Samasta syystä `zplane(b,a)`-komento ei osaa laskea tämän suotimen napa-nollakuviota oikein.

Butterworth–suotimet

- Yllä oleva kuvio saadaan käyttämällä komentoa `butter` kolmella ulostuloarvolla:

  ```
  [z,p,K] = butter(N,Wn);
  ```

  ja `zplane(z,p)`.
- Tällöin nollat ovat pystyvektorissa `z`, navat vektorissa `p` ja vahvistus nollataajuudella (ns. **gain**) muuttujassa `K`.
- Näin saatavat navat ja nollat ovat oikeat, koska Butterworth-suotimen suunnittelualgoritmi laskee ne ensin ja muuntaa tuloksen suotimen kertoimiksi.

Tyypin I Chebyshev-suotimet

- Ensimmäisen tyypin Chebyshev-suotimille on tyypillistä, että niiden estokaista on maksimaalisen tasainen (maximally flat), mutta päästökaista on tasavärähtelevä (equiripple).
- Niiden suunnittelussa on käytettävissä vastaavat komennot kuin Butterworth-suodinten tapauksessa.
- Suotimen aste voidaan arvioida komennolla

```
[N, Wn] = cheb1ord(2*Wp, 2*Ws, Rp, Rs);
```

  Suunnittelukomento on muotoa

```
[b,a] = cheby1(N,Rp,Wn);
```

Tyypin I Chebyshev-suotimet

- Muut kuin alipäästösuotimet suunnitellaan samalla tavalla kuin Butterworth-suotimen tapauksessa.
- Tarkastellaan edellisen esimerkin suodinta vastaavaa tyypin I Chebyshev-suodinta:

```
[N, Wn] = cheb1ord(0.22, 0.29, 0.01, 47)
```

- Komento palauttaa arvot:

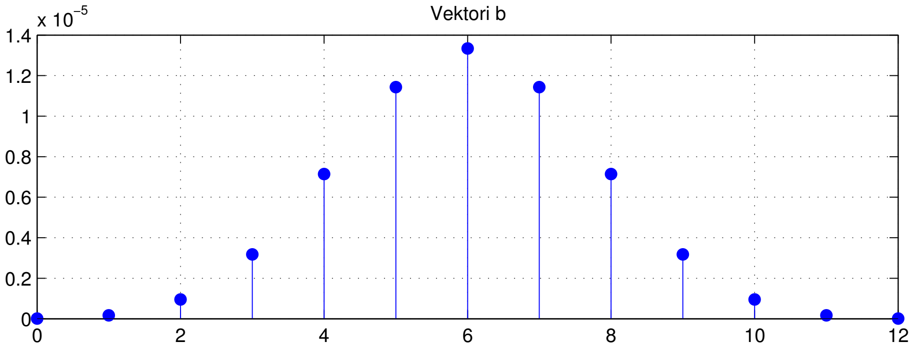
```
N = 12
Wn = 0.2200
```

- Suunnittelu tapahtuu komennolla:

```
[b,a] = cheby1 (N, 0.01, Wn);
```

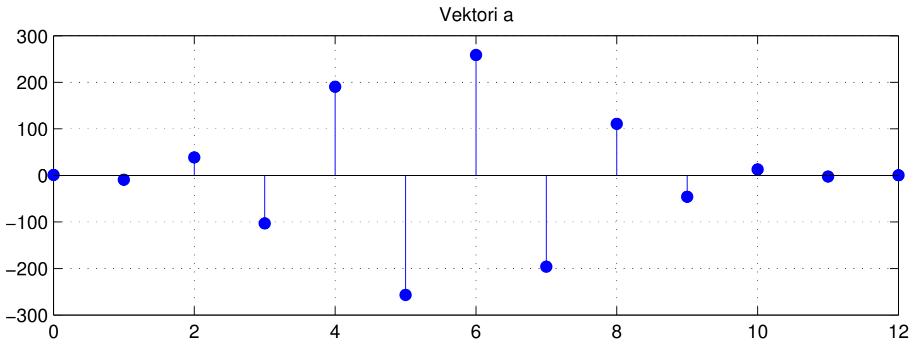
Tyypin I Chebyshev-suotimet

- Saadun suotimen kertoimet ovat vektoreissa b ja a, joiden kuvaajat ovat alla.



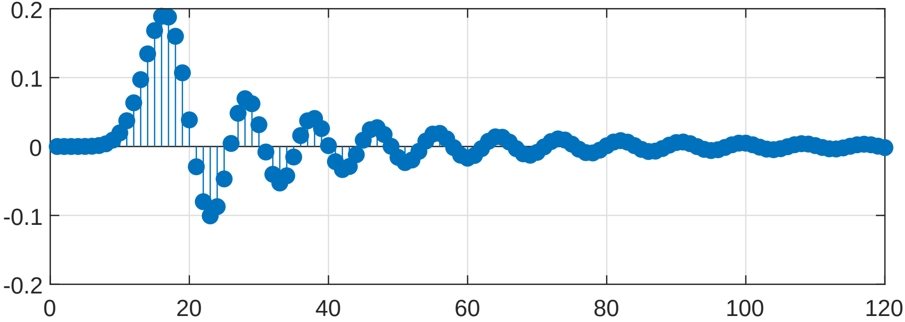
Tampereen yliopisto

Tyypin I Chebyshev-suotimet



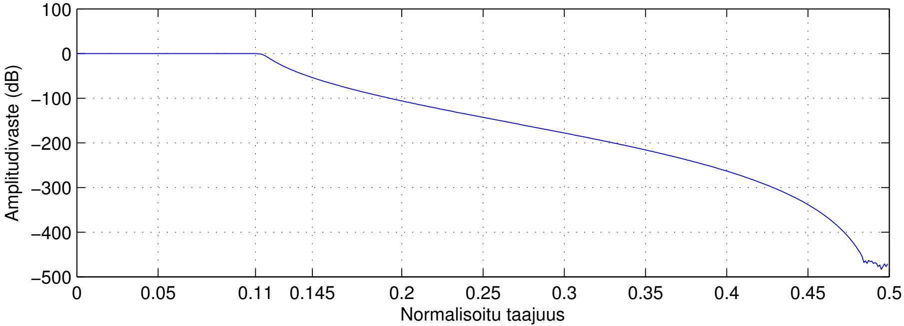
Tyypin I Chebyshev-suotimet

- Suotimen impulssivasteen alku on alla olevassa kuvassa.

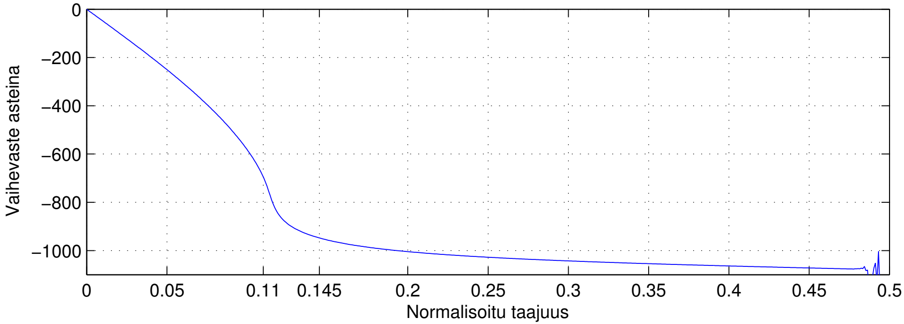


Tyypin I Chebyshev-suotimet

- Suotimen amplitudi- ja vaihevasteet ovat seuraavassa.



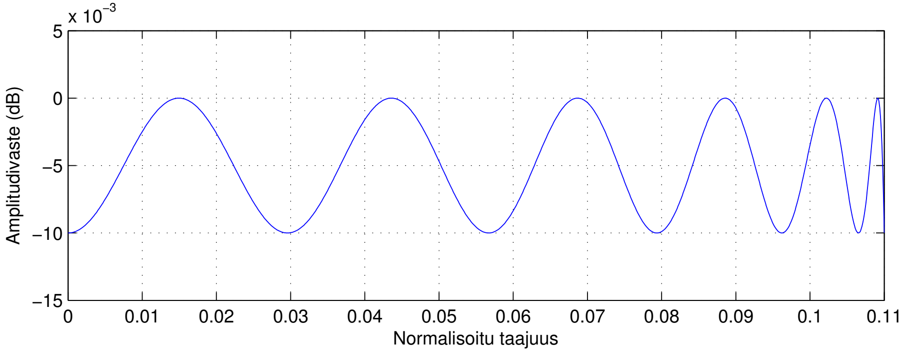
Tyypin I Chebyshev-suotimet



Tyypin I Chebyshev-suotimet

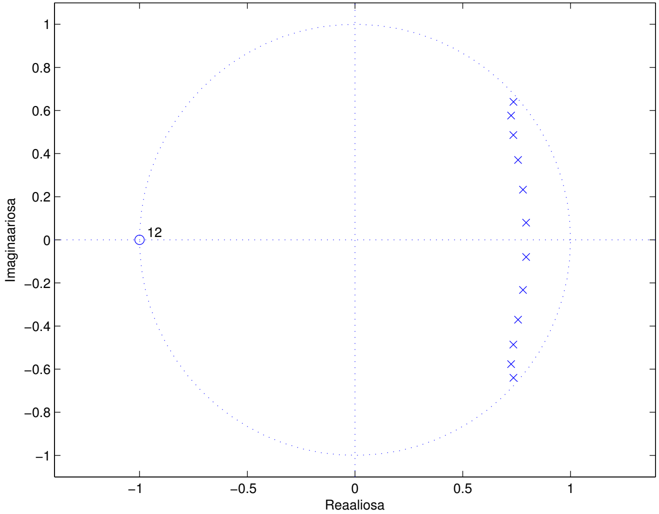
- Ylemmästä kuvasta nähdään, että suodin on estokaistalla samanlainen kuin Butterworth-suodinkin: molempien amplitudivaste lähestyy äärettömän desibelin vaimennusta Nyquistin rajataajuutta lähestyttäessä.
- Erona näillä kahdella suotimella on kuitenkin päästökaistan käyttäytyminen.
- Butterworth-suotimen amplitudivaste on tasainen laskeva funktio, mutta tyypin I Chebyshev-suotimen amplitudivaste värähtelee ideaaliarvon alapuolella (ks. kuva alla).

Tyypin I Chebyshev-suotimet



- Tässäkin tapauksessa `zplane` laskee suotimen navat ja nollat väärin, ja ne saadaan komennosta `cheb1` samoin kuin Butterworth-suotimen tapauksessakin.

Tyypin I Chebyshev-suotimet



Tyypin II Chebyshev-suotimet

- Toisen tyypin Chebyshev-suotimille on tyypillistä, että niiden estokaista on tasavärähtelevä (equiripple), mutta päästökaista on maksimaalisen tasainen (maximally flat).
- Tyypin I Chebyshev-suotimiin nähden siis päästö- ja estokaistan tyypit ovat vaihtaneet paikkaa.
- Tämänkin suodintyypin suunnittelussa on käytettävissä vastaavat komennot kuin edellä mainituissa tapauksissa.
- Suotimen aste voidaan arvioida komennolla

```
[N, Wn] = cheb2ord(2*Wp, 2*Ws, Rp, Rs);
```

- Suunnittelukomento on muotoa

```
[b,a] = cheby2(N,Rs,Wn);
```

Tyypin II Chebyshev-suotimet

- Edellisen esimerkin suodinta vastaava tyypin II Chebyshev-suodin suunnitellaan seuraavasti:

```
[N, Wn] = cheb2ord(0.22, 0.29, 0.01, 47)
```

- Tulokseksi saadaan seuraavat arvot:

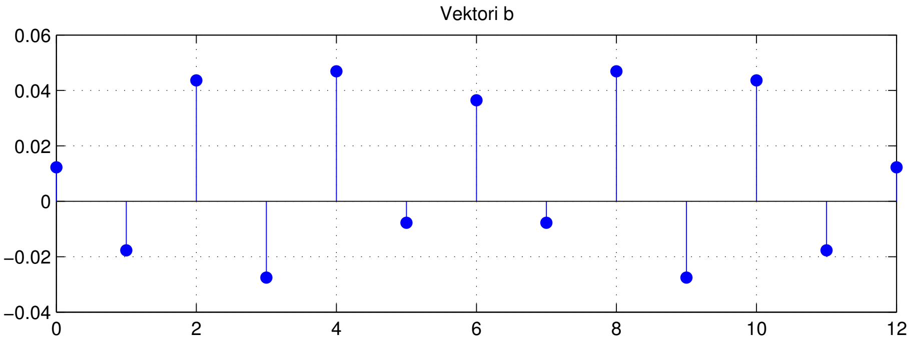
```
N = 12
Wn = 0.2795
```

- Varsinainen suunnittelu tapahtuu komennolla

```
[b,a] = cheby2 (N, 47, Wn);
```

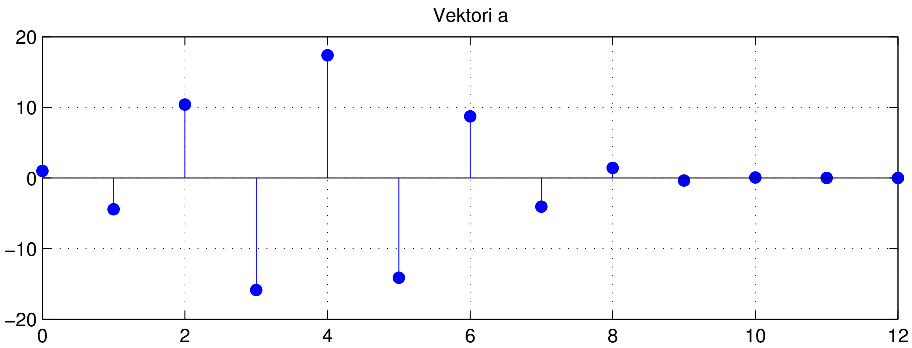
Tyypin II Chebyshev-suotimet

- Saadun suotimen kertoimet (vektorit b ja a) on kuvattu alla.



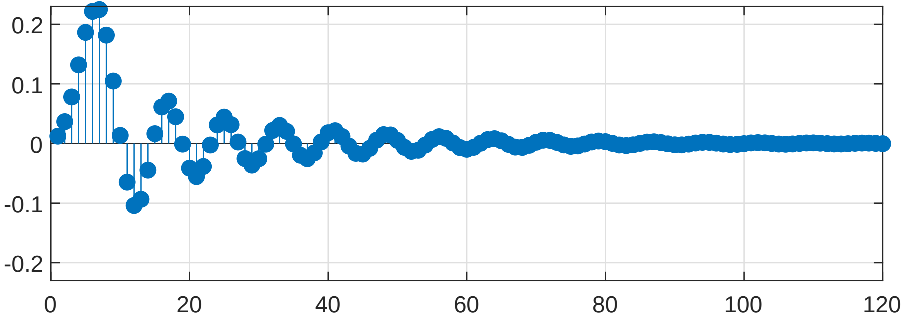
Tampereen yliopisto

Tyypin II Chebyshev-suotimet



Tyypin II Chebyshev-suotimet

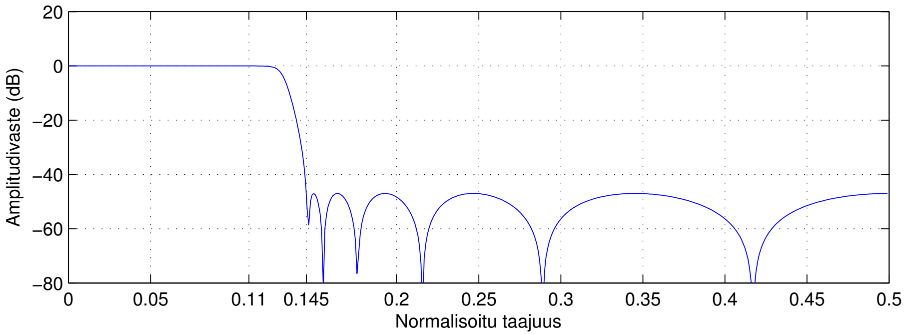
- Suotimen impulssivasteen alku näyttää hyvin samanlaiselta kuin muillakin suodintyypeillä.



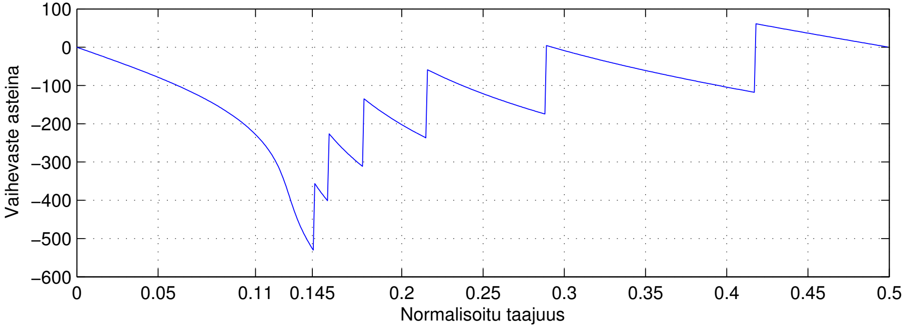
Tyypin II Chebyshev-suotimet

- Amplitudi- ja vaihevasteissa näkyy kuitenkin selkeä ero Butterworth-suotimiin ja tyypin I Chebyshev-suotimiin nähden.
- Nyt estokaista on näet tasavärähtelevä, mikä tarkoittaa kaikkien huippujen olevan samalla korkeudella.
- Päästökaista puolestaan on samanlainen kuin Butterworth-suotimella, eli se laskee monotonisesti taajuuden kasvaessa.

Tyypin II Chebyshev-suotimet



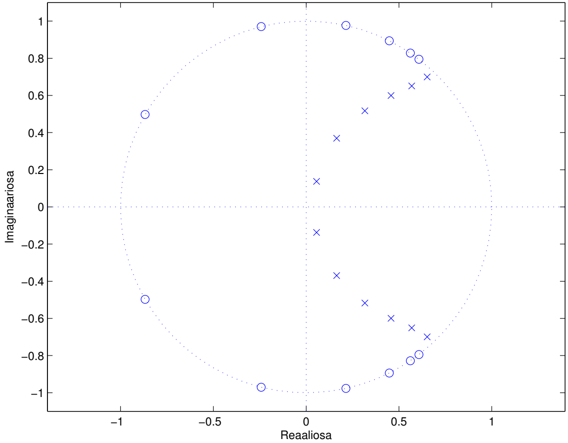
Tyypin II Chebyshev-suotimet



- Alla oleva napa-nollakuvio osittain selittää tasavärähtelyominaisuuden estokaistalla.

Tyypin II Chebyshev-suotimet

- Nollat ovat sijoittuneet sopivin välein estokaistalle, jolloin niiden yhteinen vaikutus tasapainottaa estokaistan värähtelyn.



Elliptiset suotimet eli Cauer-suotimet

- Elliptisille suotimille on ominaista, että **sekä** päästökaistan amplitudivaste **että** estokaistan amplitudivaste ovat tasavärähteleviä.
- Komennot suunnittelussa ovat

```
[N, Wn] = ellipord(2*Wp, 2*Ws, Rp, Rs);
```

ja

```
[b,a] = ellip(N,Rp,Rs,Wn);
```

Elliptiset suotimet eli Cauer-suotimet

- Esimerkkisuotimemme suunnittelu tapahtuu siis komennoilla

```
[N,Wn] = ellipord(0.22, 0.29, 0.01, 47)
```

- Tulokseksi saadaan:

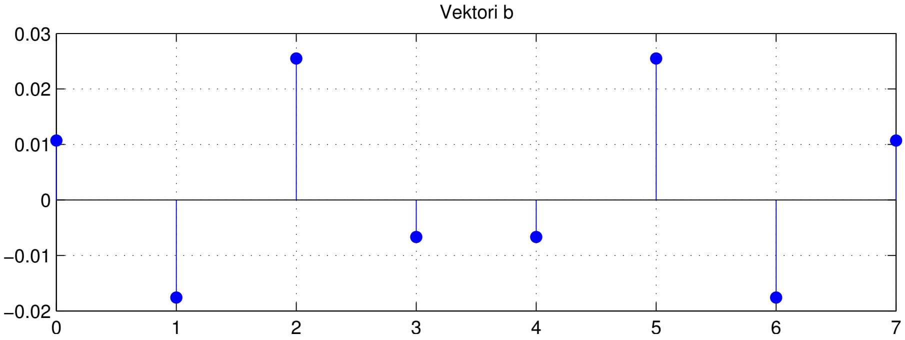
```
N = 7
Wn = 0.2200
```

- Seuraavaksi suunnittelukomento:

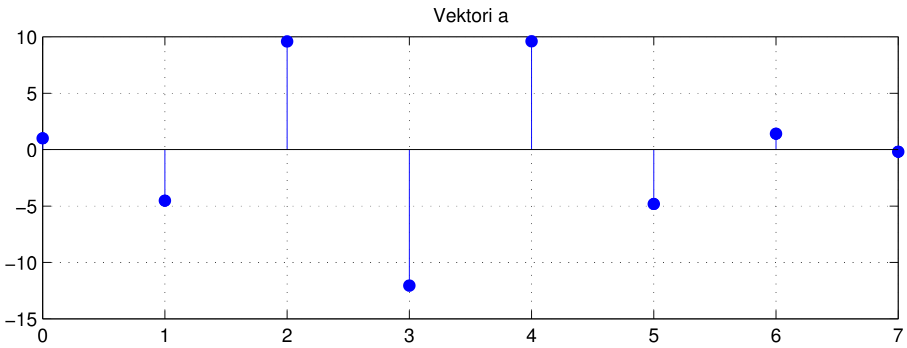
```
[b,a] = ellip (N, 0.01, 47, Wn);
```

Elliptiset suotimet eli Cauer-suotimet

- Saadun suotimen kertoimet (vektorit b ja a) ovat alla.

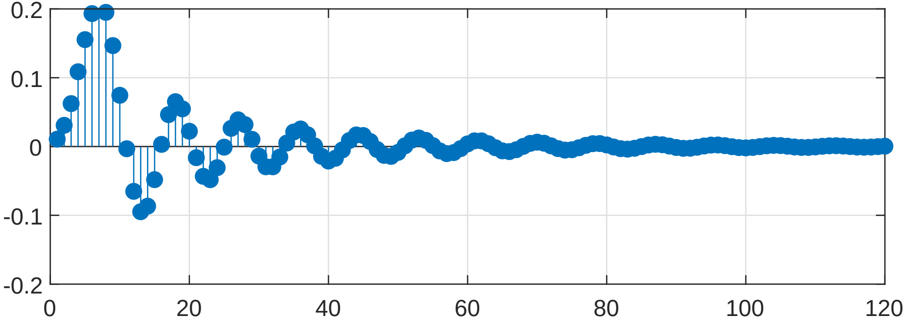


Elliptiset suotimet eli Cauer-suotimet



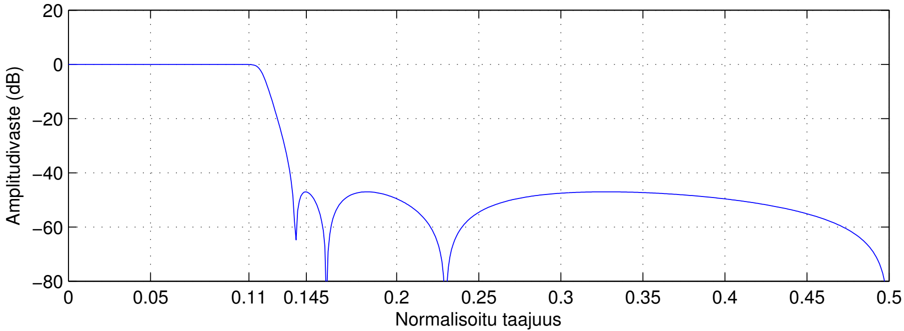
Elliptiset suotimet eli Cauer-suotimet

- Suotimen impulssivasteen alku näyttää jälleen hyvin samanlaiselta kuin muillakin suodintyypeillä.

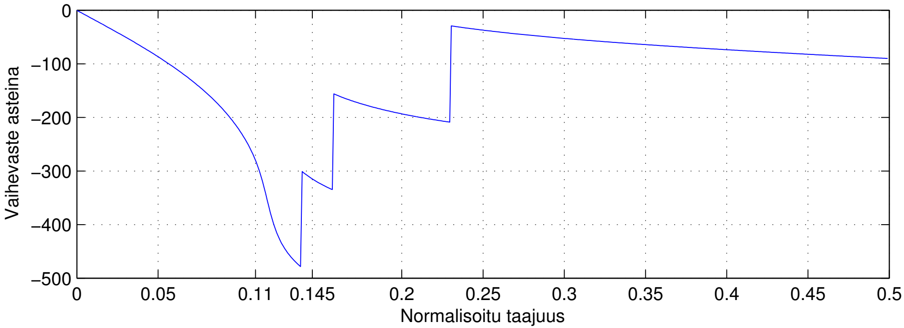


Elliptiset suotimet eli Cauer-suotimet

- Suotimen amplitudi- ja vaihevasteet näkyvät alla.

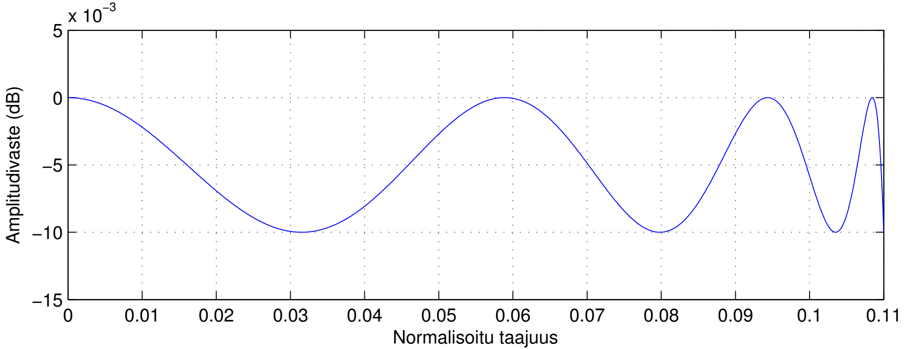


Elliptiset suotimet eli Cauer-suotimet



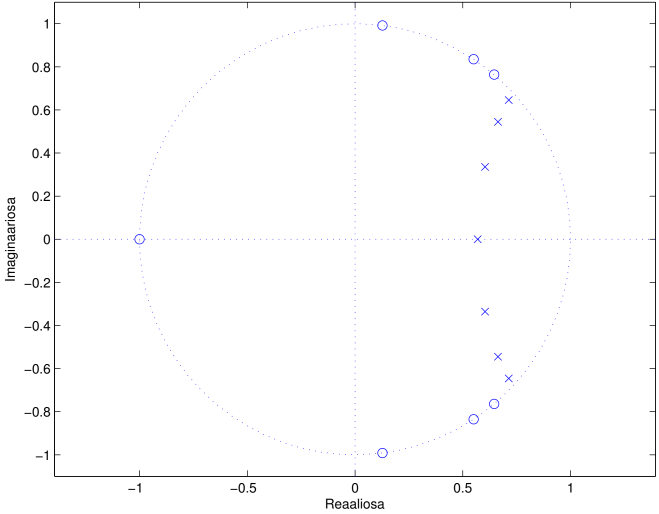
Elliptiset suotimet eli Cauer-suotimet

- Suotimen amplitudivaste värähtelee ideaaliarvon alapuolella (ks. kuva alla).



Elliptiset suotimet eli Cauer-suotimet

- Suotimen napa-nollakuvio on alla.



y=filter(b,a,x);

Suodintyyppien vertailua

- Alla olevassa taulukossa vertaillaan neljää IIR-suotimen tyyppiä ja niiden kertoimien määrää käsitellyssä esimerkissä.
- Molempien Chebyshev-suodinten kerrointen määrä on aina sama, Butterworth-suotimella on aina suurin määrä kertoimia ja elliptisillä suotimilla pienin.
- Vertailun vuoksi oikeanpuolimmaisessa sarakkeessa on vastaavan ikkunamenetelmällä suunnitellun FIR-suotimen kertoimien määrä, joka on selvästi IIR-suotimia suurempi.

	Butterworth	**Chebyshev I**	**Chebyshev II**	**Elliptinen**	**Blackman**
Värähtely päästökaistalla	ei	kyllä	ei	kyllä	kyllä
Värähtely estokaistalla	ei	ei	kyllä	kyllä	kyllä
Kerrointen määrä	$29+28$	$13+12$	$13+12$	$8+7$	159

Äärellisen sananpituuden vaikutukset

- Tähänastinen suodinten suunnittelu on tehty oletuksella, että käytössä on ääretön sananpituus ja laskentatarkkuus.
- Jopa Matlabilla suunniteltaessa saatavien suodinten kertoimet on esitetty verrattain suurella tarkkuudella.
- Kun suotimet käytännössä toteutetaan, täytyy kuitenkin kiinnittää huomiota käytettävän lukuesityksen vaikutukseen suodinten todelliseen käyttäytymiseen.
- Jos suotimet toteutetaan jollain korkean tason ohjelmointikielellä (C, Pascal, Fortran, Matlab) ja liukulukuaritmetiikkaa tukevalla suorittimella, pyöristysvirheet eri vaiheissa ovat suhteellisen pieniä.

Äärellisen sananpituuden vaikutukset

- Signaalinkäsittelyprosessorit eivät useimmiten laske liukuluvuilla, koska kiinteän pilkun aritmetiikan laskutoimitukset vaativat vähemmän resursseja (virtaa, tilaa, jäähdytystä, jne.) ja ne on halvempi toteuttaa.
- Tyypillinen signaalinkäsittelyprosessori toteuttaa laskutoimitukset kahdeksan, kahdentoista tai kuudentoista bitin tarkkuudella, joten pyöristysvirheisiin on syytä kiinnittää huomiota.

Äärellisen sananpituuden vaikutukset

- Digitaalisen IIR-suotimen tärkeimmät äärellisestä sananpituudesta johtuvat virhelähteet on lueteltu seuraavassa.
  - Kvantisointivirhe, joka syntyy muunnettaessa sisään tuleva analoginen signaali digitaaliseksi signaaliksi, jonka esityksessä käytetään verraten pientä bittimäärää.
  - IIR-suotimen kertoimien esitys äärellisellä bittimäärällä aiheuttaa muutoksia taajuusvasteessa ja stabiilisuusominaisuuksissa.
  - Äärellisen sananpituuden seurauksena järjestelmän laskutoimituksissa saattaa tulla ylivuotoa, jolloin tulokset menettävät merkityksensä lähes täysin.
  - Suodatettaessa tarvittavien kertolaskuoperaatioiden tulokset pyöristetään tai katkaistaan käytetyn sananpituuden mukaisesti.

Äärellisen sananpituuden vaikutukset

- Jos aritmetiikka on toteutettu **kahden komplementtiin** (two's complement) perustuen, niin ylivuototilanteessa esimerkiksi kahden suuren positiivisen luvun summaksi saadaan itseisarvoltaan suuri negatiivinen luku.
- Tämä luonnollisestikin sotkee koko suodatuksen, sillä tällainen suuri virhe säilyy ja kertautuu jatkossa IIR-suodinten rekursiivisen rakenteen vuoksi.

AD-muunnoksen kvantisointivirhe

- Muunnettaessa analoginen signaali digitaaliseksi on käytettävissä ainoastaan äärellinen määrä bittejä kunkin signaalin arvon esittämiseen.
- Merkitään käytettävää bittimäärää (merkkibittiä lukuunottamatta) muuttujalla b ja tarkastellaan tapausta, jossa diskreetin signaalin kaikki lukuarvot on jaettu tasavälisesti välille $[-1, 1]$.
- Jos käytössä on esimerkiksi seitsemän bittiä (+merkkibitti), voidaan kahden komplementtiaritmetiikalla esittää luvut $-128, -127, -126, \ldots, -1, 0, 1, \ldots, 125, 126, 127$.
- Nämä luvut skaalataan välille $[-1, 1]$ yksinkertaisesti jakamalla ne luvulla $2^7 = 128$. Näin saadaan kvantisointitasot $-\frac{128}{128}, -\frac{127}{128}, -\frac{126}{128}, \ldots, -\frac{1}{128}, \frac{0}{128}, \frac{1}{128}, \ldots, \frac{125}{128}, \frac{126}{128}, \frac{127}{128}$.

AD-muunnoksen kvantisointivirhe

- Nämä kolme esitysmuotoa tapauksessa $b = 2$ on esitetty alla olevassa taulukossa.

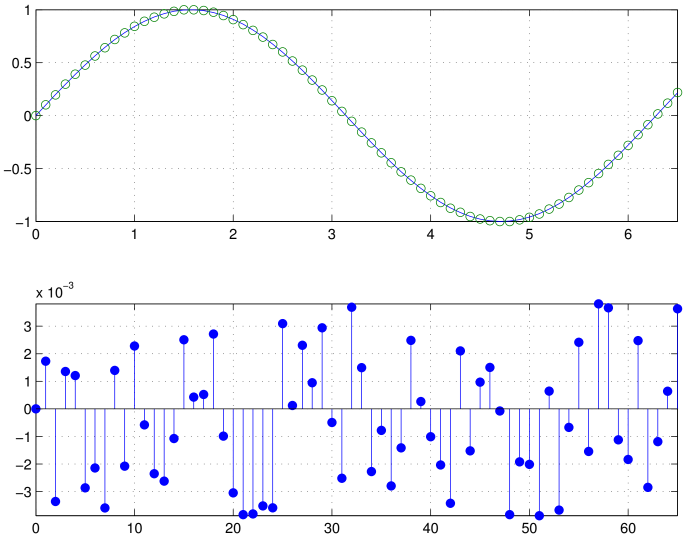
binääriesitys	100	101	110	111	000	001	010	011
desimaaliesitys	-4	-3	-2	-1	0	1	2	3
kvantisointitaso	$-\frac{4}{4}$	$-\frac{3}{4}$	$-\frac{2}{4}$	$-\frac{1}{4}$	$\frac{0}{4}$	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{3}{4}$

- Taulukon tapauksessa kahden kvantisointivälin etäisyys on $\frac{1}{4}$ ja yleisesti ottaen se on 2^{-b} käytettäessä b:tä bittiä (+merkkibittiä).

AD-muunnoksen kvantisointivirhe

- Kvantisoitaessa muodostuva virhe on enintään puolet tästä eli $2^{-b}/2$, jos käytetään pyöristystä lähimpään numeroon. (Ylin kvantisointitaso on poikkeus tästä, koska esimerkiksi taulukon tilanteessa pitää kaikki luvut väliltä $[\frac{3}{4}, 1]$ pyöristää alaspäin lukuun $\frac{3}{4}$. Tämä poikkeus jätetään yleensä huomiotta, koska suuremmilla bittimäärillä virheen suuruusluokka on hyvin pieni ja se myöskin esiintyy hyvin harvoin.)
- Tämä virhe $e(n)$ voidaan ajatella lisätyksi alkuperäiseen signaaliin $x(n)$, jolloin saadaan tulos
$$\hat{x}(n) = x(n) + e(n).$$
- Alla olevissa kuvissa on esitetty sinisignaali yhtenäisellä viivalla ja tulos kvantisoitaessa seitsemään bittiin (+merkkibittiin) ympyröillä.
- Alemmassa kuvassa on kvantisointivirhe $e(n)$.

AD-muunnoksen kvantisointivirhe



Tampereen yliopisto

AD-muunnoksen kvantisointivirhe

- Tyypillisesti kvantisointivirheestä $e(n)$ tehdään seuraavat oletukset tilastollista analyysiä varten.
  1. Signaali $e(n)$ on stationaarinen satunnaisprosessi, eli sen tilastolliset ominaisuudet eivät muutu ajan myötä.
  2. Signaalin $e(n)$ lukuarvot eivät riipu lukujonon $x(n)$ arvoista.
  3. Signaalin $e(n)$ arvot ovat riippumattomia toisistaan, eli $e(n)$ on **valkoista kohinaa** (white noise).
  4. Signaalin $e(n)$ arvot ovat jakautuneet tasaisesti välille $(-2^{-b}/2, 2^{-b}/2]$.
- Nämä oletukset ovat voimassa, jos signaali on riittävän satunnainen.
- Ne eivät pidä paikkaansa kaikille signaaleille $x(n)$: jos esimerkiksi $x(n)$ on yksikköaskel, niin useimmat edellä mainituista oletuksista eivät ole voimassa.

AD-muunnoksen kvantisointivirhe

- Näitä oletuksia käyttämällä saadaan kuitenkin johdettua yksinkertainen malli kvantisointivirheiden määrälle.
- Jos oletetetaan järjestelmän käyttävän pyöristystä lähimpään lukuun, niin silloin signaalin $e(n)$ odotusarvo (Kutakuinkin sama asia kuin lukujonon keskiarvo, s.o. $\lim_{n\to\infty} \frac{1}{2n+1} \sum_{k=-n}^{n} e(k)$.)

$$\mu_e = E[e(n)] = 0,$$

ja varianssi

$$\sigma_e^2 = E[(e(n) - \underbrace{\mu_e}_{=0})^2] = E[e(n)^2].$$

AD-muunnoksen kvantisointivirhe

- Odotusarvon (toisesta) määritelmästä (Jos tiedetään satunnaismuuttujan x jakauma $p(x)$, voidaan x:n odotusarvo laskea kaavasta $E[x] = \int_{-\infty}^{\infty} xp(x)\,dx$.) saadaan edelleen:

$$
\begin{aligned}
E[e(n)^2] &= \int_{-2^{-b}/2}^{2^{-b}/2} x^2 \frac{1}{2^{-b}}\,dx = 2^b \Big/_{-2^{-b}/2}^{2^{-b}/2} \frac{x^3}{3} \\
&= 2^b \left(\frac{2^{-3b}}{24} + \frac{2^{-3b}}{24} \right) = \frac{2^{-2b}}{12}.
\end{aligned}
$$

AD-muunnoksen kvantisointivirhe

- Kvantisoidun signaalin **signaali-kohinasuhde** (signal to noise ratio; SNR) määritellään signaalin tehon suhteena kohinan tehoon (desibeliasteikolla), ja se saadaan näiden varianssien suhteesta:

$$\text{SNR} = 10 \log_{10} \frac{\sigma_x^2}{\sigma_e^2}.$$

- Signaali-kohinasuhde on siis kvantisoinnissa b bittiin (+merkkibitti)

$$\begin{aligned} \text{SNR} &= 10 \log_{10} \frac{\sigma_x^2}{2^{-2b}/12} \\ &= 10 \log_{10}(\sigma_x^2) + 10 \log_{10}(12) + 2b \cdot 10 \log_{10}(2) \\ &\approx 10 \log_{10}(\sigma_x^2) + 10.79 + 6.02b. \end{aligned}$$

AD-muunnoksen kvantisointivirhe

- Signaali-kohinasuhde kasvaa siis noin 6 dB jokaista lisäbittiä kohden.
- Yllä olevaa kaavaa sievempään muotoon ei ole mahdollista päästä tekemättä oletuksia signaalista $x(n)$ ja sen varianssista σ_x^2.
- Kulutuselektroniikkatuotteissa halutaan yleensä mahdollisimman hyvä signaali-kohinasuhde ja siksi teknisissä tiedoissa ilmoitettu SNR lasketaan mahdollisimman suuriamplitudiselle signaalille.
- Tällaiseksi sopii esimerkiksi sini- tai kosinisignaali amplitudilla 1, esimerkiksi

$$x(n) = \cos(2\pi \cdot 0.25 \cdot n).$$

- Nyt signaalissa $x(n)$ toistuu jakso $\ldots, 1, 0, -1, 0, \ldots$, joten signaalin neliön odotusarvo on $\frac{1}{2}$.

AD-muunnoksen kvantisointivirhe

- Samaan tulokseen päästään muillakin taajuuksilla kuin 0.25.
- Oletuksella $\sigma_x^2 = \frac{1}{2}$ saadaan signaali-kohinasuhteeksi

$$\text{SNR} \approx 10\log_{10}(\frac{1}{2}) + 10.79 + 6.02b = 6.02b + 7.78.$$

- Esimerkiksi CD-soitin esittää näytteet 16 bitin tarkkuudella, joten tällä sinisignaalilla sen SNR$\approx 6.02 \cdot (16 - 1) + 7.78 \approx 98$ dB.
- Toinen yleinen oletus signaalin $x(n)$ varianssista on, että sen amplitudi on skaalattu johonkin vakioarvoon.
- Käytännön tilanteissa täytyy nimittäin varautua siihen, että analoginen signaali, josta näytteitä otetaan, saa ajoittain hyvinkin suuria arvoja.

AD-muunnoksen kvantisointivirhe

- Siksi sisääntulevan signaalin amplitudi on tapana kertoa jollain vakiolla A.
- Näin saatavan signaalin $Ax(n)$ varianssi on $A^2\sigma_x^2$, joten kvantisoinnin jälkeinen signaali-kohinasuhde on

$$\text{SNR} \approx 6.02b + 10.79 + 10\log_{10}(\sigma_x^2) + 20\log_{10}(A).$$

- Nyrkkisääntönä voidaan sanoa, että valitsemalla $A = 1/(4\sigma_x)$ käytännössä eliminoidaan liian suurien arvojen saapumisen mahdollisuus.
- Tällöin skaalatun signaalin $Ax(n)$ varianssi on $\sigma_x^2/(16\sigma_x^2) = \frac{1}{16}$ ja signaali-kohinasuhde on desibeleissä

$$\text{SNR} \approx 6.02b + 10.79 + 10\log_{10}(\frac{1}{16}) = 6.02b - 1.25.$$

AD-muunnoksen kvantisointivirhe

- Muitakin skaalaustermejä toki käytetään ja kullekin niistä voidaan johtaa oma SNR-kaava.
- Mitä pienempi skaalaustermi A on, sitä pienemmäksi tulee myös SNR. Jos esimerkiksi halutaan signaali-kohinasuhteeksi tällä skaalauksella yli 80 dB, niin pitää olla $6.02\mathrm{b} - 1.25 > 80$, eli

$$\mathrm{b} > \frac{80 + 1.25}{6.02} = 13.50.$$

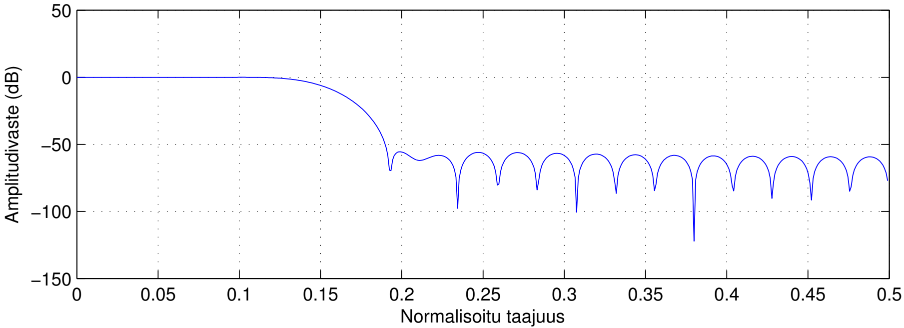
- Näin ollen riittää valita 14-bittinen esitys (+merkki).

Kertoimien pyöristämisen vaikutus

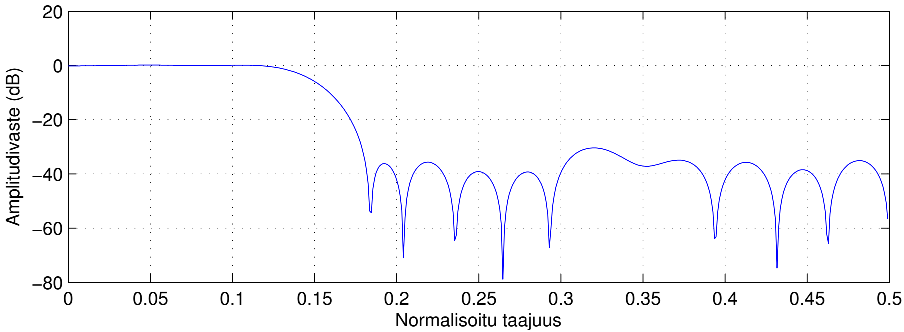
- Suodinten suunnittelumenetelmien tavoitteena on suunnitella paras mahdollinen suodin tietyille kriteereille.
- Kun suodin sitten toteutetaan pienemmällä bittimäärällä, helpointa on pyöristää kertoimet lähimpään kvantisointitasoon.
- Kertoimia pyöristettäessä suodin muuttuu optimaalisesta aina huonompaan päin, jolloin suodin ei välttämättä toteutakaan annettuja vaatimuksia.

Kertoimien pyöristämisen vaikutus

- Alla olevassa kuvassa on erään FIR-suotimen amplitudivaste ennen ja jälkeen kertoimien pyöristystä (1+7):n bitin tarkkuuteen.



Kertoimien pyöristämisen vaikutus

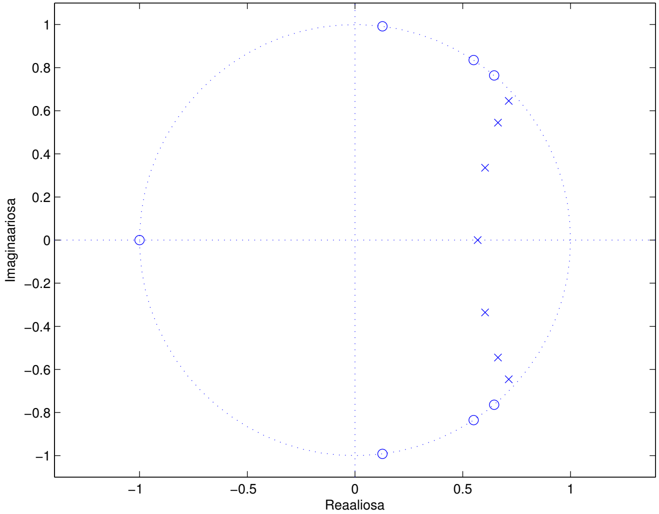


- Amplitudivaste heikkenee selvästi ja suotimen toiminta ei välttämättä ole enää määrittelyjen mukaista.

Kertoimien pyöristämisen vaikutus

- IIR-suotimen tapauksessa kvantisointi voi tuottaa vielä suuremman muutoksen suotimen toiminnassa: pahimmassa tapauksessa suotimesta tulee epästabiili.
- Alla olevassa kuvassa on edellisen kappaleen elliptisen IIR-suotimen napa-nollakuvio.

Kertoimien pyöristämisen vaikutus

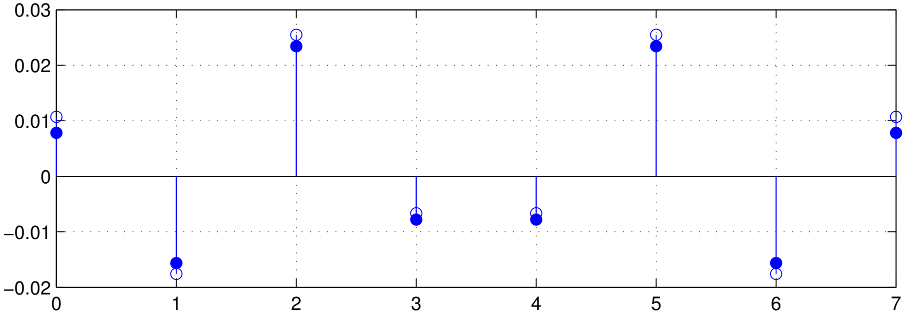


Kertoimien pyöristämisen vaikutus

- Kun suotimen kertoimet kvantisoidaan (1+7):n bitin tarkkuuteen, kertoimien muutos ei ole erityisen suuri.
- Alla olevassa kuvassa on kuvattu siirtofunktion osoittajan kertoimien muutos kvantisoitaessa.

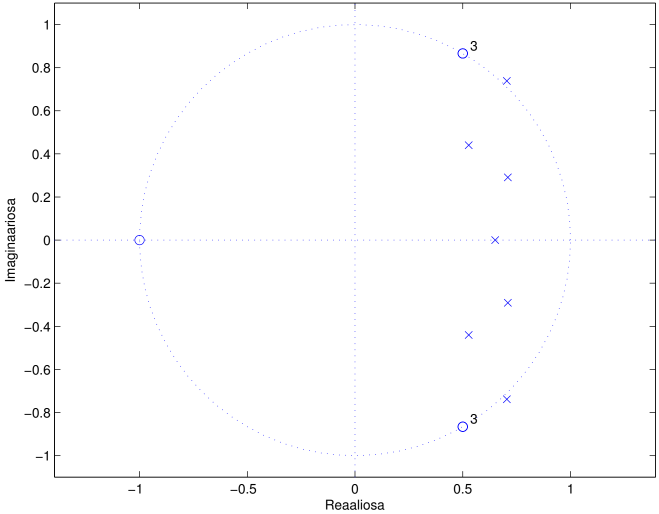
Kertoimien pyöristämisen vaikutus

- Alkuperäiset arvot on esitetty avoimilla ympyröillä ja kvantisoidut mustilla ympyröillä.



- Suotimen napoihin ja nolliin tämä vaikuttaa kuitenkin paljon enemmän.
- Alla olevasta kuvasta nähdään, että kvantisoidusta suotimesta tuli epästabiili.

Kertoimien pyöristämisen vaikutus



Kertoimien pyöristämisen vaikutus

- Useimmiten on olemassa parempiakin (1+7)-bittisiä suotimia kuin se, joka saadaan pyöristämällä kukin kerroin suoraan lähimpään (1+7)-bittiseen lukuun.
- Kertoimien muutoksen vaikutus yhteys ja nolliin ja suotimen amplitudivasteeseen on melko monimutkainen, joten suoraviivainen pyöristäminen ei välttämättä tuota parasta mahdollista amplitudivastetta.
- Kertoimien optimaaliseen sijoitteluun eri kvantisointitasoille perehdytään myöhemmillä signaalinkäsittelyn kursseilla.

Näytteenottotaajuuden muuntelu

- Näytteenottotaajuuden muuntelusta A/D-muunnoksen jälkeen käytetään englannin kielessä nimitystä **multirate DSP**, mutta sille ei ole olemassa lyhyttä suomenkielistä nimeä.
- Yksinkertaisin menetelmä näytteenottotaajuuden muuntamiseksi on tietysti muuntaa signaali ensin analogiseksi ja tämän jälkeen muuntaa se takaisin digitaaliseksi uudella näytteenottotaajuudella.
- Tämä menettely johtaa kuitenkin ylimääräisiin kvantisointi- eli pyöristysvirheisiin, joita on syytä välttää.
- Ongelma on mahdollista ratkaista pelkästään digitaalisen signaalinkäsittelyn keinoin.

Näytteenottotaajuuden muuntelu

- Tällöin saadaan taatusti optimaalinen tulos, jossa ei ole mukana ylimääräisiä kvantisointikohinoita.
- Ongelma on siis seuraava: on olemassa signaali, joka on muodostettu analogisesta signaalista näytteenottotaajuudella f.
- Tästä signaalista halutaan selvittää se digitaalinen signaali, joka on mahdollisimman lähellä sitä signaalia, joka olisi saatu näytteenottotaajuudella $\hat{f}$.
- Kouluesimerkki kyseisestä ongelmasta on muunnos CD-formaatista DAT-formaattiin.
- CD-levyillä data on nimittäin esitetty näytteenottotaajuudella 44.1 kHz.
- Sen sijaan DAT-nauhurin formaatti perustuu näytteenottotaajuuteen 48 kHz.
- Usein muunnos toki hoidetaan käytännössä muuntamalla signaali ensin analogiseksi ja edelleen digitaaliseksi taajudella 48 kHz.

Näytteenottotaajuuden muuntelu

- Parempaan tulokseen kuitenkin on mahdollista päästä multirate-signaalinkäsittelyn menetelmin.
- Taajuuden muuntaminen on tarpeellista myös, kun signaalista on syystä tai toisesta otettu näytteitä liian korkealla taajuudella.
- Tällöin siis signaalissa ei ole läheskään niin suuria taajuuksia kuin näytteenottotaajuus mahdollistaa.
- Kun kuitenkin näytteenottotaajuus on suuri, tulee suunniteltujen suodintenkin aste tarpeettoman korkeaksi.
- Pienempi aste saavutetaan, jos taajuudet muunnetaan ensin pienemmäksi ja suodatetaan vasta sitten.

Näytteenottotaajuuden muuntelu

- Tämän jälkeen signaali voidaan tarvittaessa muuntaa jälleen suurempaan taajuuteen.
- Muuntaminen käytännössä tapahtuu kahden perusoperaation avulla:
  - **Interpolointi** kasvattaa signaalin näytteenottotaajuutta lisäämällä ylimääräisiä arvoja,
  - **Desimointi** pienentää signaalin näytteenottotaajuutta poistamalla osan alkuperäisen signaalin arvoista.
- Desimointi ja interpolointi muuntavat näytteenottotaajuutta jollain kokonaislukukertoimella (esimerkiksi 1 kHz $\mapsto$ 2 kHz tai 1 kHz $\mapsto$ 0.5 kHz, jne).
- Näitä operaatioita yhdistelemällä saadaan aikaiseksi kaikkia rationaalikertoimia vastaavat taajuusmuunnokset.
- Tarkastellaan lähemmin muunnosta CD-formaatista DAT-formaattiin.

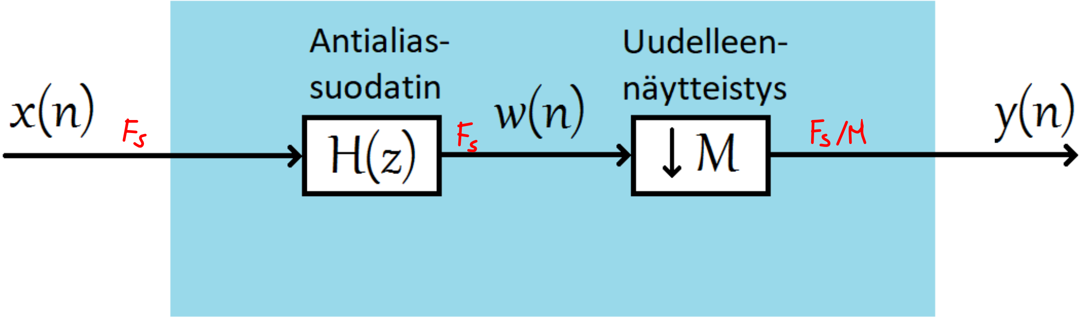
Näytteenottotaajuuden muuntelu

- Käytössä on 44.1 kHz ja tavoite olisi 48 kHz. Muunnoskerroin on $48/44.1 = 480/441 = 160/147$.
- Ensin taajuus korotetaan siis arvoon $44.1 \cdot 160 = 7056$ kHz interpoloimalla kertoimella 160.
- Tämän jälkeen signaali muunnetaan takaisin taajuuteen 48 kHz desimoimalla kertoimella 147.
- Merkittävää on, että muunnos tapahtuu juuri tässä järjestyksessä.
- Jos ensin olisi desimoitu, niin tuloksena olevassa signaalissa näytteenottotaajuus olisi ollut liian pieni, ja suurin osa signaalin informaatiosta olisi hävinnyt.
- Nyt tällaista informaatiohävikkiä ei tapahdu.

M=3

Desimointi eli näytteenottotaajuuden alentaminen

- Oheinen kaavio esittää signaalin $x(n)$ desimointiin liittyviä vaiheita.



Desimointi eli näytteenottotaajuuden alentaminen

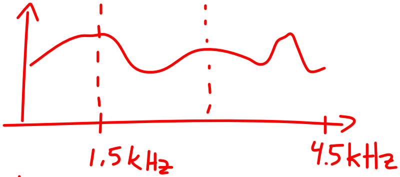
- Ennen varsinaista näytteenottotaajuuden pienentämistä signaali täytyy suodattaa alipäästösuotimella laskostumisen estämiseksi.
- Näin saadun signaalin näytteenottotaajuutta pienennetään jättämällä ainoastaan osa alkuperäisen signaalin arvoista jäljelle.
- Desimointioperaatiota merkitään alaspäin osoittavalla nuolella ja taajuuden muunnoskertoimella.
- Esimerkiksi näytteenottotaajuuden pudottamista kolmannekseen merkitsevä symboli on $\boxed{\downarrow 3}$.
- Kuvion tapauksessa alkuperäinen näytteenottotaajuus F_s pudotetaan desimoinnissa arvoon F_s/M.
- Tämä tehdään poimimalla joka M:s alkio desimoituun signaaliin.

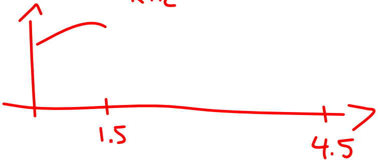
Desimointi eli näytteenottotaajuuden alentaminen

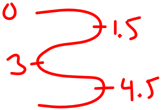
- Koska desimoitaessa näytteenottotaajuus pienenee, on laskostumisen vaara ilmeinen.
- Laskostuminenhan estetään ainoastaan poistamalla signaalista taajuudet, jotka ovat suurempia kuin puolet näytteenottotaajuudesta.
- Tämän suorittava suodin (digital anti-aliasing filter) on siis sellainen alipäästösuodin, joka poistaa kaikki arvoa $F_s/(2M)$ suuremmat taajuudet. $(F_s/M)/2 = (F_s/2)/M$
- Normalisoiduissa taajuuksissa ilmaistuna alipäästösuotimen suunnitteluvaatimukset ovat seuraavat:
  - Suotimen päästökaista on $[0, \frac{1}{2M} - \Delta f]$.
  - Suotimen estokaista on $[\frac{1}{2M}, \frac{1}{2}]$.
- Siirtymäkaistan leveys, Δf, määräytyy sovelluksen mukaan.

Desimointi eli näytteenottotaajuuden alentaminen

$F_s = 9\,\mathrm{kHz}$ $M=3$ $F_s/3 = 3\,\mathrm{kHz}$







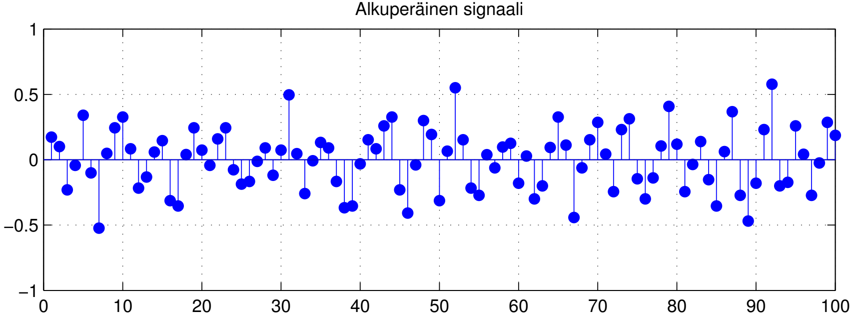
Desimointi eli näytteenottotaajuuden alentaminen

- Mitä kapeammaksi siirtymäkaista halutaan, sitä enemmän kertoimia suotimessa tulee olla.
- Myös vaimennusvaatimukset riippuvat sovelluksesta.
- Kaavoina signaalin $x(n)$ desimointiprosessi kertoimella M signaaliksi $y(n)$ voidaan ilmaista seuraavasti:

$$
\begin{aligned}
w(n) &= \sum_{k=-\infty}^{\infty} h(k)x(n-k), \\
y(n) &= w(nM) = \sum_{k=-\infty}^{\infty} h(k)x(nM-k).
\end{aligned}
$$

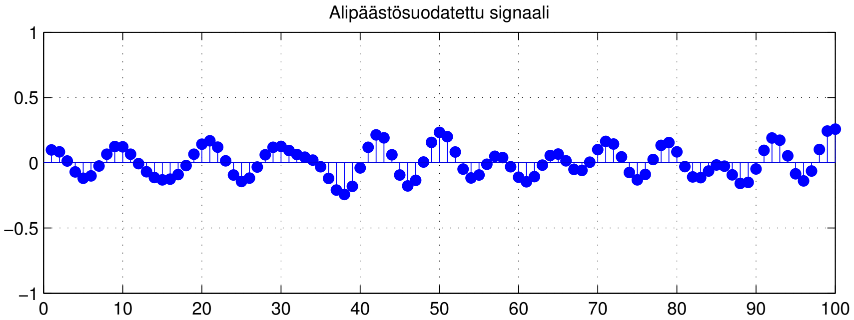
Desimointi eli näytteenottotaajuuden alentaminen

- Alla olevat kuvat esittävät desimoinnin vaikutusta aikatasossa.



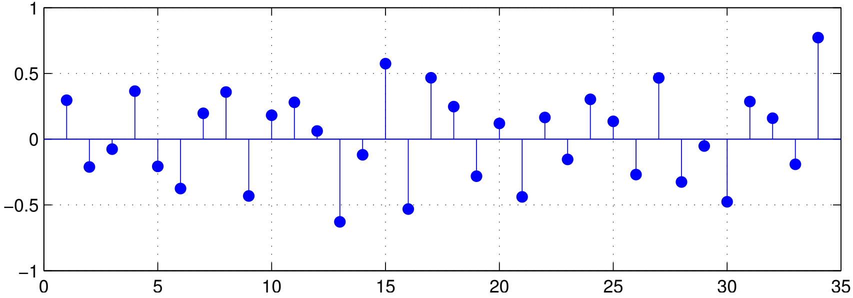
Desimointi eli näytteenottotaajuuden alentaminen

- Alkuperäinen signaali $x(n)$ suodatetaan ensin alipäästösuotimella, jolloin saadaan signaali $w(n)$, joka on valmis desimoitavaksi.



Desimointi eli näytteenottotaajuuden alentaminen

- Desimoitaessa kertoimella 3 otetaan uuteen signaaliin mukaan ainoastaan joka kolmas arvo.

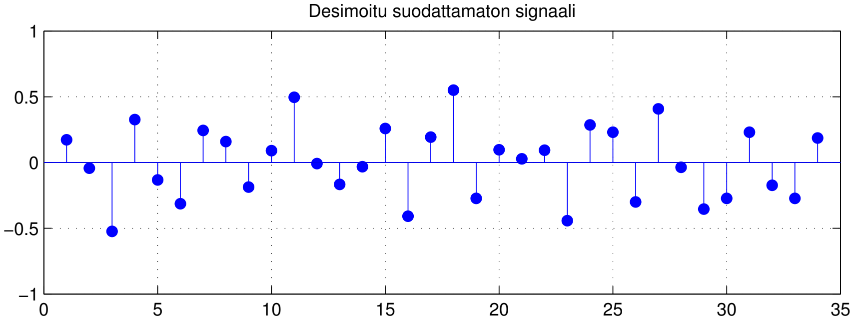


Desimointi eli näytteenottotaajuuden alentaminen

- Edellisessä kuvassa näytteet on kerrottu desimointikertoimella $M = 3$, jotta näytteiden suuruusluokka olisi sama kuin alkuperäisellä signaalilla.
- Ilman tätä kertolaskua näytteiden lukuarvot olisivat selvästi alkuperäisiä pienemmät, koska alipäästösuodatuksessa signaalin energia putoaa noin kolmannekseen.
- Yleensä tämä kertolasku jätetään pois kaavoista, koska se on helpointa toteuttaa suunnittelemalla suodin, jonka amplitudivaste päästökaistalla on M.
- Tällainen suodin saadaan normaalista alipäästösuotimesta yksinkertaisesti kertomalla sen jokainen kerroin luvulla M.

Desimointi eli näytteenottotaajuuden alentaminen

- Vertailun vuoksi mukana on myös signaali, joka on saatu desimoimalla ilman alipäästösuodatusta.

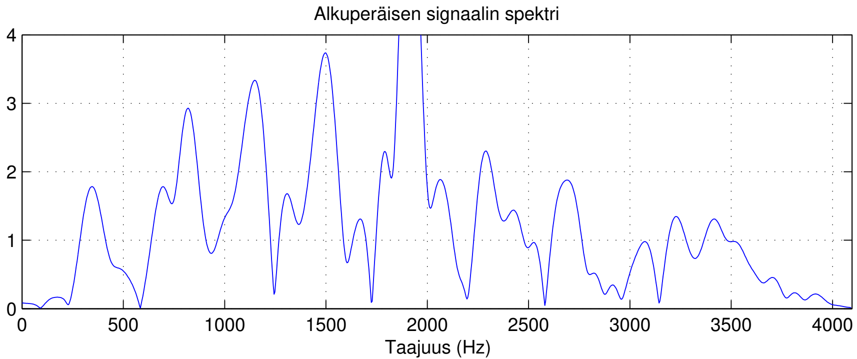


Desimointi eli näytteenottotaajuuden alentaminen

- Desimointiprosessi selvinnee helpommin taajuustason kuvaajista.
- Ensimmäinen kuvaaja esittää alkuperäisen signaalin $x(n)$ spektriä (diskreetin Fourier-muunnoksen itseisarvoa sopivasti ikkunoituna ja interpoloituna) $|X(n)|$.
- Esimerkin tapauksessa näytteenottotaajuus on 8192 Hz.
- Tällöin siis suurin signaalin sisältämä taajuus on 4096 Hz.

Tampereen yliopisto

Desimointi eli näytteenottotaajuuden alentaminen

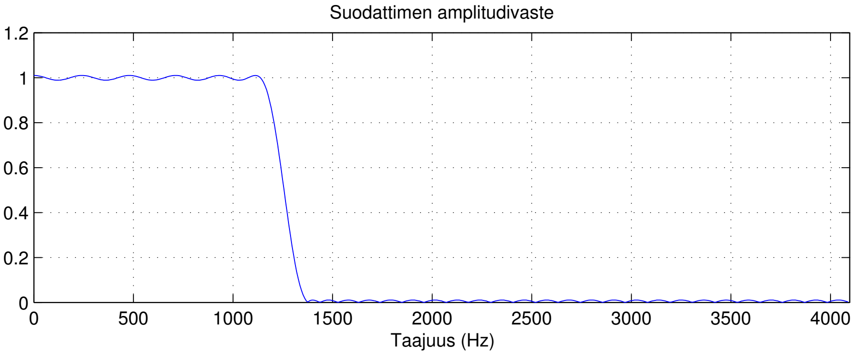


Desimointi eli näytteenottotaajuuden alentaminen

- Kun tätä signaalia halutaan desimoida kertoimella 3, täytyy ensin poistaa $1365\frac{1}{3}$ Hz suuremmat taajuudet.
- Tätä varten suunnitellaan alipäästösuodin, jonka estokaista on väli $[\frac{1}{6}, \frac{1}{2}]$ (näytteenottotaajuuden suhteen normalisoituina taajuuksina) ja päästökaista nollasta johonkin lukua $1/6$ pienempään lukuun, riippuen kuinka paljon kertoimia on varaa käyttää.
- Oheisessa esimerkissä päästökaistan rajataajuudeksi valittiin 0.14, päästökaistan maksimivärähtelyksi 0.09 dB (0.01 lineaarisella asteikolla) ja estokaistan minimivaimennukseksi 40 dB (0.01 lineaarisella asteikolla).

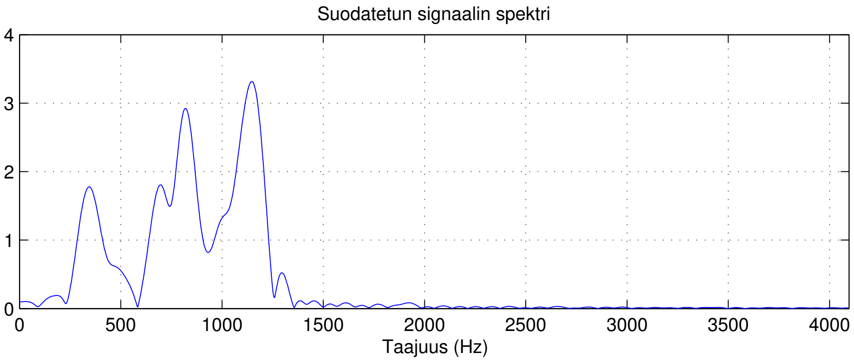
Desimointi eli näytteenottotaajuuden alentaminen

- Tämän suotimen amplitudivaste on kuvattu oheisessa kuvassa.



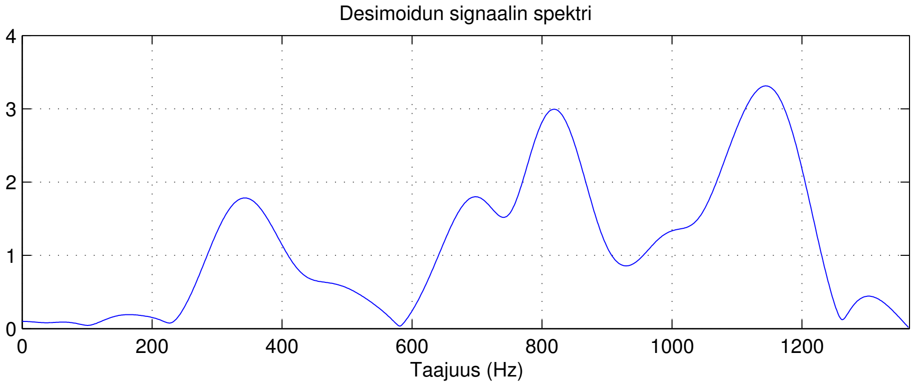
Desimointi eli näytteenottotaajuuden alentaminen

- Tällä suotimella suodatettaessa saadaan signaali $w(n)$, jonka spektri on alla.



Desimointi eli näytteenottotaajuuden alentaminen

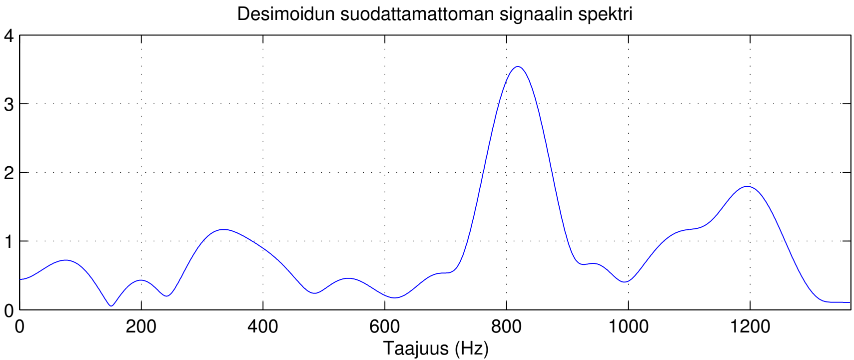
- Kun signaalista $w(n)$ jätetään jäljelle vain joka kolmas arvo (ja kerrotaan näytteet luvulla 3), tuloksena on signaali $y(n)$, jonka spektri on seuraavassa kuvassa:



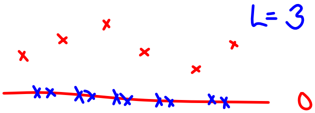
Desimointi eli näytteenottotaajuuden alentaminen

- Nyt siis uusi näytteenottotaajuus on $8192/3$ Hz.
- Viimeinen kuvaaja esittää suodattamattoman desimoidun signaalin spektriä, jossa laskostumisilmiö on selvästi havaittavissa.
- Siinä oleva spektri poikkeaa alkuperäisen signaalin vastaavasta kaistasta, vaikka ne halutaan mahdollisimman lähelle toisiaan.

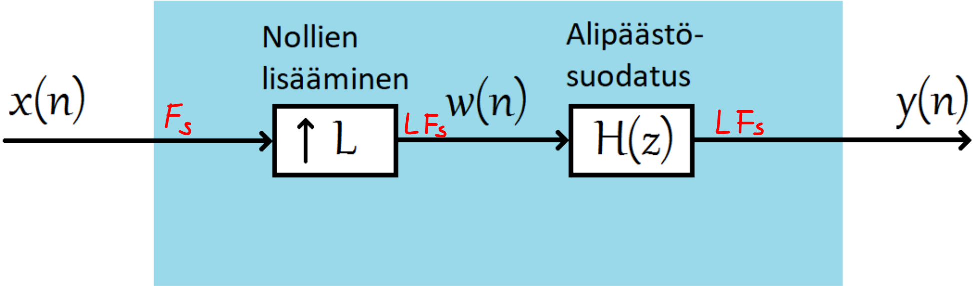
Desimointi eli näytteenottotaajuuden alentaminen



Interpolointi eli näytteenottotaajuuden nostaminen



- Signaalin interpoloinnin tarkoituksena on saada aikaiseksi signaali, joka vastaa alkuperäistä, mutta jonka näytteenottotaajuus on suurempi.
- Oheinen kaavio esittää lohkokaaviota interpolointioperaatiosta.



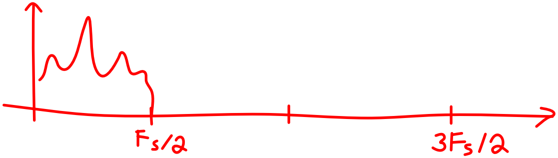
Interpolointi eli näytteenottotaajuuden nostaminen

- Toisin kuin desimoinnissa, nyt ei tarvita esisuodatusta, koska näytteenottotaajuutta nostettaessa kaikki alkuperäiset taajuudet voidaan toki esittää.
- Ensimmäinen operaatio on näytteenottotaajuuden nostaminen L-kertaiseksi.
- Tätä operaatiota merkitään nuolella ylöspäin ja interpolointikertoimella L, siis $\boxed{\uparrow L}$.
- Tämän operaation toteuttamisessa on lukuisia vaihtoehtoja; kuinka määritetään uudet ylimääräiset arvot?
- Matemaatikko alkaisi tässä tapauksessa luultavasti sovittaa polynomeja tai splinejä saadakseen uusia arvoja olemassaolevien arvojen välille.

Interpolointi eli näytteenottotaajuuden nostaminen

- Signaalinkäsittelyssä tilanne hoidetaan kuitenkin toisin: tuntemattomien arvojen tilalle sijoitetaan nollat ja näin saatu signaali suodatetaan alipäästösuotimella.
- Nollia lisättäessä mukaan tulee hyvin suuria taajuuksia.
- Suurten taajuuksien lisäämisen vaikutus on poistettava, ja se luonnollisesti tapahtuu alipäästösuodatuksella.
- Alkuperäisessä signaalissa suurin taajuus on $F_s/2$.
- Interpoloitaessa näytteenottotaajuuteen LF_s, suurin esitettävissä oleva taajuus on $LF_s/2$.
- Taajuutta $F_s/2$ suuremmat taajuudet on poistettava, koska ne ovat nollien lisäämisen tulosta.

$L=3$



Interpolointi eli näytteenottotaajuuden nostaminen

- Nollien lisäämisen jälkeen saatu signaali (näytteenottotaajuus LF_s) suodatetaan siis alipäästösuotimella, joka säilyttää taajuudet väliltä $[0, F_s/2]$ ja poistaa tätä suuremmat taajuudet.
- Koska nyt näytteenottotaajuus on interpoloinnin seurauksena LF_s, niin suotimen vaatimukset on normalisoitava tämän luvun suhteen.
- On siis suunniteltava alipäästösuodin, jonka vaatimukset ovat:
    - päästökaista on normalisoituina taajuuksina ilmaistuna väli $[0, (F_s/2)/(LF_s) - \Delta f] = \left[0, \frac{1}{2L} - \Delta f\right]$.
    - Siirtymäkaistan leveys Δf riippuu jälleen sovellutuksesta jossa interpolointia on tarkoitus soveltaa (Δf määrää osaltaan kertoimien määrän).
    - estokaista on väli $[(F_s/2)/(LF_s), 1/2] = \left[\frac{1}{2L}, \frac{1}{2}\right]$.
- Myös vaimennusvaatimukset määräytyvät enimmäkseen sovellutuksen mukaan, eikä niiden valinnasta voida antaa mitään yleistä ohjenuoraa.

Tampereen yliopisto

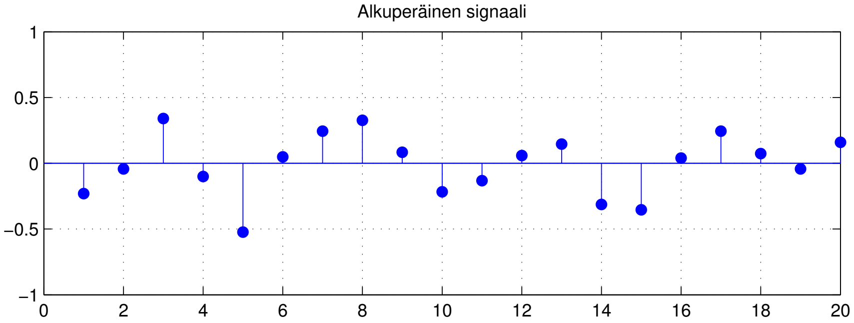
Interpolointi eli näytteenottotaajuuden nostaminen

- Koska alkuperäiseen signaaliin sijoitetaan $L-1$ nollaa jokaista signaalin arvoa kohden, signaalin amplitudi putoaa suodatettaessa yhteen L:n osaan.
- Siksi suodatuksen jälkeen (tai sitä ennen) signaalin arvot on syytä kertoa luvulla L.
- Kaavoilla esitettynä signaalin $x(n)$ interpolointiprosessi kertoimella L signaaliksi $y(n)$ on seuraava:

$$
\begin{aligned}
w(n) &= \begin{cases} x(n/L), & \text{kun } n = 0, \pm L, \pm 2L, \ldots, \\ 0, & \text{muulloin,} \end{cases} \\
y(n) &= \sum_{k=-\infty}^{\infty} h(k) w(n-k).
\end{aligned}
$$

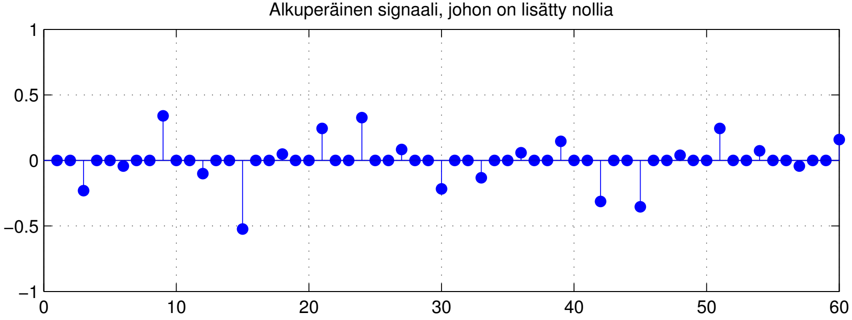
Interpolointi eli näytteenottotaajuuden nostaminen

- Alla oleva kuva esittää esimerkkitapausta, jossa signaalin näytteenottotaajuus interpoloidaan kolminkertaiseksi $(L = 3)$.



Interpolointi eli näytteenottotaajuuden nostaminen

- Ensin signaaliin $x(n)$ jokaisen kahden peräkkäisen arvon väliin sijoitetaan $L - 1 = 2$ nollaa, jolloin saadaan signaali $w(m)$.

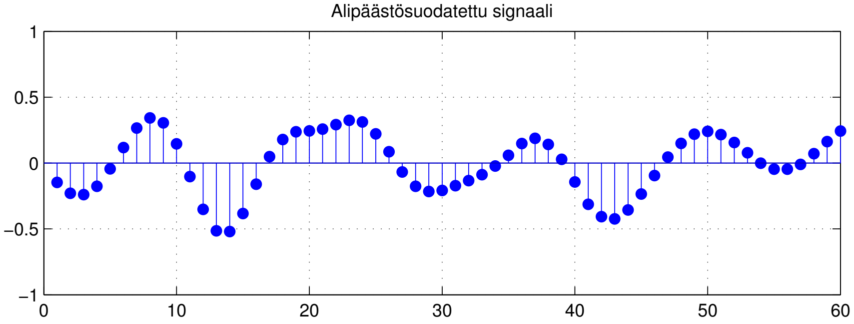


Interpolointi eli näytteenottotaajuuden nostaminen

- Tämä suodatetaan alipäästösuotimella, jonka estokaista on $[\frac{1}{6}, 0.5]$, päästökaista $[0, 0.1526]$ ja maksimivärähtelyarvot samat kuin desimointiesimerkissä.
- Tuloksena saadaan ulostulosignaali $y(m)$.
- Kuten desimoinnin tapauksessakin, tämä signaali on lopuksi vielä kerrottava luvulla $L = 3$, jotta signaalin energia säilyisi.

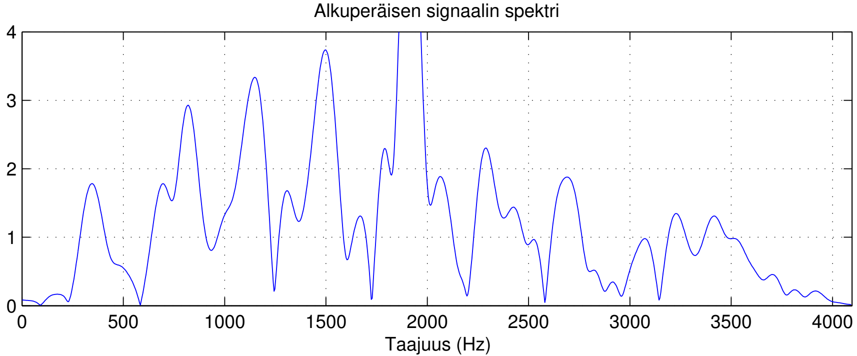
Interpolointi eli näytteenottotaajuuden nostaminen

- Vaihtoehtoisesti käytetyn alipäästösuotimen kertoimet voidaan kertoa luvulla L, jolloin päästökaistan taajuudet vahvistuvat kolminkertaisiksi.



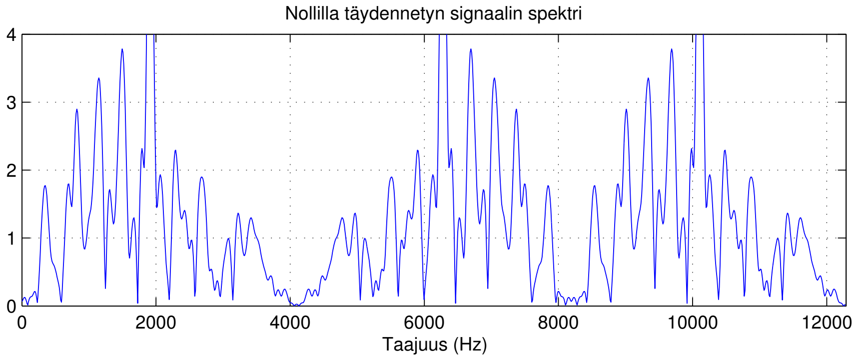
Interpolointi eli näytteenottotaajuuden nostaminen

- Kuva esittää taajuustasossa alkuperäistä signaalia, jonka näytteenottotaajuus on esimerkissämme 8192 Hz.



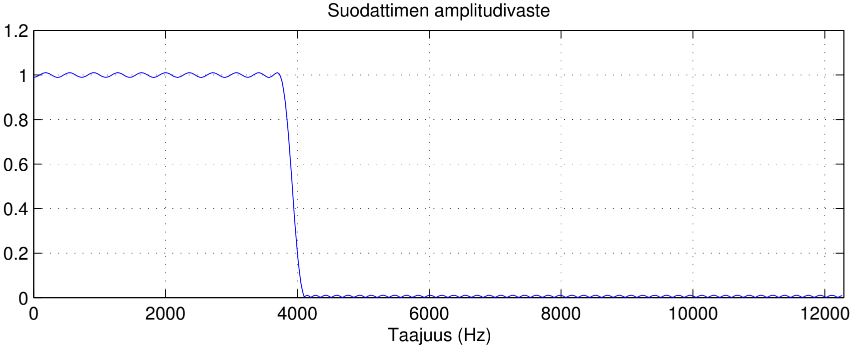
Interpolointi eli näytteenottotaajuuden nostaminen

- Lisättäessä signaaliin nollia, saavutetaan signaali, jonka spektri $|W(e^{i\omega})|$ on seuraavassa kuvassa.



Interpolointi eli näytteenottotaajuuden nostaminen

- Tästä on poistettava välillä 4096 Hz – 12288 Hz olevat taajuudet esimerkiksi suotimella, jonka amplitudivaste $|\mathrm{H}(e^{i\omega})|$ on seuraavassa kuvassa.



Interpolointi eli näytteenottotaajuuden nostaminen

- Tuloksena on signaali $y(n)$, jonka spektri on alla.