

Digitale signaalverwerking

Practica

Jan Meel

Inge Coudron

Klik en typ de subtekst

Inhoud

INLEIDING	5
<i>Doelstelling</i>	<i>5</i>
<i>Didactische Materiaal.....</i>	<i>5</i>
<i>Planning.....</i>	<i>5</i>
<i>Evaluatie</i>	<i>5</i>
<i>Goede labo-praktijk.....</i>	<i>6</i>
A. PRINCIPES EN CONCEPTEN VAN DIGITALE (DISCRETE TIJD)	
SIGNAALVERWERKING.....	7
<i>INLEIDING</i>	<i>7</i>
<i>DOEL 9</i>	
A.1 <i>Sampling en quantisatie</i>	<i>9</i>
A.2 <i>Convolutie.....</i>	<i>14</i>
A.3 <i>Fast Fourier Transform.....</i>	<i>17</i>
A.4 <i>Windowing</i>	<i>20</i>
A.5 <i>Ruis – windowing – interpolatie</i>	<i>22</i>
B. INTRODUCTIE VAN DIGITALE NETWERKEN EN DE Z-TRANSFORMATIE.	
ANALYSE VAN FILTERS.	24
<i>INLEIDING</i>	<i>24</i>
<i>DOEL 25</i>	
B.1 <i>1^e orde IIR.....</i>	<i>26</i>
B.2 <i>2^e orde IIR.....</i>	<i>30</i>
B.3 <i>2^e orde FIR</i>	<i>33</i>
C. SYNTHESE VAN DIGITALE FILTERS.	36
<i>INLEIDING</i>	<i>36</i>
<i>DOEL 37</i>	
C.1 <i>Pole-zero plaatsing</i>	<i>37</i>
C.2 <i>Bilineaire transformatie</i>	<i>38</i>
C.3 <i>(facultatief): Chebyshev-filter.....</i>	<i>40</i>
C.4 <i>Cauer-filter</i>	<i>40</i>
C.5 <i>Realisatie met 2^{de} orde-trappen.....</i>	<i>41</i>
C.6 <i>FIR-filter met lineaire fase</i>	<i>42</i>
C.7 <i>(Facultatief): FIR-filter met Parks / McClellan.....</i>	<i>43</i>

INLEIDING

Doelstelling

De student kan zelfstandig een digitale signaalverwerkende schakeling op een systematische wijze analyseren en ontwerpen.

Didactische Materiaal

- Cursustekst "Digitale Signaalverwerking", J. Meel
- MATLAB help

Planning

De verdeling van de opdrachten over de 6 labozittingen van elk 2 uur (labo nr. $i = L_i$):

Opdrachten	L1	L2	L3	L4	L5	L6
A. Signalen						
B. Analyse						
C. Synthese						

Deliverables	L1	L2	L3	L4	L5	L6	deadline
. elektronisch verslag A + code A							<i>dag vóór L3</i>
. elektronisch verslag B + code B							<i>dag vóór L5</i>
. elektronisch verslag C + code C							<i>vrijdag laatste lesweek</i>
Test							<i>1^e uur van L6</i>

Elektronische verslagen en code's in te dienen via Toledo.

Evaluatie

De evaluatie van het laboratorium gebeurt op basis van volgende elementen:

- Elektronische verslag (.pdf) en codes (.m) [30%]
- Test [70%]
 - Inzicht in de digitale signaalverwerkende theorie en praktijk behandeld in opdrachten A, B en C1.

Evaluatiecriteria:

- kwalitatieve en gedocumenteerde code (MATLAB/SPICE)
- inzicht (test)
- functionele code (MATLAB)
- voortgang (tijdig afwerken van de opdrachten)

- verslag (correct, duidelijk, volledig, opgegeven volgorde gerespecteerd, verzorgde layout)

Goede labo-praktijk

- **Laboratorium voorbereiden:** MATLAB-code uitschrijven.
- **Versiesysteem:** Houd een kopij bij van verschillende versies van de MATLAB-code.
- **Backup:** Bewaar op verschillende media een backup van je code (labo-computer, memory stick, je computer thuis). Het falen van 1 van deze media kan geen reden zijn om laattijdig een opdracht af te werken.

A. PRINCIPES EN CONCEPTEN VAN DIGITALE (DISCRETE TIJD) SIGNAALVERWERKING.

INLEIDING

DSP (Digital Signal Processing) is een van de snelst groeiende domeinen van de moderne elektronica . We vinden DSP-technieken terug op tal van plaatsen waar informatie onder digitale vorm wordt verwerkt of bewerkt d.m.v. digitale procesoren. De toepassingsgebieden zijn dan ook zeer ruim:

- Image processing : patroonherkenning, facsimile, satellietweerkarten ...
- Instrumentatie / controle : spectrumanalyse, ruisonderdrukking, datacompressie ...
- Speech / audio : spraakherkenning, spraaksynthese, digitale audio ...
- Militair : radar en sonar, communicatie-beveiliging, raketbesturing ...
- Telecommunicatie: echo cancellation, adaptieve filtering, spread spectrum, videoconferentie, datacommunicatie ...
- Medisch : scanners, EEG analyse, patiëntbewaking ...

Vele DSP-toepassingen vergen een verwerking van digitale signalen die werden bekomen door het nemen van samples van een analoog signaal. Na het verwerkingsproces worden de samples opnieuw tot een analoog signaal gereconstrueerd. De samples worden ook nog gequantiseerd omwille van de digitale voorstelling (eindige hoeveelheid bits). Beiden sampling en quantization leiden tot informatie verlies van het oorspronkelijke analoge signaal. Een belangrijk concept binnen de digitale verwerking van analoge signalen is dat het gesampled signaal een unieke voorstelling geeft van zijn overeenkomstig analoog signaal. Hiervoor dient de samplerate de zogenaamde Nyquist rate te overschrijden. De Nyquist rate is 2X de hoogste signaal frequentie.

Een typisch DSP systeem bevat volgende ingrediënten :

- Een analoge lowpass-filter of antialiasing-filter voor het beperken van de hoogste signaalfrequentie om aldus aliasing tegen te gaan.
- Een sampler welke op discrete tijdsintervallen van het analoge signaal een staal neemt en werkzaam is boven de Nyquist sampling rate.
- Een quantizer welke de gesampled signaal waarden quantiseert of met een eindige hoeveelheid discrete niveaus voorstelt (vb. 16-bit)

- Een encoder welke de gequantiseerde waarden converteert naar een binaire bitstream of word met een lengte bepaald door het aantal quantisatie-niveaus.
- Het verwerken van het geëncodeerde digitale signaal zelf...
- Een decoder welke de verwerkte bitstream converteert naar een gequantiseerd discreet tijdsignaal.
- Een reconstructiefilter welke de trapswijze benadering reconstrueert van het discreet tijdsignaal.
- Een lowpass analoog anti-imaging filter welke de centrale periode extraheert uit een periodisch spectrum en er tevens de ongewenste image componenten uit verwijdert. Dit resulteert in een vloeiend gereconstrueerd signaal.

Een en ander is samengevat in de volgende figuur.

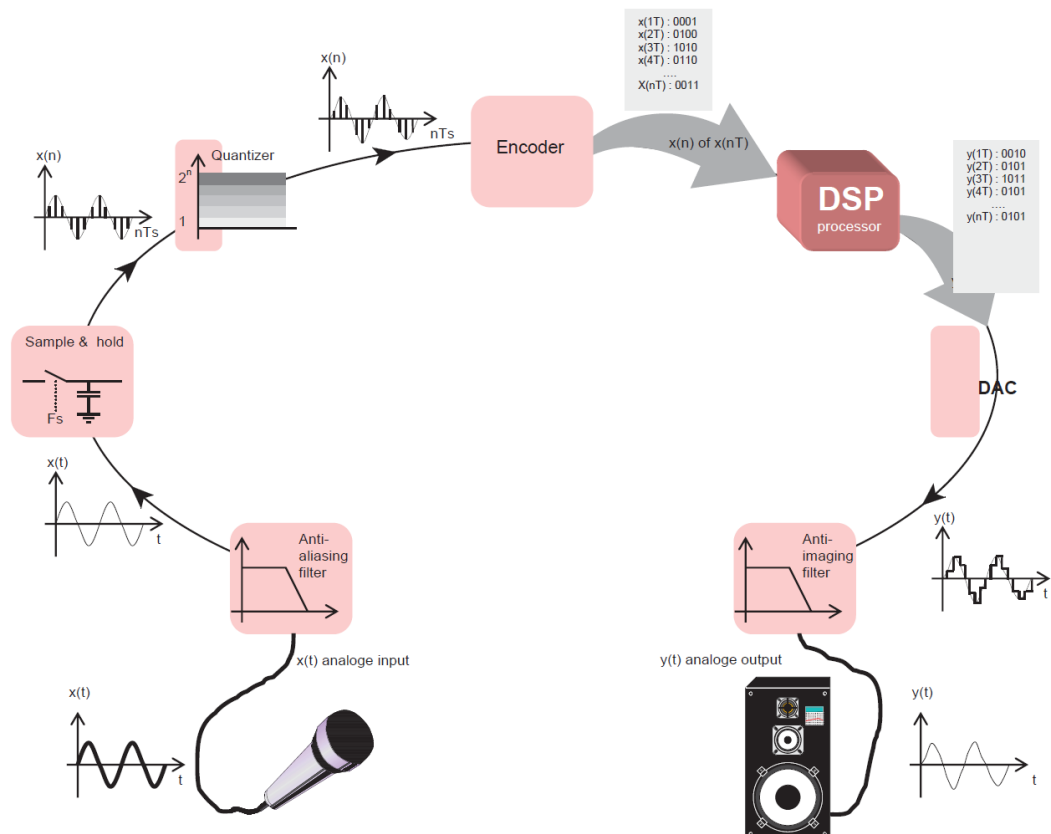


Fig. 1 Typisch DSP systeem

Het belang van het DSP-gebeuren wordt door tal van halfgeleiderfabrikanten onderstreept door het steeds maar opnieuw introduceren van krachtige en specifieke DSP devices. Dit domein is echter zeer omvangrijk en in grote mate op wiskundige technieken gebaseerd. Het is de bedoeling deze wiskundig gebaseerde DSP-concepten te visualiseren en aldus een zekere feeling te verwerven in de parameters van deze algoritmes.

DOEL

- Het leren omgaan met de **systeemontwerpcyclus**.
- Het leren opstellen van een **systeemconcept**.
- Het leren werken met de **Matlabomgeving** en de **signal processing toolbox** in het bijzonder.
- Inzicht verwerven in diverse basis-**DSP-technieken** :
Sampling, quantisatie, aliasing, convolutie, windowing, interpolatie, CFT (Continuous Fourier Transformation), DFT (Discrete Fourier Transformation) en FFT (Fast Fourier Transformation).

A.1 Sampling en quantisatie

$$a = 5$$

$$b = 10$$

$$p = 1000$$

$$q = 1500$$

$$r = 10000$$

$$n = 32$$

We gaan de volgende golfvorm $x(t) = a \sin(2\pi p t) + b \sin(2\pi q t)$ samplen met een sampling frequentie $f_s = r$ Hz.

Welke frequenties bevat deze golfvorm?

$$\rightarrow p = 1000 \text{ Hz}$$

$$\rightarrow q = 1500 \text{ Hz}$$

Bepaal de bijhorende digitale frequenties.

$$\rightarrow \frac{f}{f_s} = \frac{1000 \text{ Hz}}{10000 \text{ Hz}} = 0,10$$

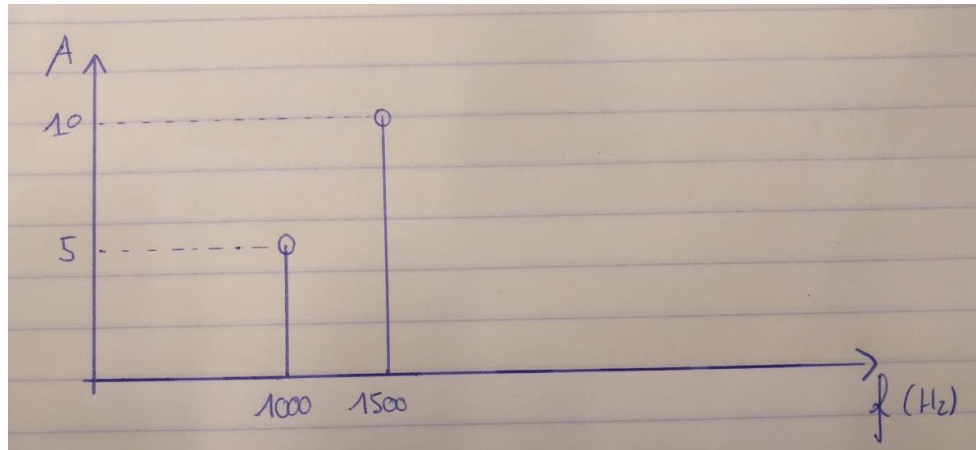
$$\rightarrow \frac{f}{f_s} = \frac{1500 \text{ Hz}}{10000 \text{ Hz}} = 0,15$$



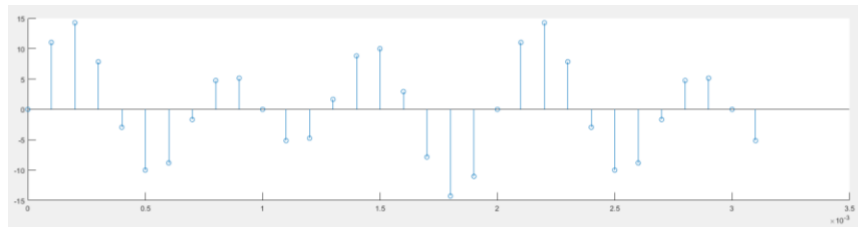
Teken het spectrum. (geen fft)

Welke is hierbij een belangrijke referentiefrequentie?

-> 2x de maximale frequentie = 3000Hz

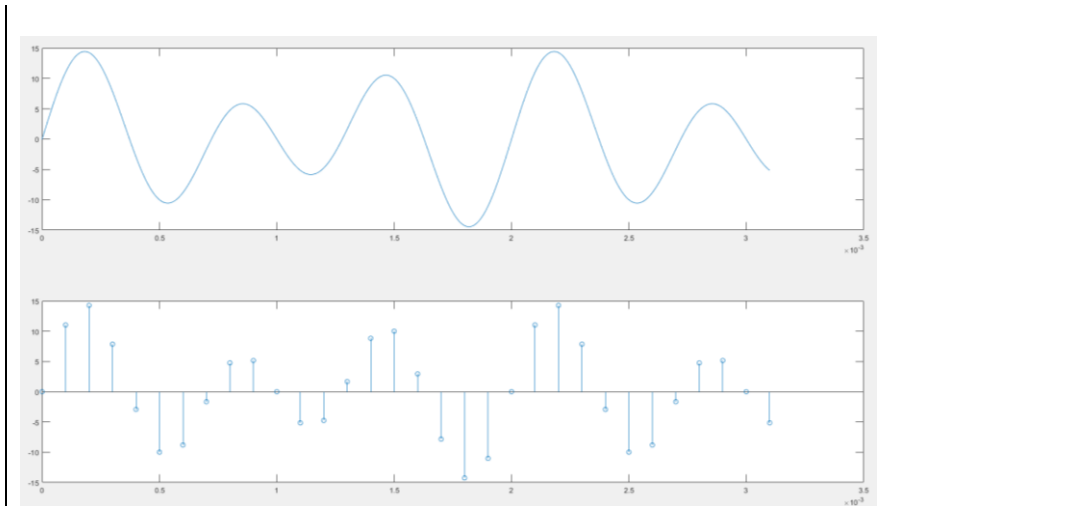


Schrijf een matlab (*.m) file. Maak een grafische voorstelling van deze gesamplede golfvorm in het discrete tijdsdomein (*stem*) door slechts n samples te gebruiken. We bestuderen $x(t)$ dus tot $T=(n-1) \cdot T_s$, waarbij $T_s=1/f_s$.




Combineer deze gesamplede golfvorm met een grafische voorstelling van de golfvorm in het continue tijdsdomein(2 *subplots*). Benader deze continue golfvorm door hiervoor 10x meer samples per periode te gebruiken en de samples te interconnecteren.





Verander enkel de frequentie f_s zodanig dat volgende situaties worden uitgetest : $F = f/f_s = 0 \ 1/10 \ 1/4 \ 1/2 \ 3/4 \ 9/10 \ 1 \ 11/10 \ 5/4 \ 3/2 \ 7/4 \ 19/10 \ 2$.

Welke digitale frequenties F hebben met elkaar verband? 

Visualiseer enkele proefnemingen in het tijdsdomein. Geef bij elke grafiek f_s aan. 

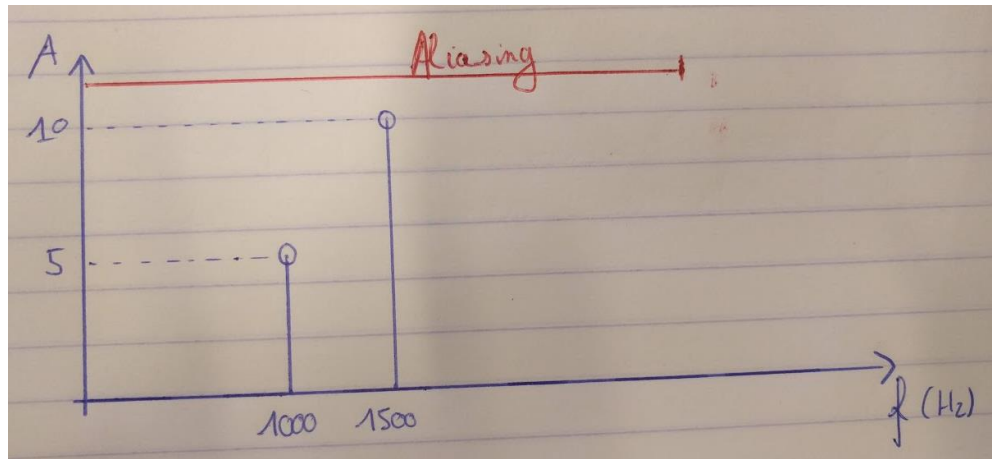
Opgave klopt niet en dient niet gemaakt te worden

In welk interval van samplefrequenties / aantal samples per periode treden problemen op? Teken dit interval in het frequentiedomein.

-> Minimum 2x de maximale frequentie. (Nyquist)

Hoe noemt men het verschijnsel dat optreedt? 

-> Aliasing

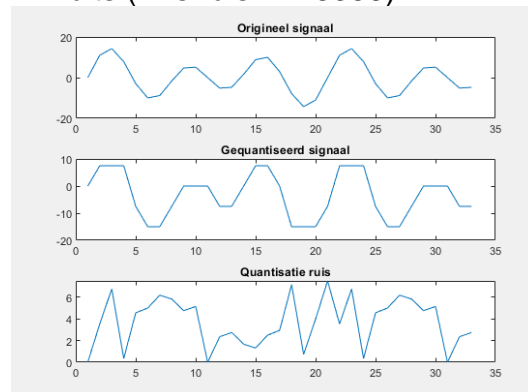


Visualiseer (telkens in 3 *subplots*) de invloed van quantisatie (*round, fix, floor, ceil...*).

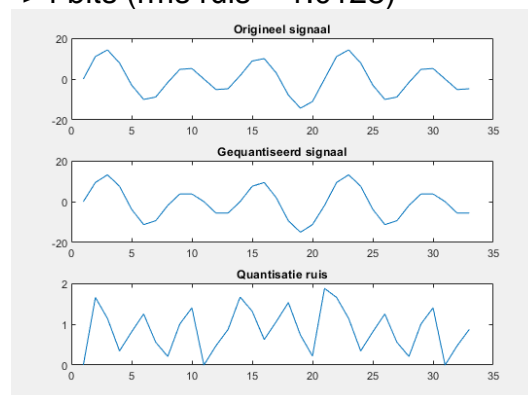
Plot de quantisatiefout (= gequantiseerd signaal - correct signaal) voor een verschillend aantal quantisatieniveaus (afhankelijk van het aantal bits in de DAC). Bepaal telkens de quantisatieruis (rms waarde van de quantisatieruis).

Plot het verband tussen het aantal bits in de DAC en de quantisatieruis (rms).

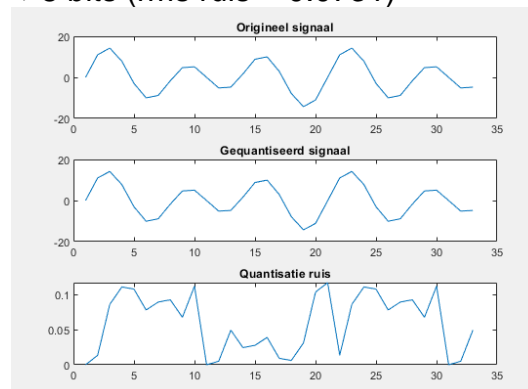
->2 bits (rms ruis = 4.3359)



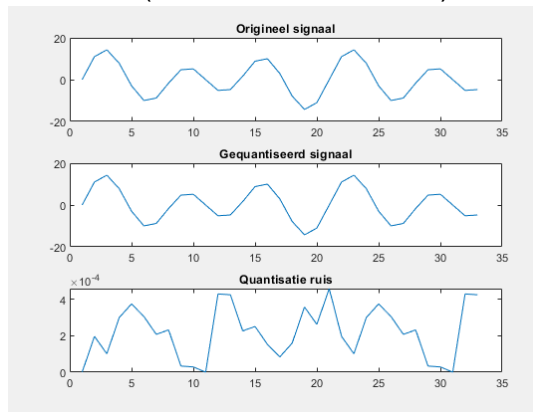
->4 bits (rms ruis = 1.0128)



->8 bits (rms ruis = 0.0731)

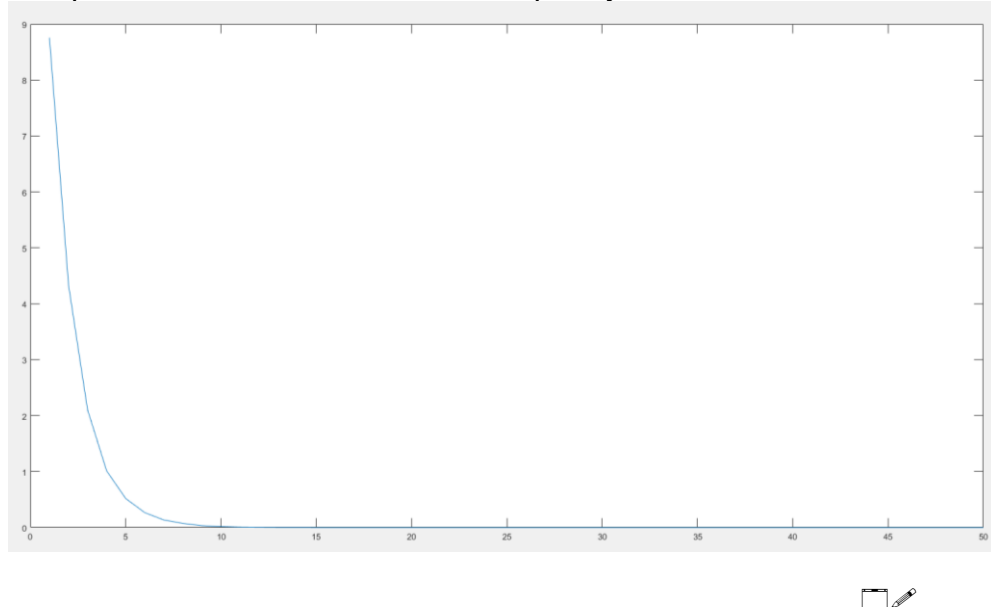


-> 16 bits (rms ruis=2.6063e-04)



Wat is het wiskundige verband?

-> Op de x-as staan het aantal bits, op de y-as de rms waarde van de ruis

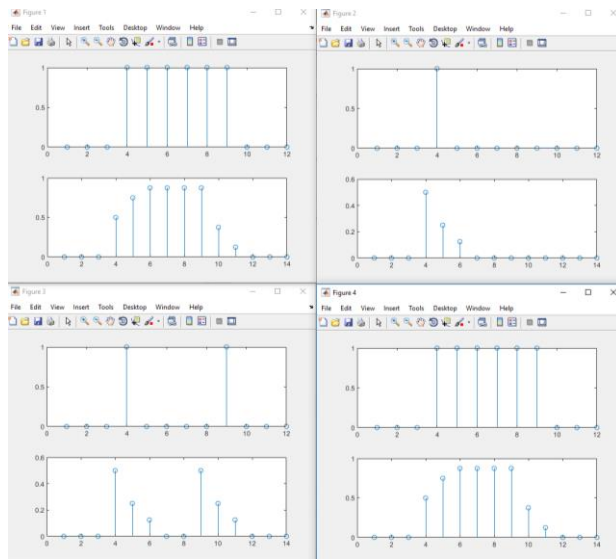


A.2 Convolutie

Schrijf een matlab (*.m) file voor het simuleren van de functionaliteit van **convolutie**. Gebruik hierbij enerzijds een gesamplede functies (0 0 0 1 1 1 1 1), (0 0 0 1 0 0 0 0 0), (0 0 0 1 0 0 0 0 1 0 0 0), (0 0 0 1 1 1 1 1 1 0 0 0) en anderzijds een set eenvoudige coëfficiënten (vb. 0.5 0.25 0.125) en convolveer (*conv*) beiden.

Maak een grafische voorstelling (*stem*, *subplot*) van de gesamplede stapfunctie, de set coëfficiënten en de convolutiesom in het discrete tijdsdomein.

mask = [0.5 0.25 0.125];

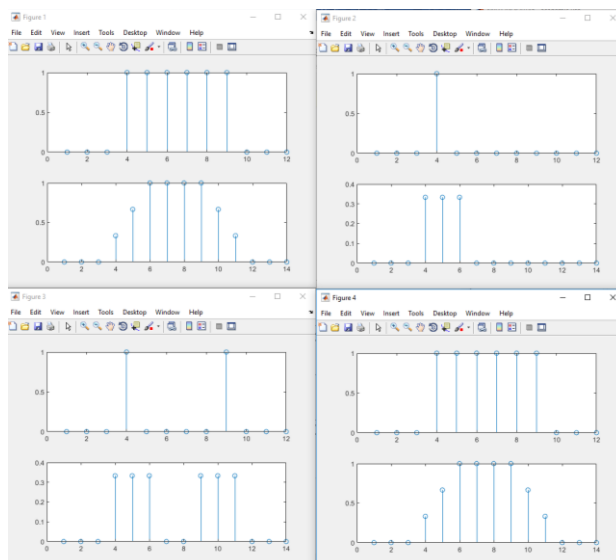


Wat gebeurt er als de coëfficiënten worden vervangen door (1/3 1/3 1/3)?

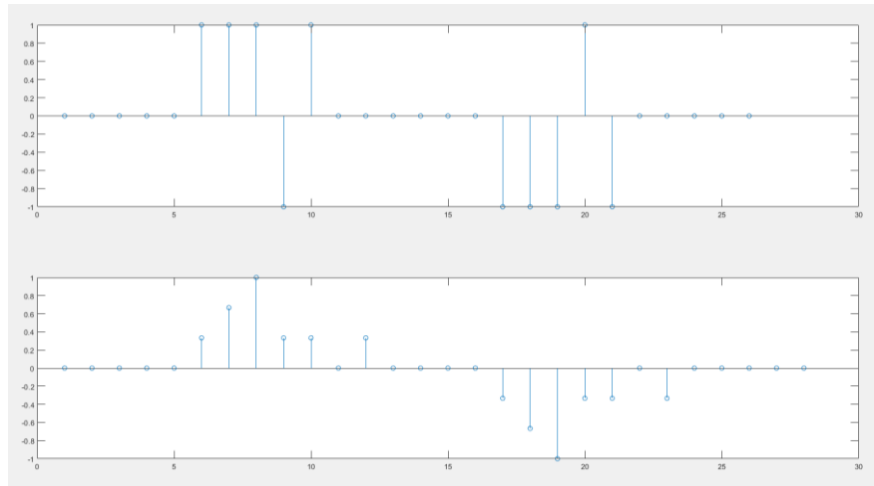
-> De convolutie wordt symetrisch



mask = [1/3 1/3 1/3];

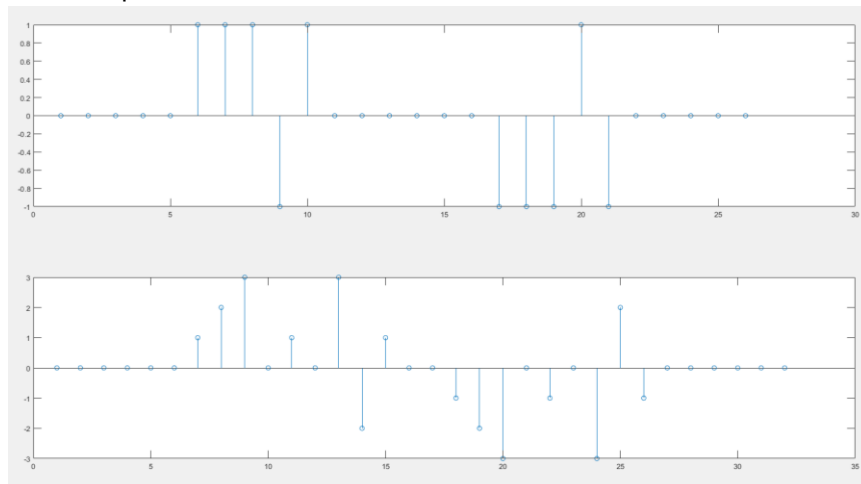


Gebruik volgende ingangsfunctie: (0 0 0 0 0 1 1 1 -1 1 0 0 0 0 0 -1 -1 -1 1 -1 0 0 0 0 0).



Wat gebeurt er als de coëfficiënten (0 1 1 1 -1 1 0) een kopie zijn van de gesamplede ingangsfunctie?

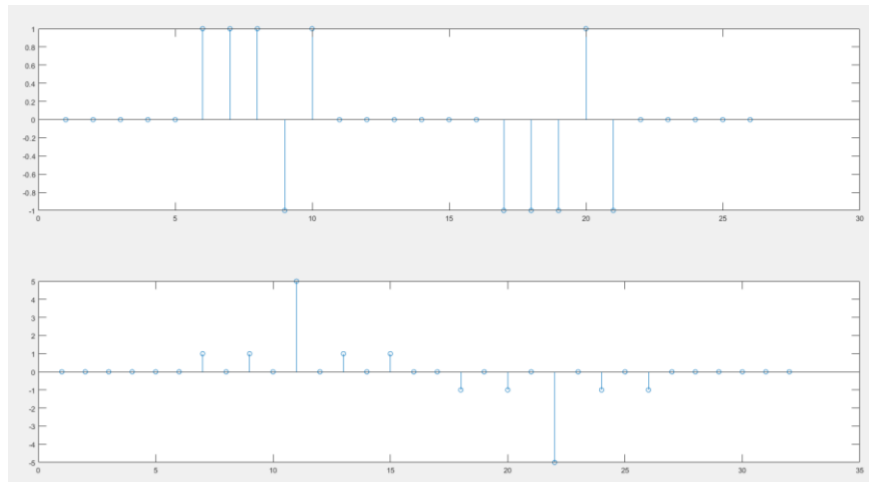
-> Niets speciaal



Wat gebeurt er als de coëfficiënten (0 1 -1 1 1 1 0) een gespiegelde kopie zijn van de gesamplede ingangsfunctie?



-> Pieken waar geïnverteerd signaal overeenkomt.



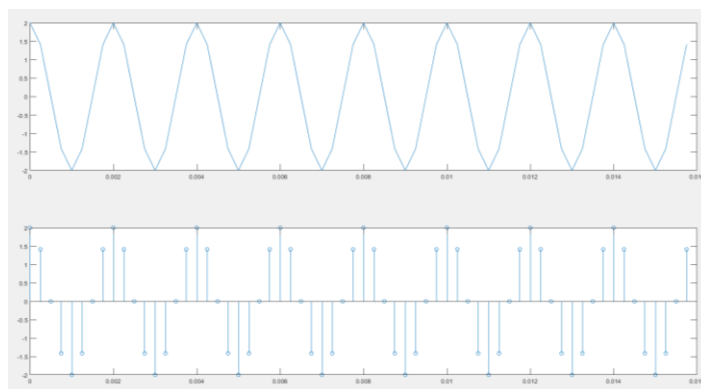
A.3 Fast Fourier Transform


$a=2$


$r=4000$

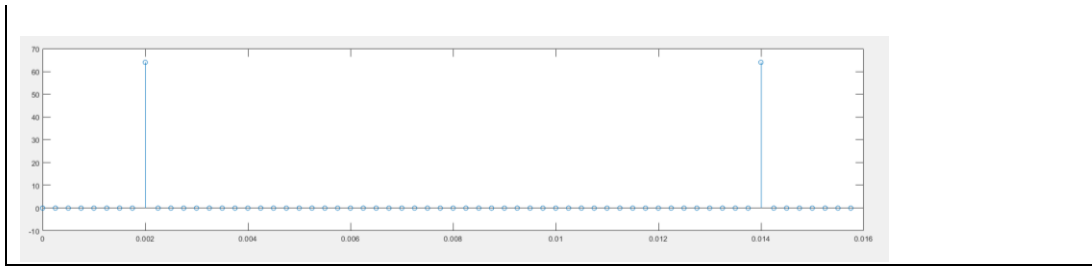
$p=500$

Schrijf een matlab (*.m) file voor het onderzoeken van de invloed van sampling op het frequentiespectrum van een golfvorm in het continue tijdsdomein. Gebruik volgende eenvoudige golfvorm $x(t) = a \cos(2\pi p t)$ en sample deze met een $f_s = r$ Hz. Beeld een grafiek af in het discrete tijdsdomein van het originele signaal. Maak hierbij gebruik van een dataset van 64 samples.

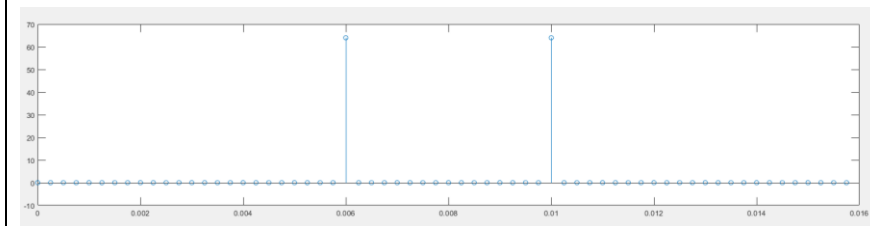


Voer een Fast Fourier Transformatie (*fft*, *abs*) uit op de gesampled dataset en beeld de magnitude af in functie van de frequentie. 

Wat is het verschil met het spectrum van eenzelfde niet-gesampled signaal? 

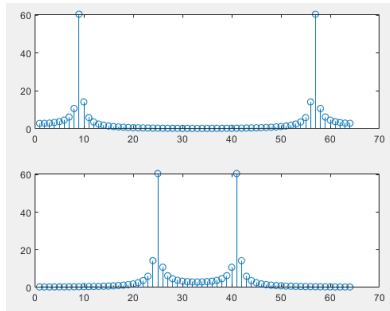


Pas de (*.m) file aan voor het maken van een spectraalgrafiek die meer gelijkenis vertoont met deze van continue Fouriertransformatie (*fftshift*).

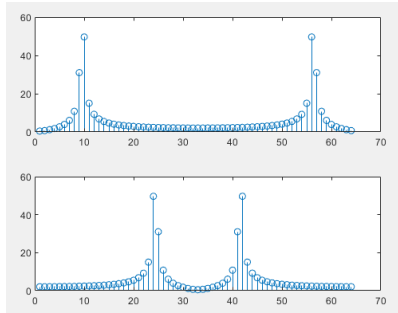


Wijzig de golfvormfrequentie : $f = a + 12.5$, $f = a + 37.5$, $f = a + 62.5$ Hz.

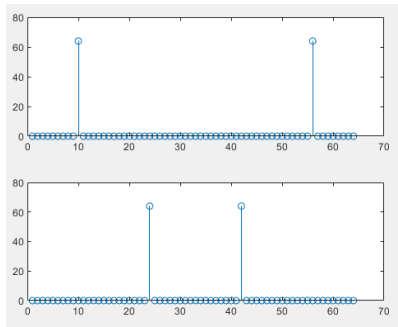
+12.5



+37.5



+62.5



Wat gebeurt er met het frequentiespectrum?

Hoe noemt men dit verschijnsel?



A.4 Windowing

$r = 8000$

$p = 1000$

$b = 62,5$

$n = 64$

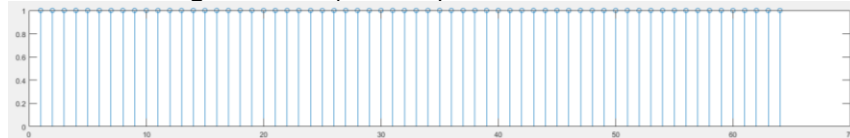
$a = 625$

$c = 250$

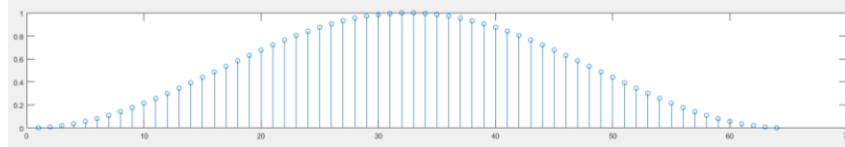
$d = 125$

Schrijf een (*.m) file welke toelaat de invloed te onderzoeken van windowing. Vergelijk volgende veelgebruikte windows op het vlak van het discrete tijdsdomein alsook op het vlak van het discrete frequentiedomein:

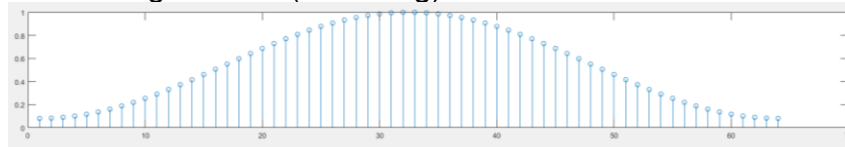
- Rechthoekig window (*boxcar*)



- von Hann window (*hanning*)



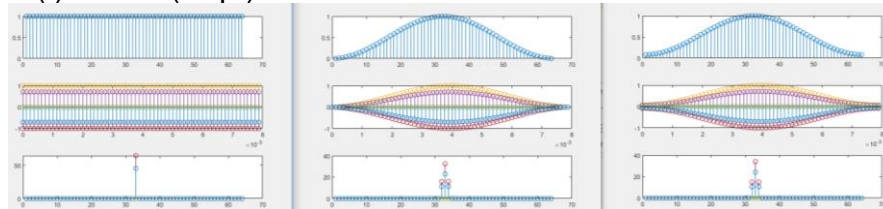
- Hamming window (*hamming*)

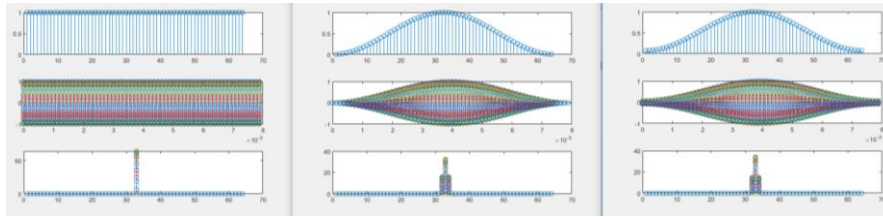


Plot (*subplot,stem*) het tijds- en frequentieverloop van deze drie window-types voor een lengte van n samples.

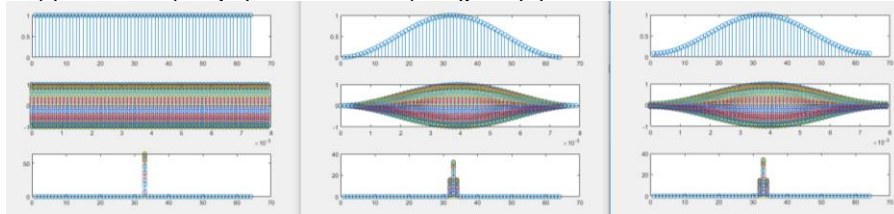
Gegeven een golfvorm $x(t)$. Sample de golfvorm met $f_s = r\text{Hz}$ en gebruik een window van n samples. Pas op volgende situaties de verschillende windows toe:

$-x(t) = 1 \sin(2\pi p t)$

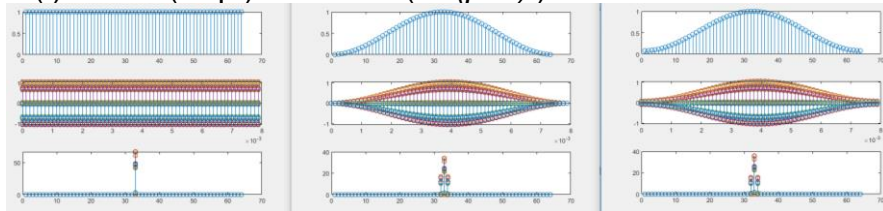




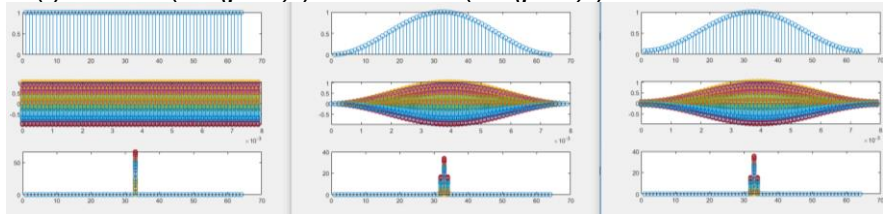
$$-x(t) = 1 \sin(2\pi pt) + 0.05 \sin(2\pi(p+b)t)$$



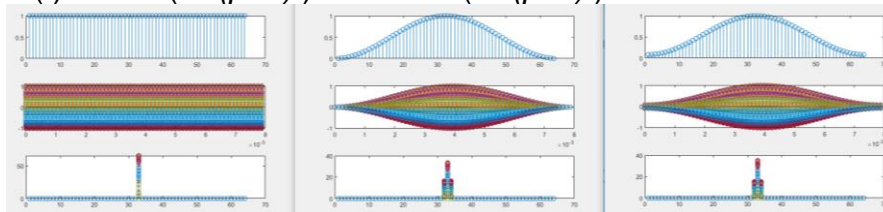
$$-x(t) = 1 \sin(2\pi pt) + 0.05 \sin(2\pi(p+c)t)$$



$$-x(t) = 1 \sin(2\pi(p+d)t) + 0.05 \sin(2\pi(p+b)t)$$



$$-x(t) = 1 \sin(2\pi(p+d)t) + 0.05 \sin(2\pi(p+c)t)$$




Leg uit wat de functie van windowing is?



Welk type window geeft de grootste leakage? Duid voor dit venster in één van de grafieken (zowel tijd als frequentie) van voorgaande signalen aan waar de leakage zich voordoet.



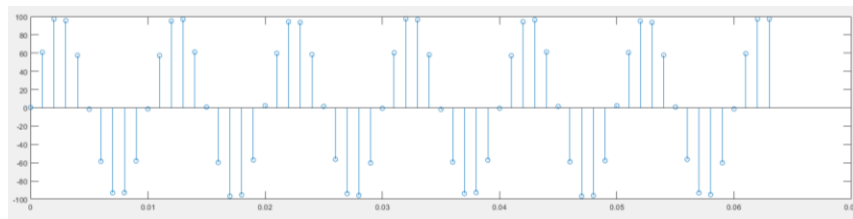
Welk type window geeft de grootste resolutie? Duid voor dit venster in één van de grafieken (zowel tijd als frequentie) van voorgaande signalen aan waar de resolutie zichtbaar is. 


A.5 Ruis – windowing – interpolatie

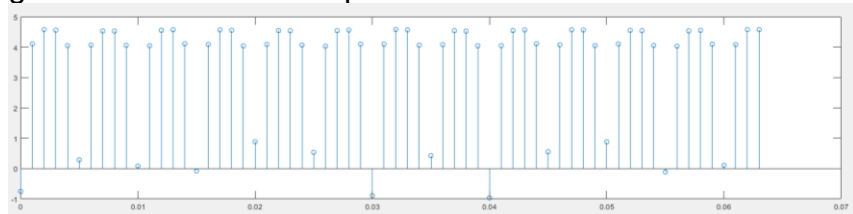
Schrijf een (*.m) file welke toelaat de invloed van Gaussische ruis te onderzoeken (*randn*)

op de spectraalinformatie van een golfvorm. Sample hiervoor volgende golfvorm:

$$x(t) = 100 \sin(2\pi 100t) + 2 \sin(2\pi 164.25t) + 1 * \text{noise}$$

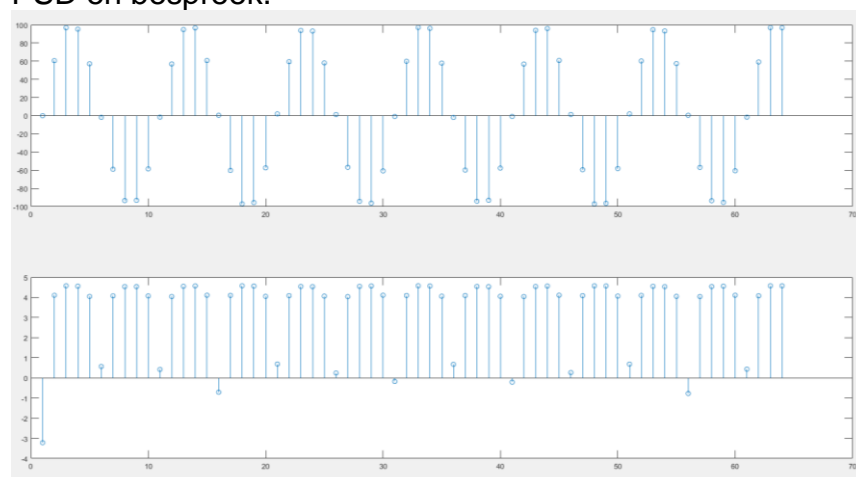


waarbij $f_s = 1000$ Hz en een boxcarwindow van 64 samples wordt gebruikt. Beeld in het discrete tijdsdomein de 64 samples van de golfvormdataset af, alsook de PSD (Power Spectral Density, in dB) (\log_{10}) van de golfvorm. Wat is nu het probleem? 



Voeg achteraan aan de gesamplede data set nullen toe zodat de data set wordt uitgebreid tot 512 (*zeros, length*). Bekijk het effect hiervan op de

PSD en bespreek.



B. INTRODUCTIE VAN DIGITALE NETWERKEN EN DE Z-TRANSFORMATIE. ANALYSE VAN FILTERS.

INLEIDING

De meeste analoge signalen worden tegenwoordig gedigitaliseerd om ze daarna digitaal met behulp van een computer, microprocessor of speciale DSP-chip te bewerken. Dit stelt ons in staat om op zeer flexibele wijze te kunnen werken. In de vorige labotekst hebben we reeds gebruik gemaakt van FFT voor spectraal analyse en voor convolutie. Net zoals in analoge signaalverwerking is het noodzakelijk bepaalde frequenties te kunnen onderdrukken relatief t.o.v. andere door gebruik te maken van filters. Digitale filters zijn dan ook binnen signaalverwerking van cruciaal belang.

Analoge netwerken worden meestal beschreven met een transferfunctie in het s-domein. Ook voor digitale netwerken kan men dit gebruiken, maar dit leidt tot een oneindig aantal polen en zeros (periodisch), wat dit s-domein onbruikbaar maakt bij digitale signaal verwerking. Dit is eveneens waar voor de Laplace-transformatie van discrete tijdsignalen. Alles wordt eenvoudiger door over te gaan naar het z-domein en de variabele s te vervangen door een variabele $z=e^{sTs}$ (e^{-sTs} is de transferfunctie in het s-domein van een tijdsvertraging Ts). Dit leidt ons naar de z-transformatie. Een hulpmiddel bij de interpretatie van resultaten is zeker het verband tussen het s-domein en het z-domein zoals in figuur 1 en 2 wordt getoond.

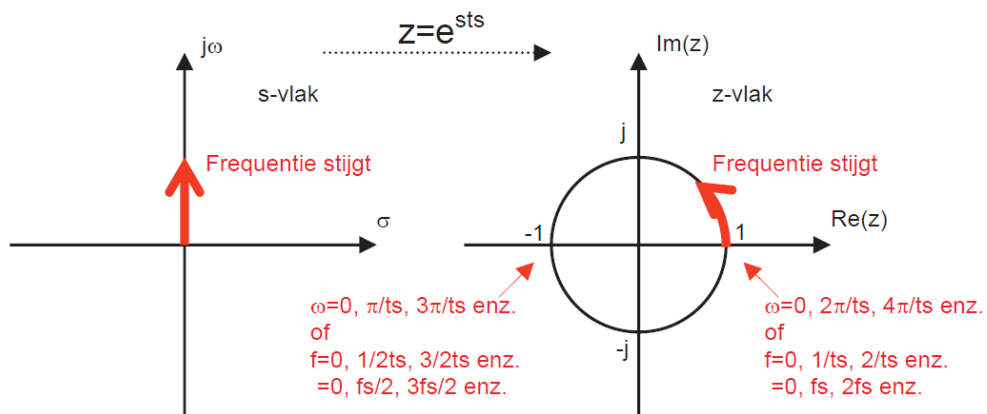


Fig. 1. : De imaginaire as in het s-vlak wordt op de eenheidscirkel in het z-vlak afgebeeld

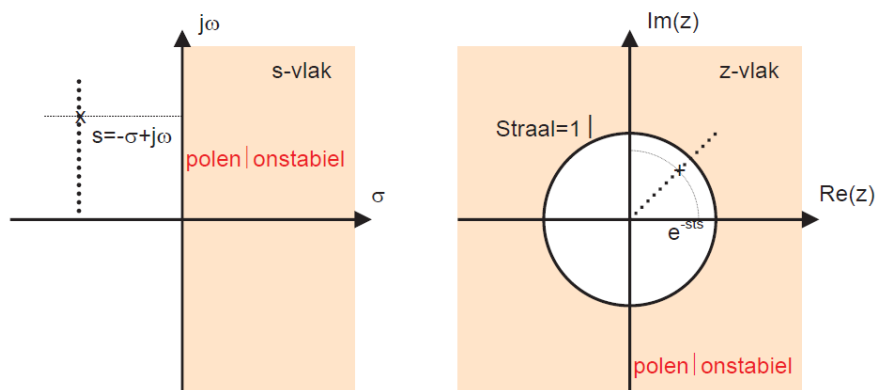


Fig. 2. : Polen in het rechter halfvlak van het s-vlak komen overeen met polen gelegen buiten de eenheidscirkel en wijzen op een onstabiel systeem.

De z-operator kan simpel omschreven worden als een tijdsverschuiving over een sample interval (T_s). Dit maakt het verband met digitale filternetwerken duidelijker zoals in figuur 3.

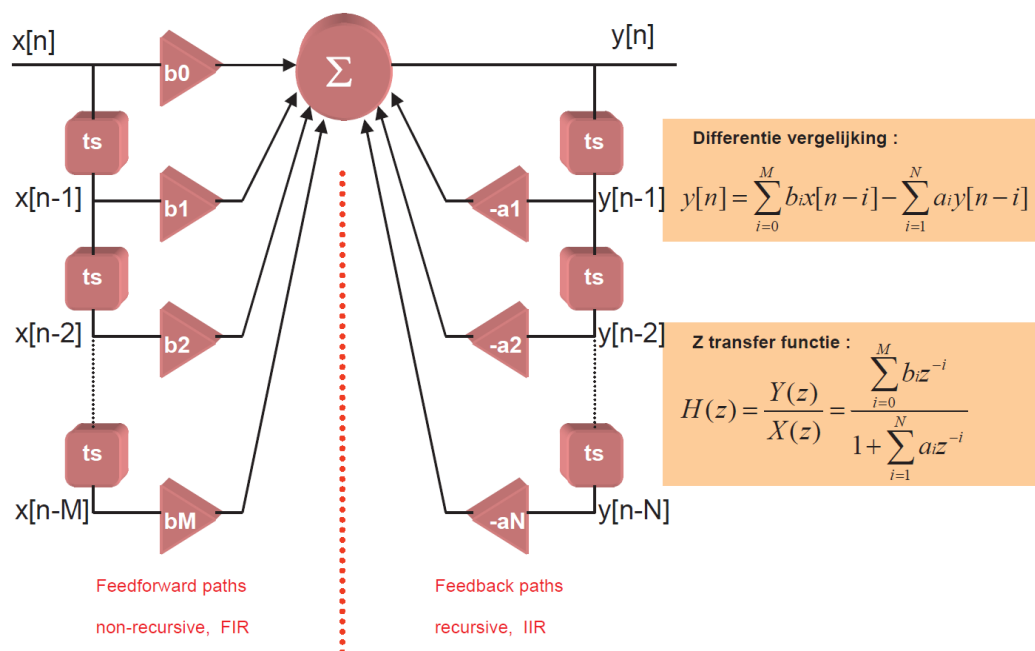


Fig. 3. : Digitale filter topologie met feedforward en feedback paden.

DOEL

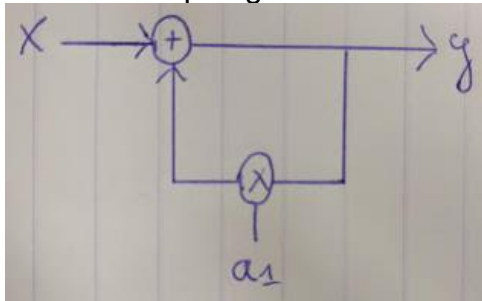
- Het leren omgaan met de **systemontwerpcyclus**.
- Het leren opstellen van een **systemconcept**.
- Het leren werken met de **Matlab** omgeving en de **signal processing toolbox** in het bijzonder.

- Inzicht verwerven in diverse basis **DSP-technieken** :
Differentie vergelijking van een digitaal netwerk, Z-transformatie, transfer functie $H(z)$ in het z-domein, recursieve (IIR Infinite Impulse Response) - en niet recursieve (FIR Finite Impulse Response) filters, interpretatie van pole-zero plots in het z-domein.

B.1 1^e orde IIR

Gebruik een IIR-netwerk met volgende differentievergelijking van 1^eorde : $y(n) = x(n) + a_1 y(n-1)$. De coëfficiënt a_1 neemt volgende waarden aan : 0.1 0.5 0.9 1 1.1 . Als input x nemen we een stap met amplitude 1.

-Teken de topologie.



-Waaraan kan je zien dat dit een IIR-netwerk is?

Aan de terugkoppeling (n-1)

-Bereken manueel voor de eerste 4 samples $y(n)$.

$$y(0) = 1$$

$$y(1) = 1.1$$

$$y(2) = 1.11$$

$$y(3) = 1.111$$

-Bereken de limietwaarde (DC versterking).

1 + a1 (Rechtstreekse output + de terugkoppeling)

$$\rightarrow 1 + 0,1 = 1,1$$

- Geef de wiskundige beschrijving voor de stapresponsie van een 1^{ste} orde analoog LDL en voor dit 1^{ste} orde digitaal LDL. Haal uit deze beschrijving een verband tussen de tijdsconstante van het analoge filter en de coëfficiënt a_1 .

$$\text{Analoog: } w(t) = a(1 - e^{-\frac{t}{\tau}})$$

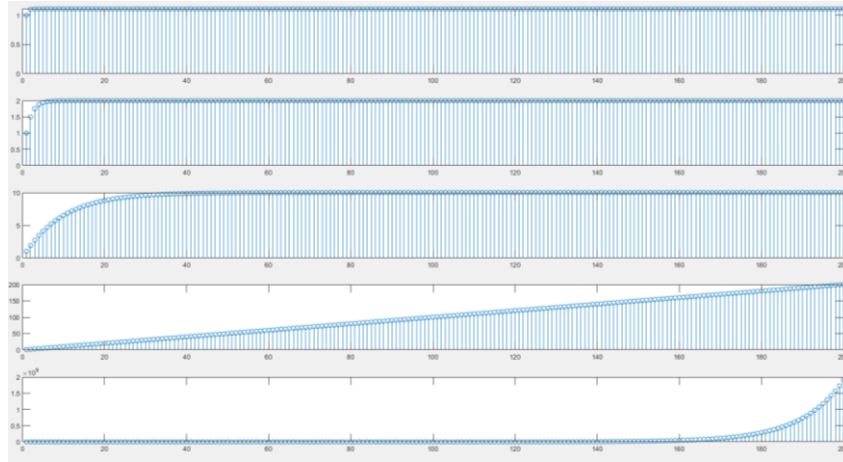
Digitaal



Schrijf een matlab (*.m) file en sample een dataset $x(t)$ van 200 maal 1 (ones). Maak gebruik van de functie *filter*. Verifieer de hierboven berekende waarden grafisch.



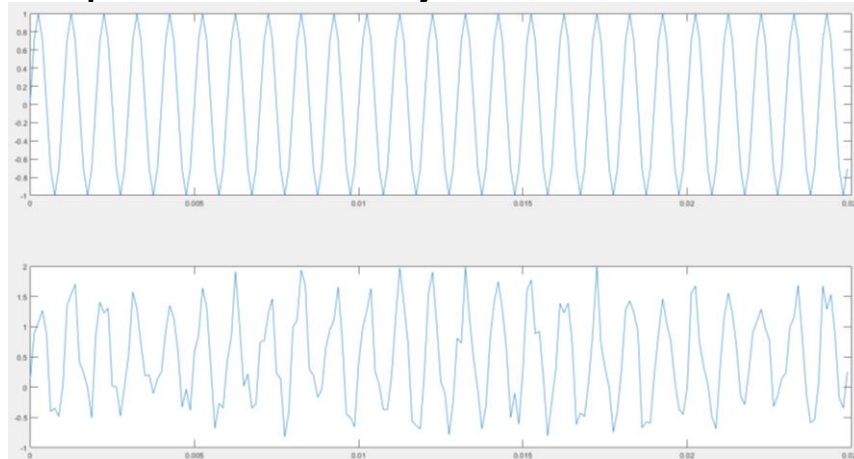
Respectievelijk volgens bovenstaande inputwaarden



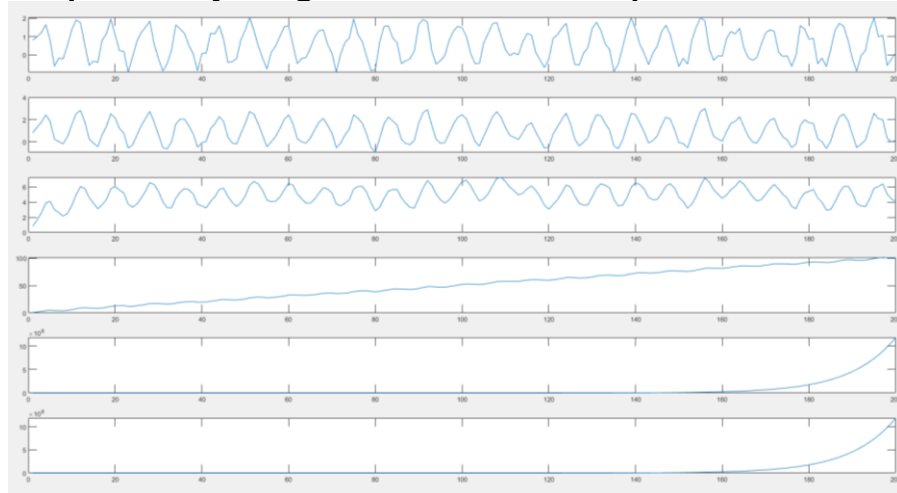
Pas de matlab (*.m) file aan en sample volgende golfvorm $x(t) = \sin(2\pi r t)$ + ruis (eenheidsruis tussen 0 en 1) met een sampling frequentie $f_s = r$ Hz. Gebruik opnieuw het IIR-netwerk met differentievergelijking van 1^eorde : $y(n) = x(n) + a_1 y(n-1)$ om de golfvorm te filteren. Vergelijk in het discrete tijdsdomein $x(n)$ en $y(n)$ met elkaar. Wat is de invloed van de waarde van de coëfficiënt a_1 op de frequentie-inhoud van het signaal? Maak hiervoor de coëfficiënt gelijk aan 0.1 0.5 0.9 1 1.1



De inputsinus en sinus + zijn ruis




Respectievelijk volgens bovenstaande inputwaarden




Analyseer de 1^eorde differentievergelijking van opdracht 1: $y(n) = x(n) + a_1 y(n-1)$ in het z-vlak. Hiervoor gebruiken we de z-transformatie die leidt tot volgende transferfunctie :

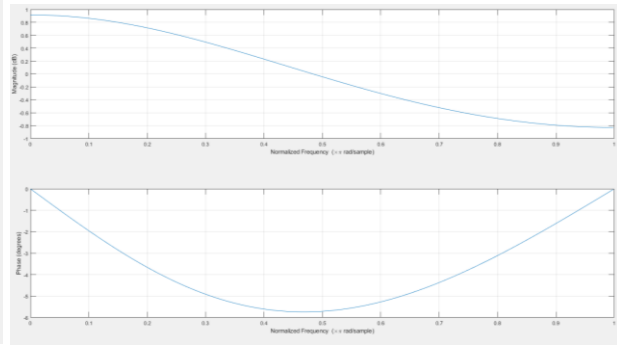
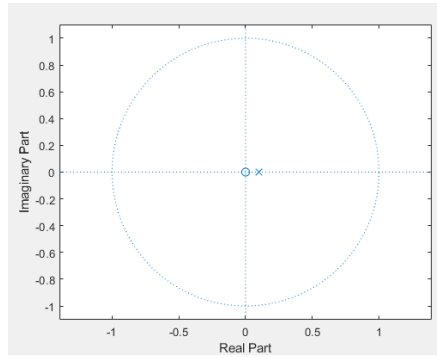
$$H(z) = |H(z)|e^{j\angle H(z)}$$

$$H(z) = \frac{1}{1 - a_1 z^{-1}} = \frac{z}{z - a_1}$$

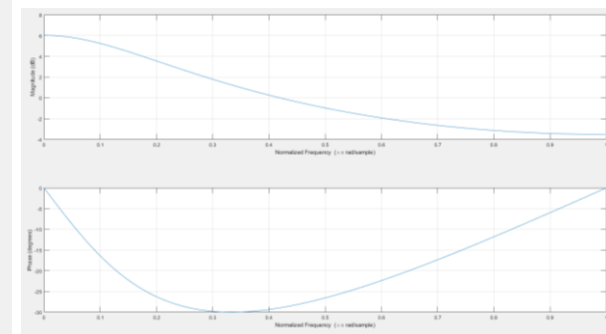
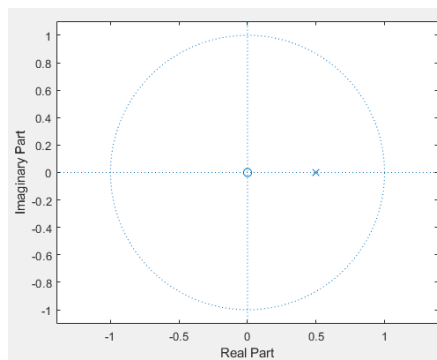
Bereken de frequentieweergave (zowel amplitude als fase) bij $f=0$, $f_s/4$, $f_s/2$. 

f	Z	H	H	$\angle H$
0	1	0,91	0,91	0
$f_s/4$	j	0,19+j0,9	0,92	78
$f_s/2$	-1	2,71	2,71	0

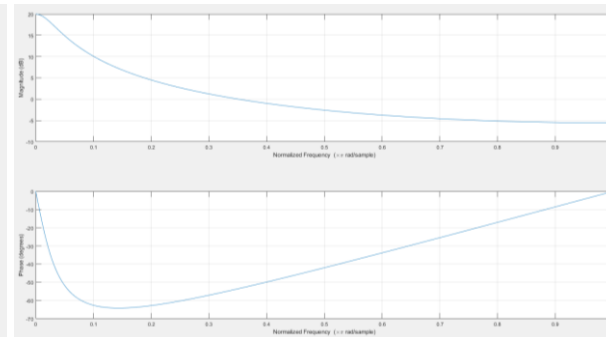
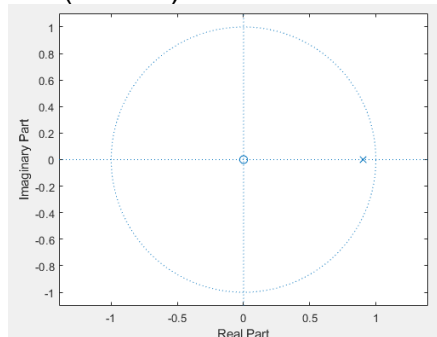
Schrijf een matlab (*.m) file. Beeld de polen en zeros af (*zplane*). Beeld de amplitude- en faseresponsie af in het discrete frequentiedomein (*freqz*). Maak hiervoor de coëfficiënt gelijk aan 0.1 0.5 0.9 1 1.1 en vergelijk de resultaten met deze van opdracht 1 in het discrete tijdsdomein. 



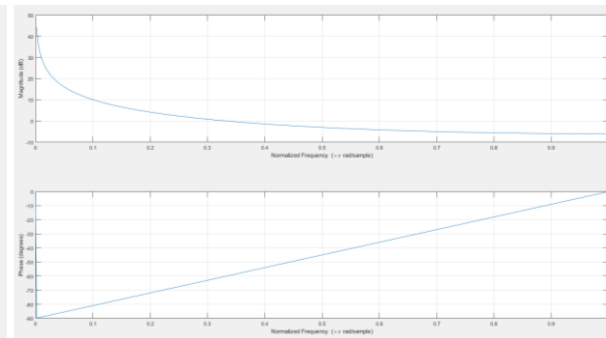
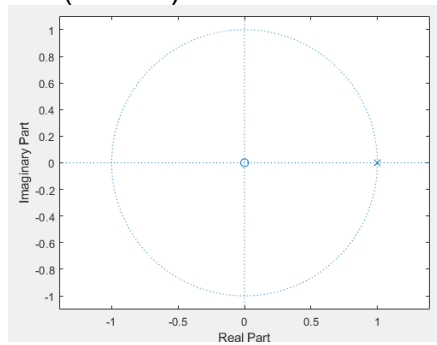
0.5 (Stabil)

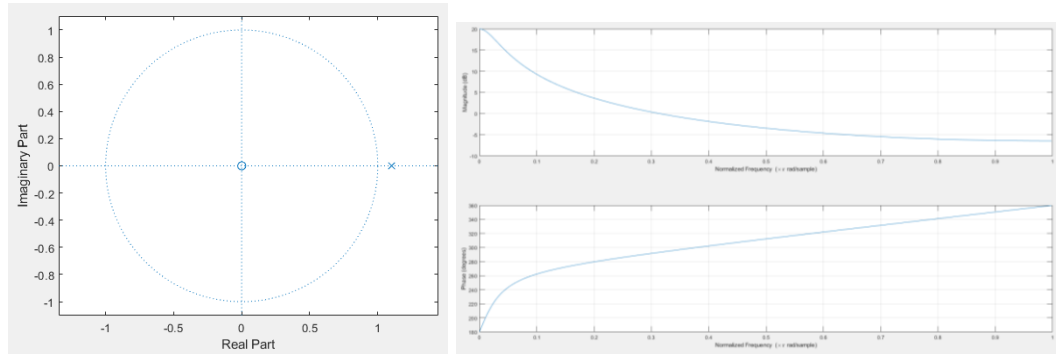



0.9 (Stabil)



1.0 (Stabil)





Wanneer is dit digitale netwerk onstabiel ? 

Wanneer de coëfficiënt groter dan 1 is.

B.2 2^e orde IIR


$$a_1 = -1.45$$

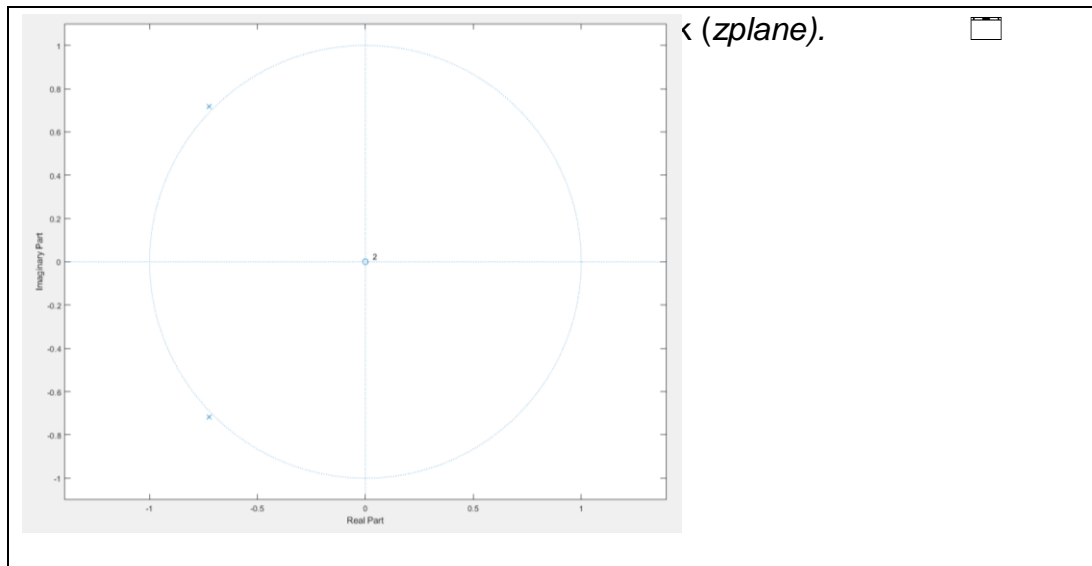
$$a_2 = 1.04$$

Schrijf een matlab (*.m) file om volgende 2^e orde *feedback* differentievergelijking

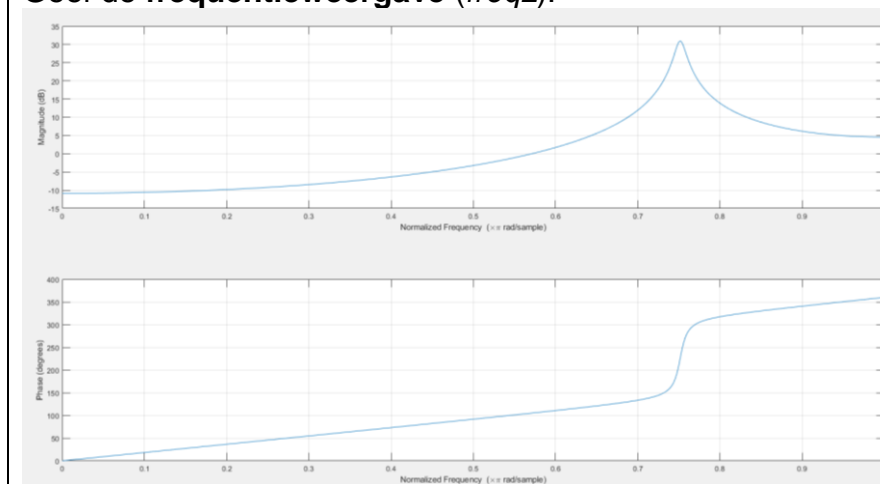
$y(n) = x(n) - a_1 \cdot y(n-1) - a_2 \cdot y(n-2)$ (zie fig. 3) te analyseren op het vlak van stabiliteit en frequentieresponsie. De vergelijking kan worden omgevormd tot de transferfunctie in z :

$$H(z) = \frac{1}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} = \frac{z^2}{z^2 + a_1 \cdot z + a_2}$$

Bepaal de **poolcoördinaten** (r_p , θ_p) van de polen (zie cursus). 



Geef de **frequentieweergave** (*freqz*).



Wat is de invloed van de polen op de frequentieweergave?

- > **Bandbreedte van de fase + helling van de fase**
- > **Resonantiefrequentie**

Bepaal de **resonantiepiek** (zie formularium).

Bereken $H(\text{DC})$, $H(f_s/4)$ en $H(f_s/2)$.

$$H(\text{DC}) \Rightarrow H(Z=1) = -1,45 + 1,04 = -0,41$$

$$H(f_s/2) \Rightarrow H(Z = -1) = -1,45 - 1,04 = -2,49$$

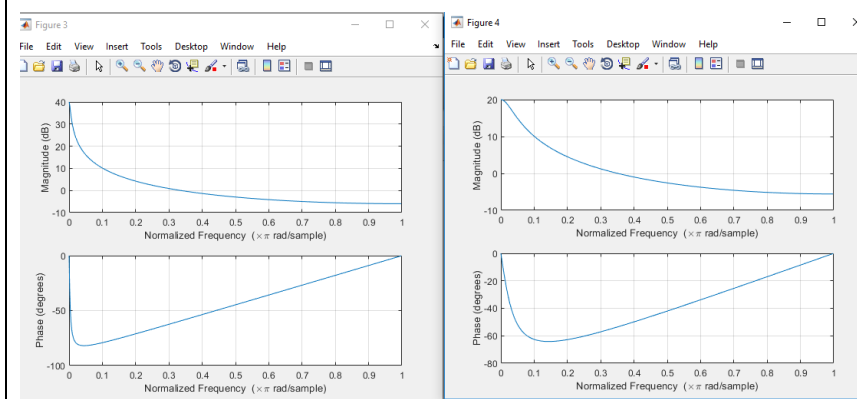
Laat matlab de poolfrequentie (roots) berekenen indien $f_s=8\text{kHz}$.

$$-0.7250 + 0.7172i$$

$$-0.7250 - 0.7172i$$

Welk **soort filter** betreft het hier?
Een hoogdoorlaatfilter

Laat r_p naar 1 evolueren (0,9 0,99). Wat is de invloed van de ligging van de polen op de frequentieweergave? Maak gebruik van de *poly*-functie in matlab om de coëfficiënten te berekenen.



Laat θ_p tussen 0° en 180° (0° 90° 180°) evolueren. Wat is de invloed van de ligging van de polen op de frequentieweergave? Maak ook hier gebruik van de *poly*-functie.

Om een aantal tijdsgedrag-eigenschappen van digitale netwerken te onderzoeken worden zoals bij analoge netwerken eenvoudige ingangssignalen aangelegd. We kunnen een aantal eigenschappen afleiden uit het antwoord (responsie) van het digitale systeem op een stap (stapresponsie) en op een impuls (impulsresponsie). In het discrete frequentiedomein gebruiken we hiervoor in Matlab (*dstep*, *dimpulse*).

Plot de **stap- en impulsresponsie** van het oorspronkelijke IIR-netwerk.

Leg het verband tussen de ligging van de polen (r_p , θ_p) en de golfvorm van de impulsresponsie (raadpleeg formularium).



Bepaal aan de hand van de stapresponsie de **DC-versterkingsfactor**.



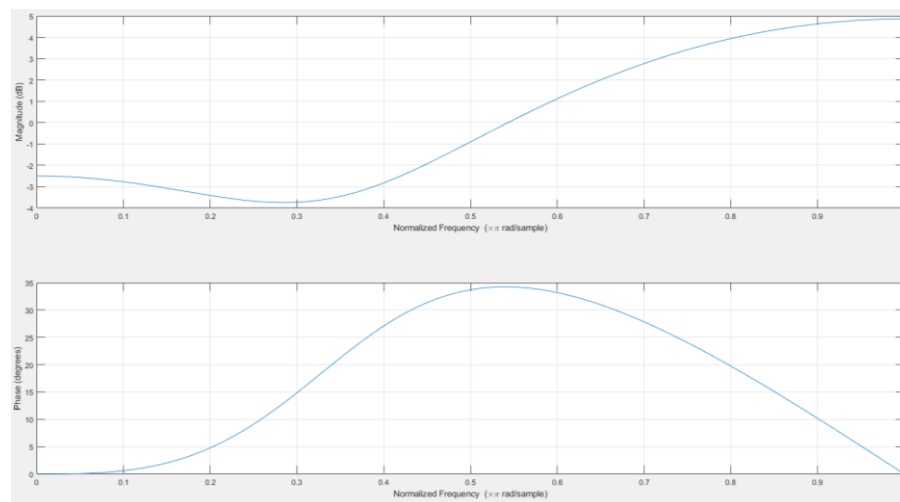
B.3 2^e orde FIR

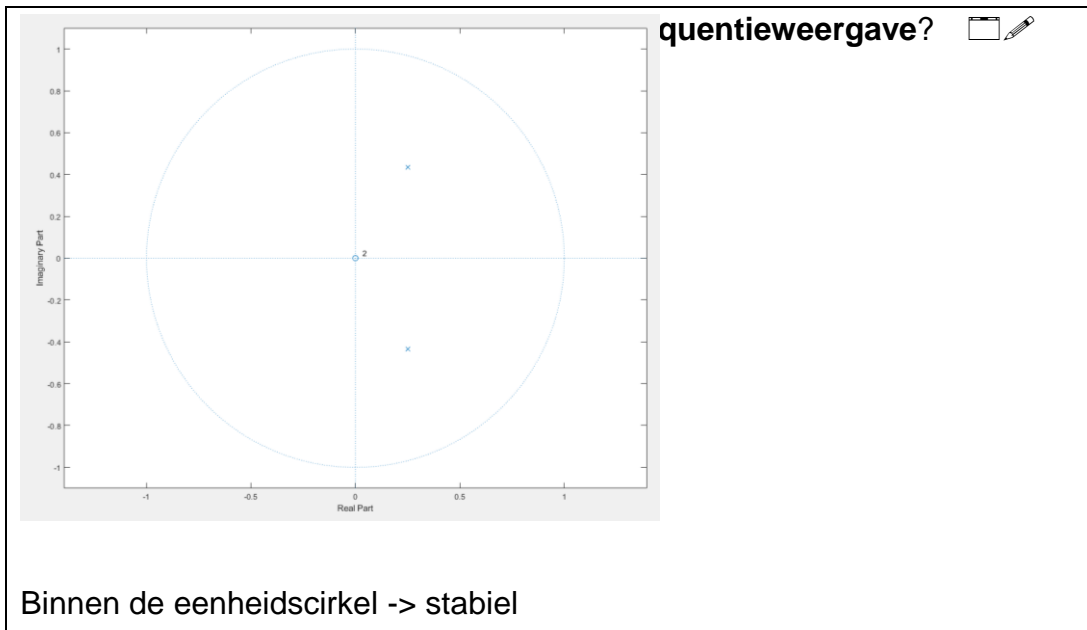
$b_1 = 0,5$

$b_2 = 0,25$


Schrijf een matlab (*.m) file om volgende 2^e orde *feedforward* differentievergelijking $y(n) = x(n) + b_1 \cdot y(n-1) + b_2 \cdot y(n-2)$ (zie fig. 3) te analyseren op het vlak van stabiliteit en frequentieresponsie. De vergelijking kan worden omgevormd tot de transferfunctie in z :

$$H(z) = 1 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} = \frac{z^2 + b_1 \cdot z + b_2}{z^2}$$








Bepaal de **resonantiepiek** (zie formularium). 

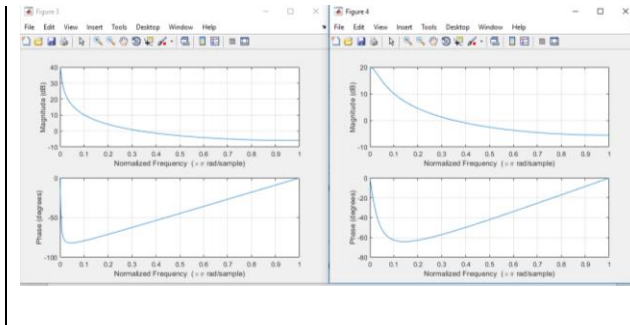
Bereken $H(\text{DC})$, $H(f_s/4)$ en $H(f_s/2)$. 


f	Z	H	H	$\angle H$
0	1	0,91	0,91	0
$f_s/4$	j	$0,19 + j0,9$	0,92	78
$f_s/2$	-1	2,71	2,71	0

Laat matlab de de zerofrequentie (*roots*) berekenen indien $f_s = 8\text{kHz}$ 
 $0.2500 + 0.4330i$
 $0.2500 - 0.4330i$

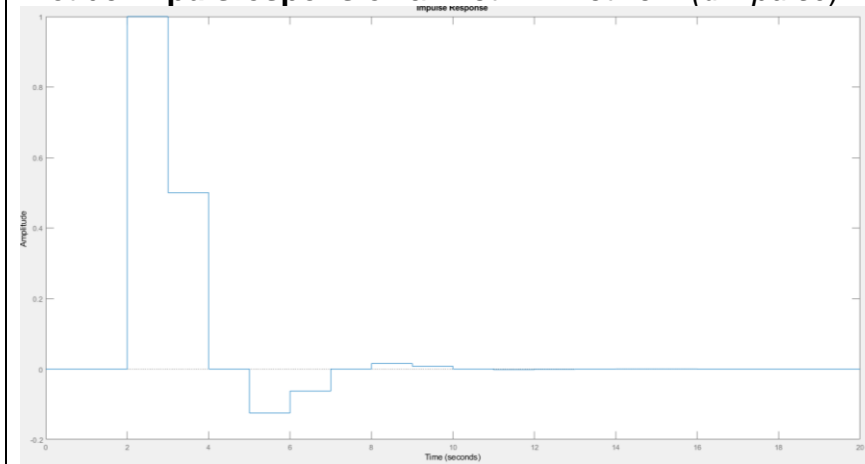
Welk **soort filter** betreft het hier? 
 Een hoogdoorlaatfilter

Laat r naar 1 evolueren (0,9 0,99). Wat is de invloed van de ligging van de nulpunten op de frequentieweergave?  



Laat θ tussen 0° en 180° (0° 90° 180°) evolueren. Wat is de invloed van de ligging van de nulpunten op de frequentieweergave? 

Plot de **impulsresponsie** van het FIR-netwerk (*dimpulse*). 



Vergelijk de impulsresponsie van het IIR- en het FIR-netwerk (*dimpulse*). Wat stel je vast? Verklaar? 

De iir filter deint langer uit en start onmiddellijk. De fir filter start pas na enige vertraging maar deint wel bijna onmiddellijk uit.

C. SYNTHESE VAN DIGITALE FILTERS.

INLEIDING

De bedoeling van een signaal te filteren is gebaseerd op het verbeteren van de gewenste eigenschappen van dat signaal. Digitaal filteren kan op zeer flexibele wijze gebeuren. Het is gemakkelijk verenigbaar met niet lineaire operaties zoals clipping of het verwijderen van incompatibele data samples in de omgeving van de gewenste data samples. Hoe dan ook, is een van de belangrijkste operaties deze van het lineair filteren van een signaal. Hierbij wordt een gekozen frequentieband versterkt of verzwakt relatief t.o.v. de andere frequenties. Zoals bij analoge filters is het ook hier gangbaar een lage rimpel in de doorlaatband, een hoge attenuatie in de sperband en een hoge roll-off rate na te streven. In tegenstelling tot analoge filters is het bij digitale filters echter gemakkelijk een berekende transferfunctie te vertalen in een praktisch filterontwerp.

Een gangbaar alternatief bij de voorstellingswijze van het sample interval T_s bij een digitaal filter is gebruik te maken van z^{-1} zoals in fig. 1. De z -operator kan simpel omschreven worden als een tijdsverschuiving over een sample interval (T_s).

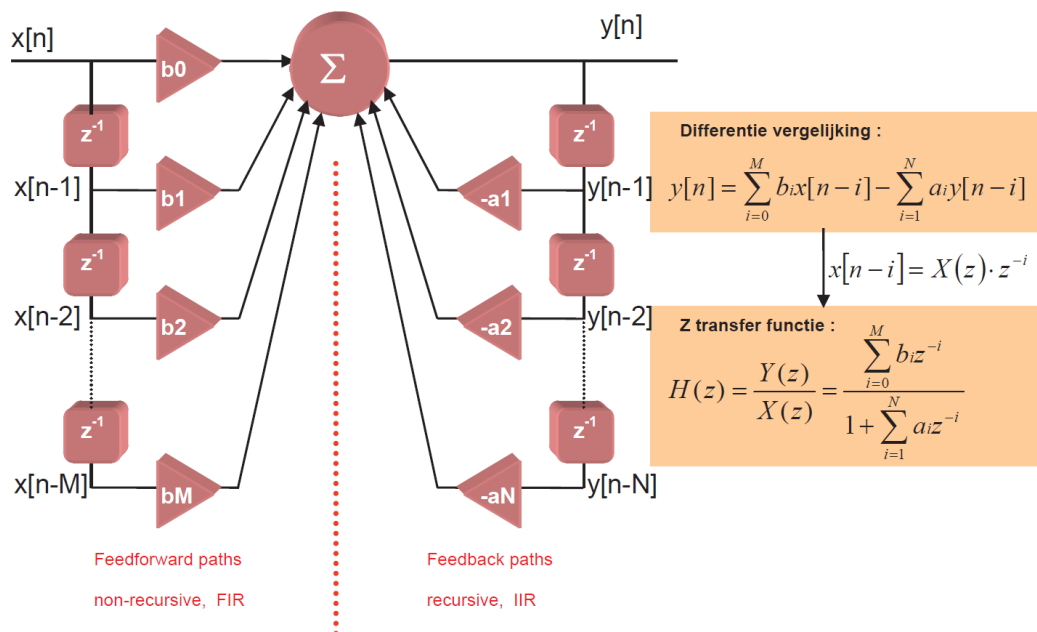


Fig. 1. : Digitale filter topologie met feedforward en feedback.

DOEL

- Het leren omgaan met de **systeemontwerpcyclus**.
- Het leren opstellen van een **systeemconcept**.
- Het leren werken met de **Matlabomgeving** en de **signal processing toolbox** in het bijzonder.
- Inzicht verwerven in diverse **synthesetechnieken** voor **digitale filters** :
Filterontwerp m.b.v. het plaatsen van polen en zeros in het z-vlak.
Ontwerp van recursieve filters gebaseerd op de frequentie responsie eigenschappen van analoge filters met de methode van de bilineaire transformatie. Coëfficiëntquantisatie en het in cascade plaatsen van lage orde recursieve filters ter bevordering van de stabiliteit van het filter. Ontwerp van niet-recursieve filters met de methode van Fourier (window) en met de methode gebaseerd op het algoritme van Parks / McClellan.

C.1 Pole-zero plaatsing

Wat is het effect van polen op de frequentiekarakteristiek? (zie B2)



Wat is het effect van nulpunten op de frequentiekarakteristiek? (zie B3)

Wat is het effect van nulpunten op de eenheidscirkel op de frequentiekarakteristiek?



Waar moeten de nulpunten liggen voor een LDL-, HDL-, BDL-filter?



X= LDL, HDL

Ontwerp een 2^{de} orde recursief X-filter dat een sample-interval gebruikt van 1ms, een poolresonantie heeft met een centerfrequentie van 200Hz, een bandbreedte van 40Hz. Gebruik hierbij de methode van de pole-zero plaatsing in het z-vlak (fig.2). Een omwenteling op de eenheidscirkel komt overeen met een frequentiewijziging van 0 tot de samplefrequentie f_s . Het verband tussen de benodigde bandbreedte BW en de straal r van de polen kan benaderd worden door: $r \approx 1 - (BW/f_s)\pi$ (voor $r > 0.9$). Voor de

nulpunten worden twee transmissienulpunten gebruikt. Duid bijgevolg op fig.2 de ontwerpspecificaties aan die u toelaten de polen en zeros te lokaliseren.

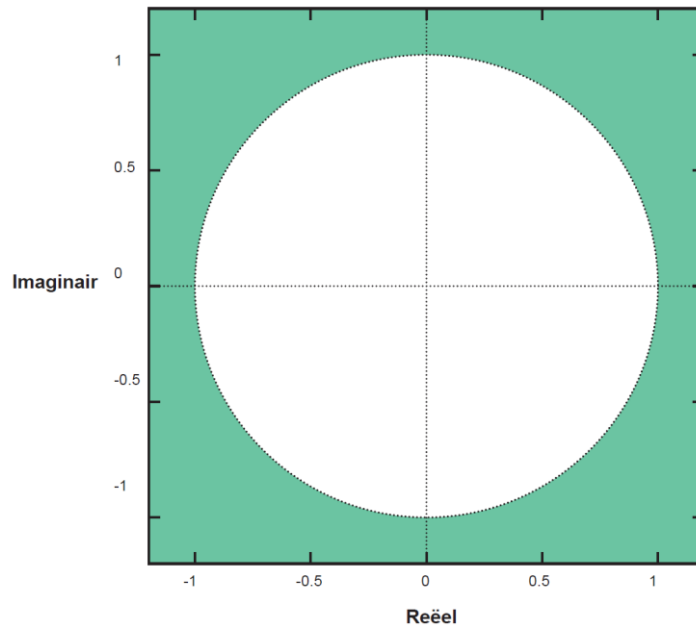


Fig.2 : z-vlak

Bereken de coëfficiënten van de transferfunctie uit de ligging van de nulpunten en polen.

Stel de transferfunctie en de differentievergelijking op.

Bereken de frequentieweergave bij $f=0$, $f_s/4$, $f_s/2$.

Schrijf nu een matlab (*.m) file. Verifieer de coëfficiënten van de transferfunctie en differentievergelijking grafisch door gebruik te maken van de functie *poly*. Controleer de ligging van de polen (*zplane*) en de frequentieresponsie (*freqz*).

C.2 Bilineaire transformatie

Ontwerp een 3^eorde Butterworth LDL-filter met :

-Cut-off frequentie : 1 rad/s

-Sample frequentie : 2Hz

-Transferfunctie in s :

$$H(s) = \frac{1}{(s+1)(s^2 + s + 1)}$$

Gebruik de methode van de bilineaire transformatie en vervang s in de transferfunctie H(s) door $2f_s(1-z^{-1})/(1+z^{-1})$ of $2f_s(z-1)/(z+1)$ om de transferfunctie H(z) op te stellen. (Gebruik eventueel Wolfram Alpha voor het uitwerken van de vergelijking.)

Nota : Dit veroorzaakt geen noemenswaardige afwijking in de frequentieresponsie op voorwaarde dat de cutoff-frequentie beneden $f_s/2$ blijft. Als $s = 2f_s (z-1)/(z+1)$ dan is $(sz/2f_s) + (s/2f_s) - z + 1 = 0$ en is de vergelijking zowel lineair in s als in z. Dit verklaart de benaming bilineaire transformatie.

Schrijf een matlab (*.m) file en verifieer de transferfunctie en differentievergelijking. Controleer hierbij de ligging van de polen (*zplane*) en de frequentie responsie (*freqz*). Hoeveel dB/dec bedraagt de helling van de verzwakking?

Pas de matlab (*.m) file aan om uitgaande van de filterspecificaties de recursieve en niet-recursieve coëfficiënten te laten berekenen (*butter*) door Matlab i.p.v. op manuele wijze in het vorige punt. Transformeer deze transferfunctie van het s- naar het z-domein (*bilinear*). Controleer opnieuw met (*zplane*, *freqz*) de ligging van de polen en de frequentie responsie. Zorg er tevens voor dat de frequentieresponsie van 0Hz tot 0.5Hz wordt weergegeven.

Bespreek kwalitatief het verband tussen de frequentieweergave (amplitude) en de pole-zero-plot.

C.3 (facultatief): Chebyshev-filter

Ontwerp een 8 polen Chebyshev BDL-filter met :

- Bandbreedte : 40Hz
- Centerfrequentie : 200Hz
- Rimpel in doorlaatband : 0.2dB
- Samplefrequentie : 1kHz

Schrijf een Matlab (*.m) file en bereken de filtercoëfficiënten (*cheby1*) om daarna de ligging van de polen en zeros te checken alsook de frequentieresponsie van het filter (*zplane*, *freqz*).

Beeld de frequentie responsie af tussen 100Hz en 300Hz.



Wat is de orde van dit filter? Waarom?



Bespreek kwalitatief het (grafische) verband tussen de frequentieweergave (amplitude) en de pole-zero-plot.




Gebruik dit filter om een geluidssignaal (*wavread*, *sound*) te filteren (*filter*).
Wat hoor je?







C.4 Cauer-filter



Ontwerp een 8^eorde Cauer LDL-filter met :


- Cutoff frequentie : 300Hz
- Rimpel in doorlaatband : 0.5dB
- Minimum sperbandattenuatie : 50dB
- Sample frequentie : 4kHz

Schrijf een matlab (*.m) file en bereken de filtercoëfficiënten (*ellip*) om daarna de ligging van de polen en zeros te checken alsook de frequentieresponsie van het filter (*zplane, freqz*). 

Onderzoek de invloed van de coëfficiëntnauwkeurigheid op de ligging van de polen en zeros en op de amplituderresponsie van het filter. Simuleer een coëfficiëntnauwkeurigheid van 16 digits, 6 digits en 5 digits. Beeld hierbij de amplituderresponsie af tussen 0Hz en 2kHz en tussen -80dB en 20dB. Vergelijk deze 3 resultaten (z-vlak en amplitude resp.). Zijn alle nauwkeurighedsbenaderingen bruikbaar?  

Vervolledig de matlab (*.m) file en bereken de polen van $H(z)$ voor de drie nauwkeurighedsbenaderingen (*roots*). Wat besluit je hieruit?  

Vervolledig de matlab (*.m) file en verifieer je vorig besluit met een onderzoek naar de impulsresponsie bij deze verschillende coëfficiëntnauwkeurigheden.  

Leg vervolgens het verband m.b.t. stabiliteit tussen alle analyses (z-vlak, amplitude resp., straal van de polen en impulsresponsie) bij deze verschillende coëfficiëntnauwkeurigheden. 

C.5 Realisatie met 2^{de} orde-trappen

Ontwerp opnieuw het filter uit opdracht 4. Om een betere stabiliteit te verwezenlijken met een beperkte nauwkeurigheid van de coëfficiënten, splitsen we het 8^e orde filter op in vier 2^eorde filters. De tweede-orde-trappen worden in cascade geplaatst.

$$H(z) = \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}} = \frac{K(z-z_1)(z-z_2)(z-z_3)(z-z_4)(z-z_5)(z-z_6)(z-z_7)(z-z_8)}{(z-p_1)(z-p_2)(z-p_3)(z-p_4)(z-p_5)(z-p_6)(z-p_7)(z-p_8)}$$

Dit kan dus herschreven worden als:

$$H(z) = KH_1(z)H_2(z)H_3(z)H_4(z)$$

Waarbij:


$$H_1(z) = \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)} \quad H_2(z) = \frac{(z - z_3)(z - z_4)}{(z - p_3)(z - p_4)}$$

$$H_3(z) = \frac{(z - z_5)(z - z_6)}{(z - p_5)(z - p_6)} \quad H_4(z) = \frac{(z - z_7)(z - z_8)}{(z - p_7)(z - p_8)}$$

Nota: Het is belangrijk met volgende twee ontwerpregels rekening te houden :

1. Van alle polenparen is het polenpaar dichtst bij de eenheidscirkel datgene met de meeste invloed op de amplitude responsie (heeft de hoogste Q). Deze invloed kan worden afgezwakt door er het dichtst bij gelegen paar van zeros mee te associëren. Dit dient voor de andere polenparen en zeroparen uiteraard ook te gebeuren.

2. De sub-filters dienen dan in cascade te worden geplaatst zodat op de eerste positie diegene staat met de minste invloed (van Q laag tot Q hoog). Dus in volgorde van invloed of in volgorde van afstand van de polen tot de eenheidscirkel. De laatste sub-filter is bijgevolg diegene met grootste invloed op de amplitude responsie (hoogste Q).

Pas de matlab (*.m) file uit opdracht 4 aan zodat je kan aantonen dat een juiste cascadering van de 2^eorde sub-filters (zp2sos) zelfs stabiel is met een coëfficiëntnauwkeurigheid van 2 digits. 

C.6 FIR-filter met lineaire fase

Nota: De recursieve filters hebben slechts weinig coëfficiënten nodig. Ze gebruiken vorige uitgangswaarden waardoor slechts weinig vermenigvuldigingen moeten worden uitgevoerd. Dit maakt hen vanuit rekenkundig standpunt gezien efficiënt en snel. Het probleem is echter dat ze uitermate gevoelig zijn voor coëfficiënt afwijkingen met een onstabiel gedrag als gevolg. Dit probleem kan worden geminimaliseerd door 2^eorde subfilters in cascade te plaatsen.

Een tweede probleem is echter moeilijker te omzeilen met name dit van de faseresponsie. De fase karakteristiek is niet lineair en dit heeft een

belangrijke golfvorm distorsie voorgevolg. In audio toepassingen vorm dit geen bezwaar, maar in sommige anderetoepassingen is dit onaanvaardbaar.

In toepassingen waar een lineair faseverloop wordt vereist gebruiken we FIR- of niet-recursieve filters. Deze filters vereisen een groot aantal coëfficiënten omwille van de theoretisch oneindige hoeveelheid impulsresponsies die nodig zijn voor de gewenste frequentieresponsie.

Ontwerp een 15 coëfficiënt causal LDL-filter met :

-Cutoff frequentie : 250Hz

-Sample frequentie : 1kHz

-Schrijf hiervoor een matlab (*.m) file en bereken de FIR-coëfficiënten met de methode van Fourier (*fir1*) en maak in deze functie gebruik van een Hamming window. Controleer de frequentie responsie van het LDL filter (*freqz*). Wat valt je op m.b.t. het faseverloop van het filter?



C.7 (Facultatief): FIR-filter met Parks / McClellan

Ontwerp een FIR-filter met 32 coëfficiënten, een samplefrequentie van 5kHz en volgende gewenste amplituderesponsie :

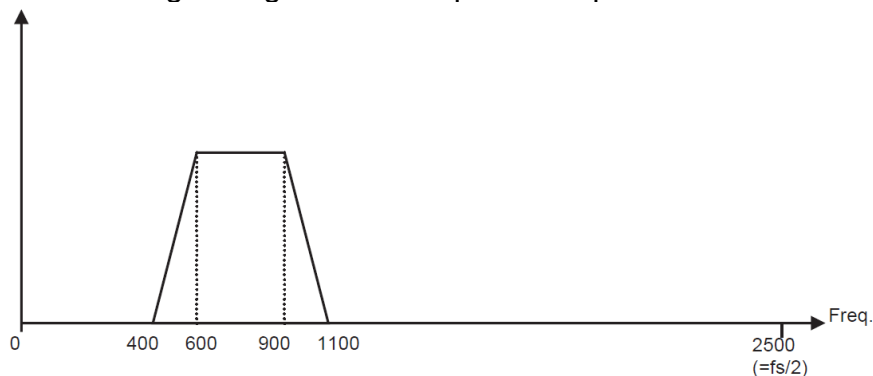


Fig.3 : gewenste FIR-amplituderesponsie.

Schrijf hiervoor een matlab (*.m) file en bereken de FIR-coëfficiënten met de methode van Parks / McClellan (*remez*). Controleer de frequentie responsie van het BDL filter (*zplane*, *freqz*). Beeld 400 frequentiepunten af.



FACULTEIT INDUSTRIELE INGENIEURSWETENSCHAPPEN
CAMPUS DE NAYER (Thomas More)
Jan De Nayerlaan 5
2860 SINT-KATELIJNE-WAVER, België
tel. + 32 15 31 69 44
iiw.thomasmore.denayer@kuleuven.be
www.iw.kuleuven.be

