

Practicum Toegepaste Signaalanalyse

Eindopdracht

Rick van de Zedde & Volker Nannen

11 februari 2002

Inhoudsopgave

1	Probleembeschrijving	2
2	Implementatie	2
2.1	De verschillende methoden	2
2.2	Implementatie van een eigen AR schatter: de covariantie methode	3
3	Energie	4
4	Nauwkeurigheid	5
4.1	AR filter	5
4.2	ARMA filter	8
4.3	ARMA-herkenning met veel AR-parameters	8
4.4	Spectrale resolutie	11
4.5	Conclusie	11
5	Overdrachtsfuncties en Coherentie	15
5.1	Implementatie	15
5.2	Conclusie	17
6	Evaluatie	17
6.1	Implementatie	17
6.2	Energie	18
6.3	Nauwkeurigheid	18
6.4	Overdracht en Coherentie	18
6.5	Mogelijkheden voor verdere analyse	19
	Referenties	19

1 Probleembeschrijving

De spectrale schatter van een eindig signaal wordt meestal via de Fourier Transformatie berekend. Dit levert, na smoothing en windowing, de gemiddelde sterkte van een frequentie over het gemeten signaal. Er zijn echter ook andere analysemethoden. Eén ervan is de parametrische methode. Deze is verder onder te verdelen in autoregressieve (AR) modellen, moving average (MA) modellen en combinaties van die twee (ARMA). In deze opdracht beperken we ons tot de AR modellen.

Bij de parametrische methode wordt de Fourier Transformatie niet van het signaal zelf gemaakt maar van een parametrisch model van dit signaal. Een signaal wordt daarbij beschouwd als bestaande uit een parametrisch model plus ruis. Ruis bevat per definitie geen informatie en het voldoende aantal parameters plus de variantie van de ruis bevatten dus dezelfde informatie als het echte signaal.

Het aantal relevante parameters is bekend of wordt geschat uit b.v. de autocovariantiefunctie. Ook is het nuttig om te weten dat bij de z-transformatie per pool twee recursieve parameters nodig zijn. Om b.v. 4 frequentie pieken te modelleren zijn minimaal 8 AR parameters nodig.

Bij de conventionele Fourier Transformatie wordt vaak gesmoothed om de grafiek leesbaarder te maken. Dit wordt verweten aan de in het signaal aanwezige ruis. De parametrische methode daarentegen levert vanaf het begin een zuivere grafiek op omdat de ruis al op een eerder tijdstip werd verwerkt.

Omdat de ruis al uit het signaal is verwijderd verwachten wij dat de parametrische methode een nauwkeurigere spectrale schatter oplevert dan de conventionele methode. Met nauwkeurig bedoelen we dat de schatting zo dicht mogelijk komt bij de spectrale verdeling van een oneindig signaal uit dezelfde bron.

Tenslotte willen we weten of de parametrische methode geschikt is om ook voor beperkte frequentiegebieden de spectrale power te bepalen en om de overdrachtsfuncties en coherentie te bepalen.

We onderzoeken de parametrische methode dus met betrekking tot

1. implementatie
2. prestatie, met in het bijzonder
 - (a) energie
 - (b) nauwkeurigheid
 - (c) overdrachtsfuncties
 - (d) coherentie.

2 Implementatie

2.1 De verschillende methoden

Voor parametrisch AR analyse zijn er twee standaard-methoden: de YULE-WALKER methode en de LEAST SQUARE methode. Bij de YULE-WALKER

methode wordt een schatter van de autocorrelatie gebruikt om de parameters te berekenen. De BURG methode is daar een variatie op die, vergelijkbaar met PRINCIPAL COMPONENT ANALYSIS, alle parameters apart berekend voor het maximum aan informatie. Het weglaten van de laatste parameter is bij Burg dus minder erg dan bij de standaard YULE-WALKER methode waar de informatie over alle parameters gelijk verdeeld is.

De LEAST SQUARE methode wordt ook linear prediction genoemd. Om n parameters te schatten wordt het signaal op alle tijdstippen als lineaire combinaties van de n vorige tijdstippen opgevat. De n parameters moeten nu het signaal op alle tijdstippen zo voorspellen dat de kleinste kwadratische fout ontstaat. Afhankelijk van of en hoe bij deze methode het gemeten signaal met nullen wordt aangevuld spreekt men¹ ook van de AUTOCORRELATIE en de COVARIANTIE methode.

Volgens de literatuur² is een AR model dat afkomstig is van de Yule-Walker methode gegarandeerd stabiel, terwijl dat bij de Least Square methode niet het geval is. Darentegen is de Least-Square methode nauwkeuriger in het schatten van de parameters. Vooral bij kleine steekproeven is Yule-Walker daarom minder geschikt.

We willen de parameters niet gebruiken om een filter te construeren maar om de data te analyseren. We vinden dus nauwkeurigheid belangrijker dan stabiliteit. Bovendien is niet gegarandeerd dat de data afkomstig zijn van een stabiel systeem en de keuze voor een gegarandeerd stabiel systeem lijkt een vervalsing van de data.

2.2 Implementatie van een eigen AR schatter: de covariantie methode

Matlab levert standaard twee manieren om via een AR model direkt een fourier getransformeerde te leveren: de functies PYULE en PBURG. Omdat uit de beschrijving niet helemaal duidelijk is hoe deze zijn geïmplementeerd gaan we ook zelf een methode implementeren, waarbij we eerst de parameters schatten en daaruit het spectrum afleiden.

Voor het schatten van de parameters implementeren we de zogenaamde covariantie methode. Deze methode werkt zonder het toevoegen van nullen en heeft daarom ook geen bias. Wel wordt de steekproef daardoor kleiner om een waarde n , die gelijk is aan de orde van het systeem.

Als we de parameters van een AR model van orde n zoeken en we een signaal $X(t)$ van lengte m hebben dan maakt onze implementatie een $(m - n - 1)$ bij n matrix:

$$M = \begin{bmatrix} X(0) & X(1) & \dots & X(n) \\ X(1) & X(2) & \dots & X(n+1) \\ \dots & \dots & \dots & \dots \\ X(m-n-1) & X(m-n) & \dots & X(m-1) \end{bmatrix} \quad (1)$$

¹[2], pagina 92 onderaan

²[2], pagina 93 onderaan, 94 bovenaan

Nu zoeken we een parameter vector $P = [a_1, a_2 \dots a_n]$ die, vermenigvuldigd met deze matrix, het target-sigitaal

$$Y = X(t) \{ t \mid n \leq t \leq m \} \quad (2)$$

produceert, waarbij de kwadratische fout minimaal moet zijn:

$$\begin{bmatrix} X(0) & X(1) & \dots & X(n-1) \\ X(1) & X(2) & \dots & X(n) \\ \dots & \dots & \dots & \dots \\ X(m-n-1) & \dots & \dots & X(m-1) \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix} \approx \begin{bmatrix} X(n) \\ X(n+1) \\ \dots \\ X(m) \end{bmatrix} \quad (3)$$

of, in het kort,

$$M * P \approx Y. \quad (4)$$

Door beide kanten op de linkerkant door M te delen hebben we:

$$P \approx M^{-1}Y. \quad (5)$$

In Matlab kan dit direct worden geïmplementeerd door te schrijven

$$P = M \setminus Y. \quad (6)$$

waarop Matlab de P met de kleinste kwadratische fout berekend. De spectrale schatter over 512 frequentiepunten berekenen we nu met behulp van de Matlab functie `FREQZ`:

$$\begin{aligned} H &= \text{freqz}(1, P, 512); \\ P_{xx} &= (H * \text{conj}(H)) / 512; \end{aligned} \quad (7)$$

3 Energie

We hebben onze implementatie vergeleken met de prestaties van de Matlab functies `PYULE`, `PBURG` en `PSD`³. Daarbij viel op dat onze implementatie het zelfde powerspectrum leverde, op een belangrijk verschil na: terwijl de energie van `PYULE` en `PBURG` min of meer in de beurt laag van het door `PSD` berekend powerspectrum, was de energie bij onze implementatie constant. Een schaling van de input (bv. $X = X * 100$) had geen invloed op onze spectrale schatter maar wel op `PYULE` en `PBURG`.

Dat `PYULE` en `PBURG` wel energie kunnen beschrijven lijkt raar, want theoretisch kan een AR model nooit de energie beschrijven. Gelukkig staan sommige Matlab versies toe om de recursieve parameters die door `PYULE` werden berekend op te vragen:

$$[P_{yule}, w_{yule}, parameters] = \text{pyulear}(X, \text{aantal}_{parameters}). \quad (8)$$

Deze parameters kunnen ook direct met de functie `LPC` worden berekend. Ook `LPC` maakt van het Yule-Walker algoritme gebruik. Als je uit deze parameters via `FREQZ` het powerspectrum berekend dan is dit even constant als dat van

³`PSD` berekend het powerspectrum via de Welsh methode, dus een van conventionele oplossingen.

onze implementatie. Matlab maakt dus achter de schermen nog gebruik van MA-parameters. In figuur 1 op pagina 6 is te zien dat het spectrale spectrum van PYULE een geschaalde versie is van de spectrale schatter die alleen op grond van de parameters was berekend. Het MA gedeelte kan dus geen onderscheid maken tussen frequenties. Er kan daardoor maar sprake zijn van slechts één MA-parameter.

Conclusie: een pure AR-schatter is nooit in staat om de energie van een signaal adequaat te beschrijven. We gaan daarom in de verdere opdracht altijd met implementaties werken die naast de AR-parameters ook een MA-parameter voor de schaling van de energie heeft.

Figuur 1 op pagina 6 laat een aantal spectrale schatters zien: De conventionele Welsh-schatter, de implementatie van het Yule-Walker algoritme en onze eigen implementatie. CONTROL-YULE is een herimplementatie op grond van alleen het AR-gedeelte van PYULE. Duidelijk is te zien dat deze even constant is als onze eigen implementatie.

Het aantal geschatte parameters is voor alle functies tien. Bij langere signalen en meer dan vier parameters zijn er geen zichtbaar verschillen tussen PYULE en PBURG. Daarom is alleen de schatting van de PYULE functie weergegeven.

Om ook de energie te kunnen beschrijven is dus minimaal één MA parameter nodig. Deze geeft de schaling van de input weer. Helaas is uit de ons beschikbare literatuur geen eenvoudige implementatie voor een MA-schatter in combinatie met een AR-schatter af te leiden. Ook blijft het een radsel hoe dit binnen PYULE geïmplementeerd is.

4 Nauwkeurigheid

In figuur 1 op pagina 6 was te zien dat zowel PSD als PYULE dicht bij de echte energie van het signaal kwamen. De preciese fout hangt natuurlijk van de omstandigheden af en verschilt per signaal.

Maar naast de totale fout willen we ook weten hoe groot de fout per frequentiegebied is. Daarbij onderzoeken we drie soorten gebieden: de pieken (het echte signaal), overgangsgebieden en dalgebieden.

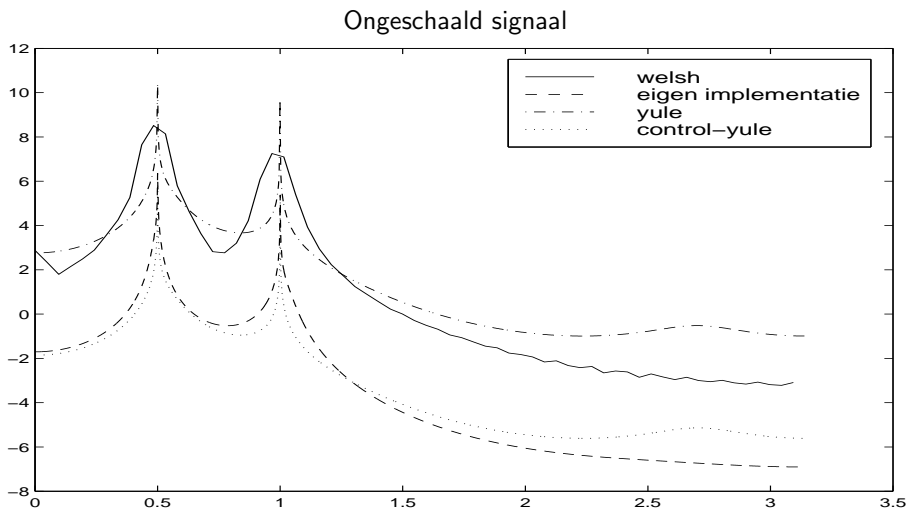
Om de prestaties van PSD en PYULE te vergelijken maken we verschillende kunstmatige signalen door gausische ruis te filteren. Het voordeel tegenover echte signalen is dat de filter-parameters bekend zijn en we het controle-spectrum daarom direct uit de filter-parameters kunnen berekenen.

4.1 AR filter

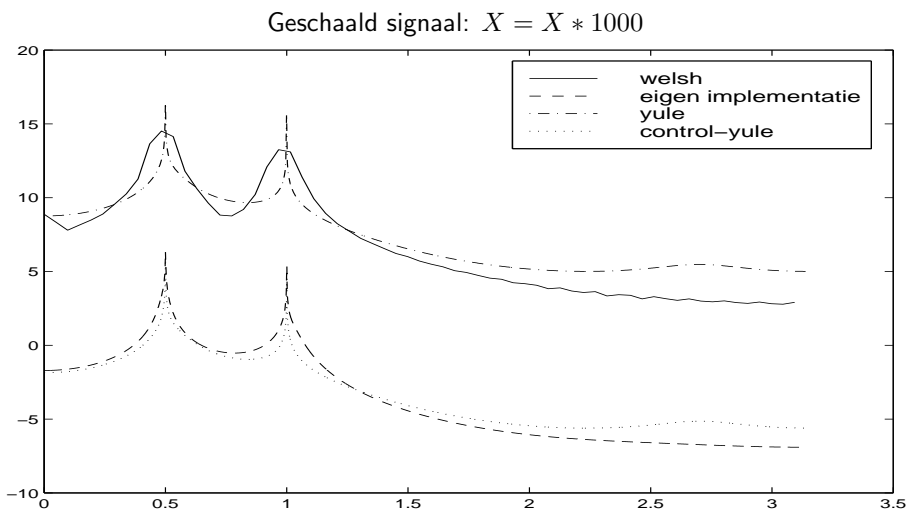
In figuur 2 op pagina 7 is het signaal gemaakt door middel van een puur recursieve filter. Boven op de pagina zijn het controle-spectrum en de spectra van PSD en PYULE te zien, op logaritmische schaal getekend. Daaronder is de fout van PSD en PYULE ten opzichte van de het controle spectrum te zien.

Boven op de pagina werden 4096 sample waarden gebruikt. Beide schatters kunnen de pieken en dalen van het recursieve signaal goed volgen. Ze maken beide een vergelijkbare fout voor zowel de pieken als de dalen. Bovendien is de fout van beide signalen zichtbaar evenredig aan de energie van het signaal

Figuur 1:
 Verschillende Spectrale Schatters (getekend op logaritmische schaal)
 en Totale Energie

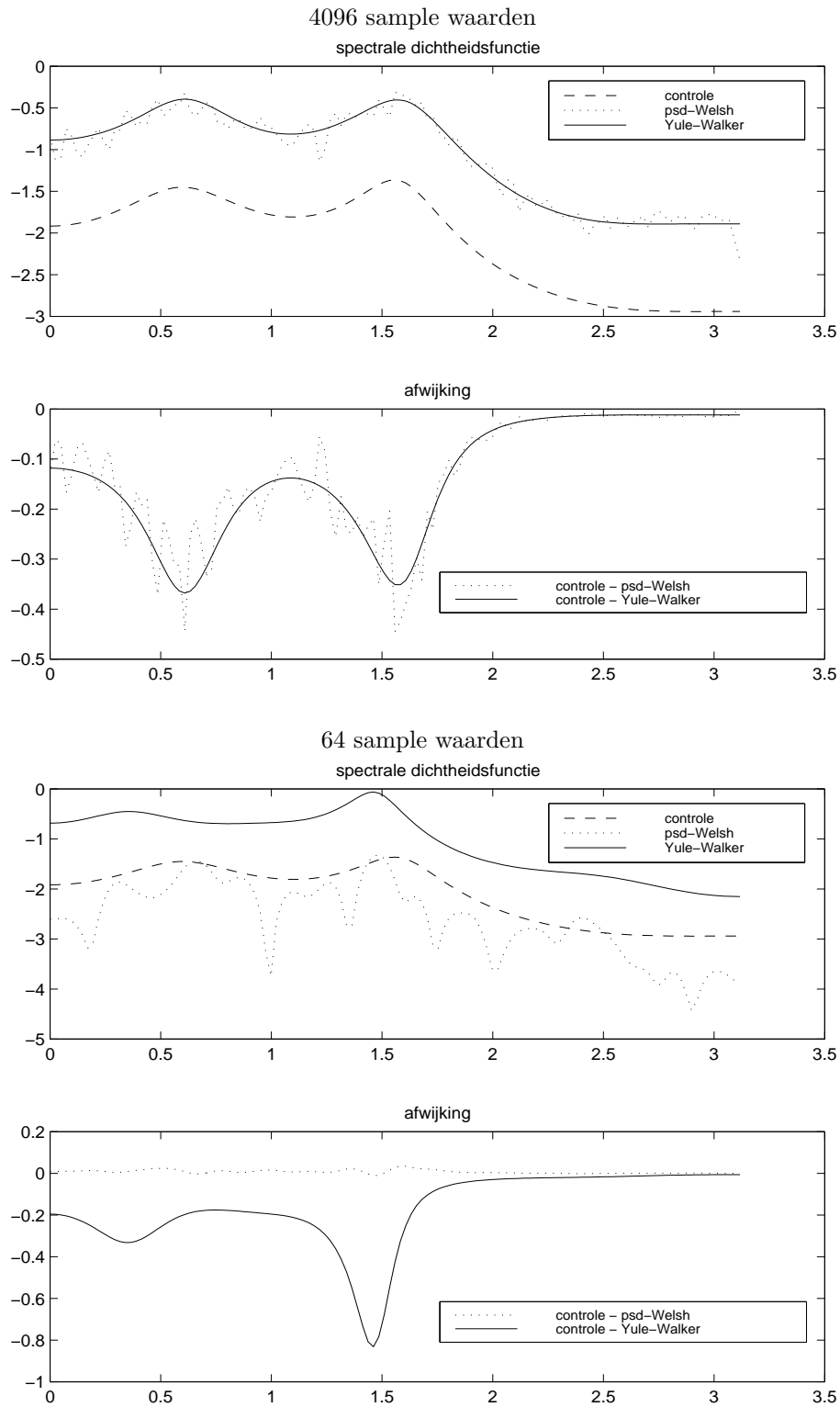


$$\text{energie : } \begin{cases} \text{welsh} & = 2.655e + 06 \\ \text{eigen} & \\ \text{implementatie} & = 8.032e + 02 \\ \text{yule} & = 4.715e + 06 \\ \text{controlyule} & = 1.180e + 02 \\ C_{xx}(0) & = 8.371e + 06 \end{cases}$$



$$\text{energie : } \begin{cases} \text{welsh} & = 2.655e + 12 \\ \text{eigen} & \\ \text{implementatie} & = 8.032e + 02 \\ \text{yule} & = 4.715e + 12 \\ \text{controlyule} & = 1.180e + 02 \\ C_{xx}(0) & = 8.371e + 12 \end{cases}$$

Figuur 2:
Prestaties van PSD en PYULE: puur AR signaal



in een bepaald frequentiegebied. Kortom, beide schatters zijn een geschaalde versie van het signaal.

De schaling is trouwens te verklaren uit het feit dat bij de berekening van het controle spectrum geen rekening gehouden wordt met de energie van de ruis. Een twee keer zo sterke ruis zal daarom ook de spectra van de schatters met de faktor vier schalen maar geen invloed hebben op de controle-functie.

Beneden op de pagina zijn maar 64 sample waarden gebruikt. Duidelijk is te zien dat PYULE het echte spectrum veel beter van de ruis kan scheiden dan PSD. Maar voor beide functies geldt dat de fout per frequentiegebied weer evenredig is aan de energie.

4.2 ARMA filter

In figuur 3 op pagina 9 is een filter gebruikt die ook meerdere MA-parameters heeft: een stop-pas-band filter. Omdat de PYULE functie maar een enkele MA-parameter gebruikt kan zij dit onmogelijk reproduceren.

Boven op de pagina met 4096 samplewaarden is goed te zien hoe PYULE er hopeloos naast zit. Buiten het stopbandgebied en het bijbehorende overgangsggebied lijken de prestaties van PSD en PYULE nog steeds heel erg op elkaar. Maar van de stopband is in de spectrale schatter geen spoor terug te vinden. PSD daarentegen volgt de controlefunctie nauwkeurig zowel in het pas- als in het stopbandgebied.

Beneden met maar 64 samplewaarden lijkt het vergelijken van de twee methodes moeilijker. PYULE volgt het controle spectrum weer vrij betrouwbaar, behalve in het stopbandgebied en de bijbehorende overgangsggebieden, waar PYULE weer hopeloos naast zit.

PSD slaat er wel in het stopbandgebied te beschrijven, maar de fout is groot en aangezien de algemene onbetrouwbaarheid van PSD over het hele spectrum heeft deze informatie maar weinig waarde. PSD detekteerd naast de bestaande stopband immers nog een reeks niet bestaande stopbanden.

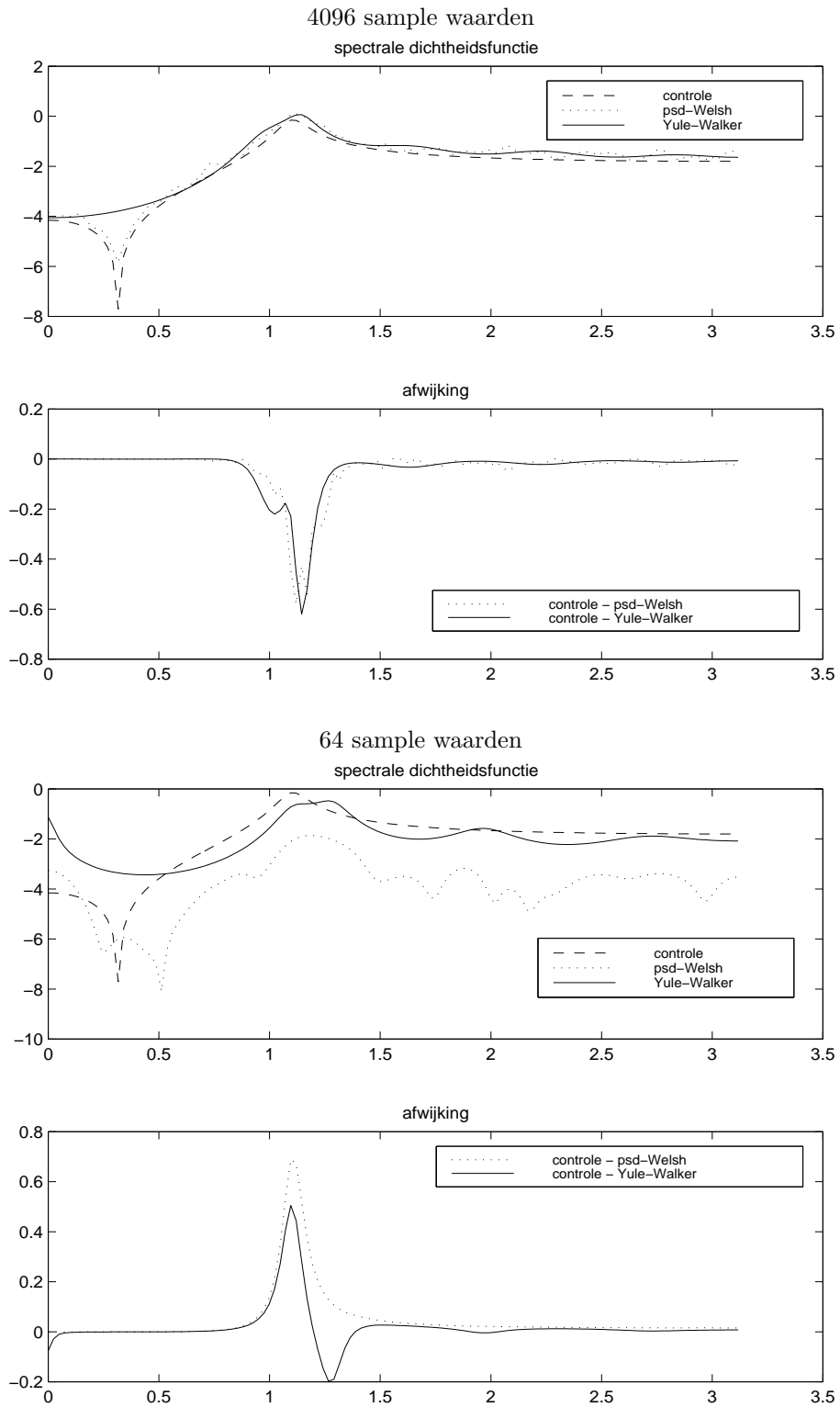
Zowel rechts als links is in de grafiek “afwijking” geen fout van PYULE ten opzichte van het controle spectrum te zien. Dit komt omdat de absolute fout na verhouding te klein is. Hij heeft een logaritmische schaling nodig, zo als boven.

4.3 ARMA-herkenning met veel AR-parameters

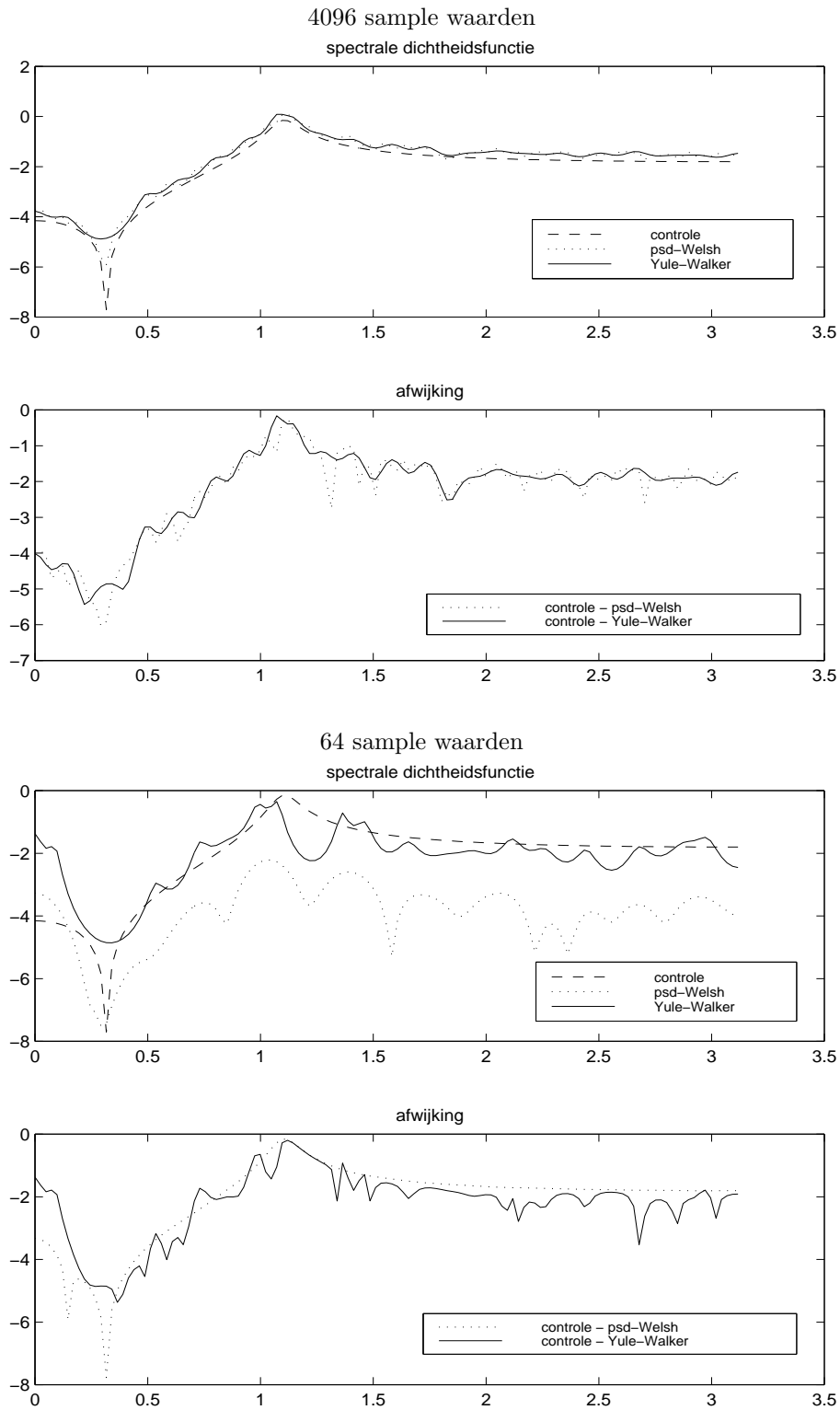
Een pure AR-schatter als PYULE is natuurlijk niet geschikt voor het herkennen van een stopband. Heelaas hebben we geen werkende implementatie van een ARMA-schatter. Maar we kunnen het gebrek van PYULE in het stopbandgebied proberen op te vangen door meer AR-parameters te gebruiken. Tot nu toe hebben we 10 parameters gebruikt. In figuur 4 op pagina 10 gaan het met 40 parameters proberen.

Bij 4096 samplewaarden kan PYULE het controle spectrum in het overgangsggebied rond het stopbandgebied goed volgen. Het stopband zelf is nog steeds niet te zien. Een geoefende persoon zou in staat zijn om uit de aanwezigheid van de kuil in het overgangsggebied een stopband te concluderen, maar de exacte demping valt op geen manier af te leiden.

Figuur 3:
Prestaties van PSD en PYULE: ARMA signaal met 10 parameters



Figuur 4:
ARMA signaal, 40 parameters



Bij 64 wordt het nog moeilijker. Rond het stopbandgebied is weer een kuil te zien, maar dat geldt in minder mate ook voor andere frequentiegebieden waar helemaal geen stopband te meten valt. Even als bij PSD wordt PYULE nu onbetrouwbaar.

4.4 Spectrale resolutie

De spectrale resolutie is een van de hoofdredenen om van de parametrische methode gebruik te maken, althans de literatuur⁴. Een AR-model bestaat immers theoretisch uit polen in het z -vlak, wat overeenkomt met de pieken in het signaal. De redenatie is, dat als de polen goed geplaatst zijn, juist de pieken te voorschijn komen. AR-detektie is dus piek-detektie.

Voor de verificatie maken we een signaal door gaussische ruis⁵ en twee signalen met vaste frequenties ω_1 en ω_2 te combineren. De afstand tussen ω_1 en ω_2 noemen we ϕ :

$$x[t] = \cos\left(t * \frac{\pi}{\omega_1}\right) + \cos\left(t * \frac{\pi}{\omega_2}\right) + \sqrt{e_{ruis}} * (\text{random} - 0.5) \quad (9)$$

$$\phi = \omega_2 - \omega_1 \quad (10)$$

We proberen het eerst met PYULE, met een tijdreeks van 4096 samples en 10 AR-parameters. Figuur 5 op pagina 12 op pagina 12 laat zien dat PYULE een slechte spectrale resolutie heeft. Boven op de pagina doet PSD het zichtbaar beter. Beneden kan PYULE helemaal geen twee pieken meer scheiden. Voor minder parameters, sterkere ruis of een kortere tijdreeks geldt dat deze prestaties alleen maar slechter worden.

Maar we hebben ook nog onze eigen implementatie. Figuur 6 op pagina 13 en figuur 7 op pagina 14 laten zien hoe deze implementatie het doet. Onder de zelfde condities waar PYULE op pagina 12 beneden geen onderscheid meer kon maken tussen de frequenties lukt het met onze implementatie op pagina 13 boven wel.

Daarbij valt op dat de resolutie slechter wordt naarmate de energie van de ruis stijgt. Bij lage ruis doet onze implementatie het dus beter dan PSD, bij sterke ruis doet PSD het beter.

Voor korte tijdreeksen geldt het zelfde als voor lange, alleen is meer energie van de ruis nodig om PSD een voordeel te geven.

4.5 Conclusie

Onze conclusie is dat in pasbandgebieden een parametrische schatter het tenminste evengoed doet als de meest gangbare conventionele methode, de Welsh-methode. De fout is in beide gevallen even groot. Er is geen verschil te zien bij de pieken, de dalen en de overgangsgebieden. Voor kleine samples geldt zelfs dat PYULE veel betrouwbaarder is dan PSD. En PYULE heeft duidelijk minder variatie en is beter te lezen dan PSD.

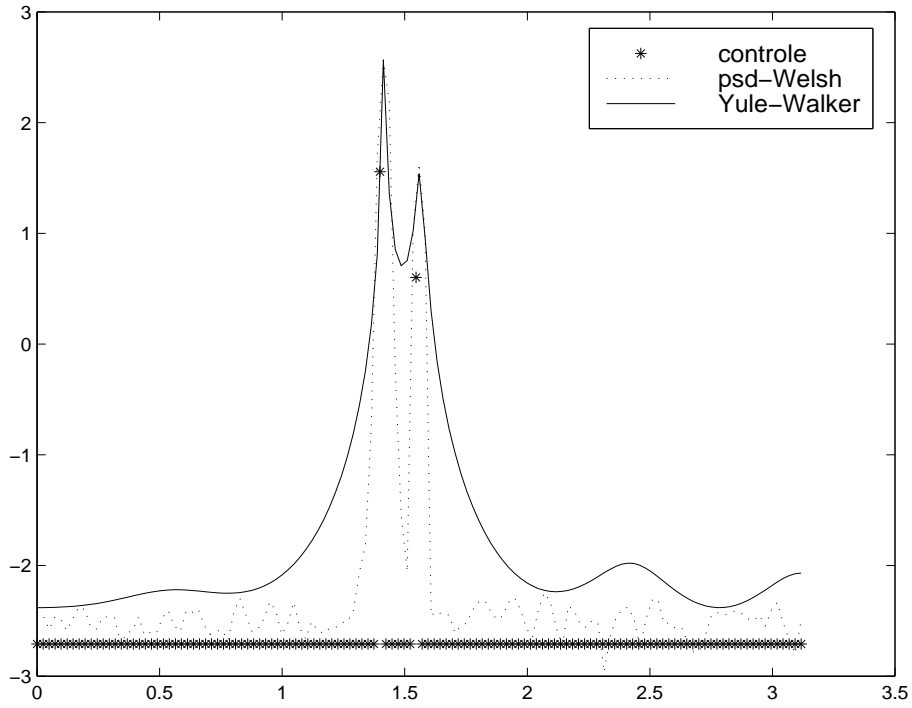
Voor stopbanden geldt dit niet. De overgangsgebieden kan PYULE met een hoog aantal parameters nog even beschrijven. Het stopband zelf en de maximale

⁴[1], pagina 41

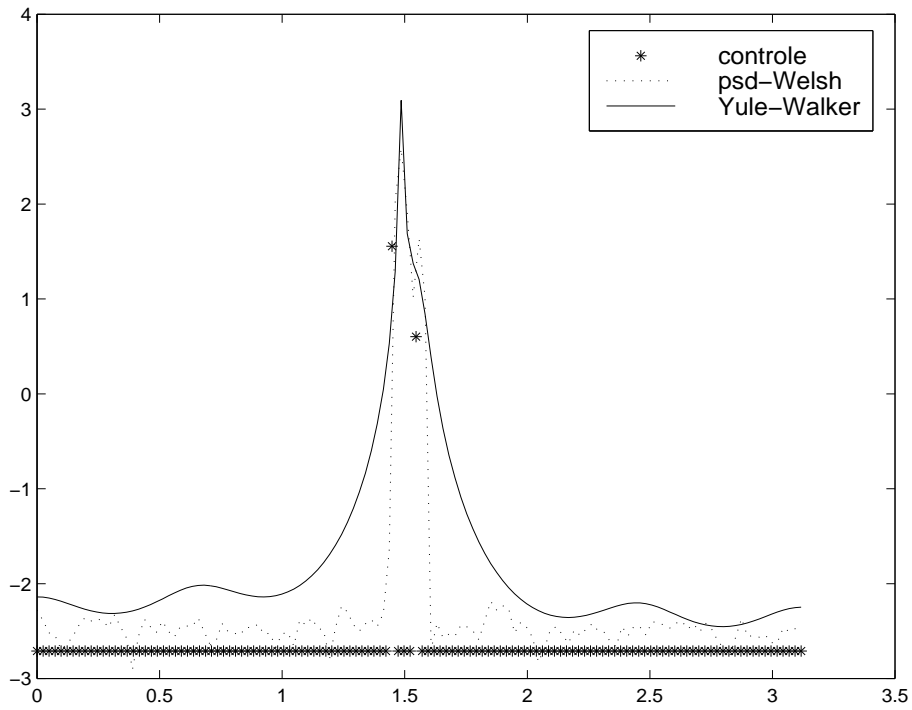
⁵de functie RANDOM in Matlab

Figuur 5:
spectrale resolutie met PYULE

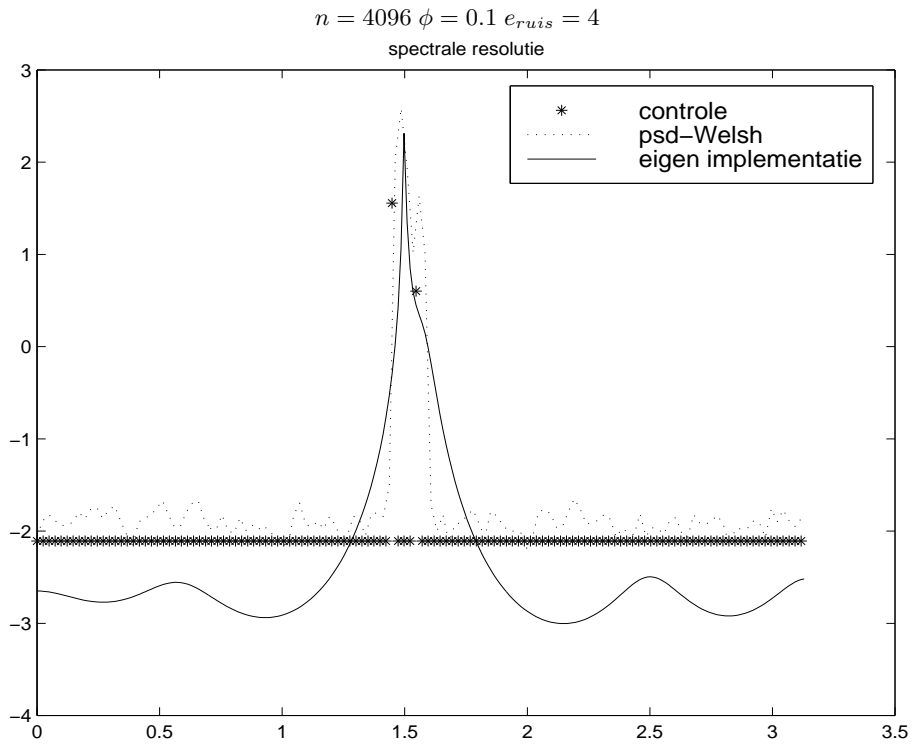
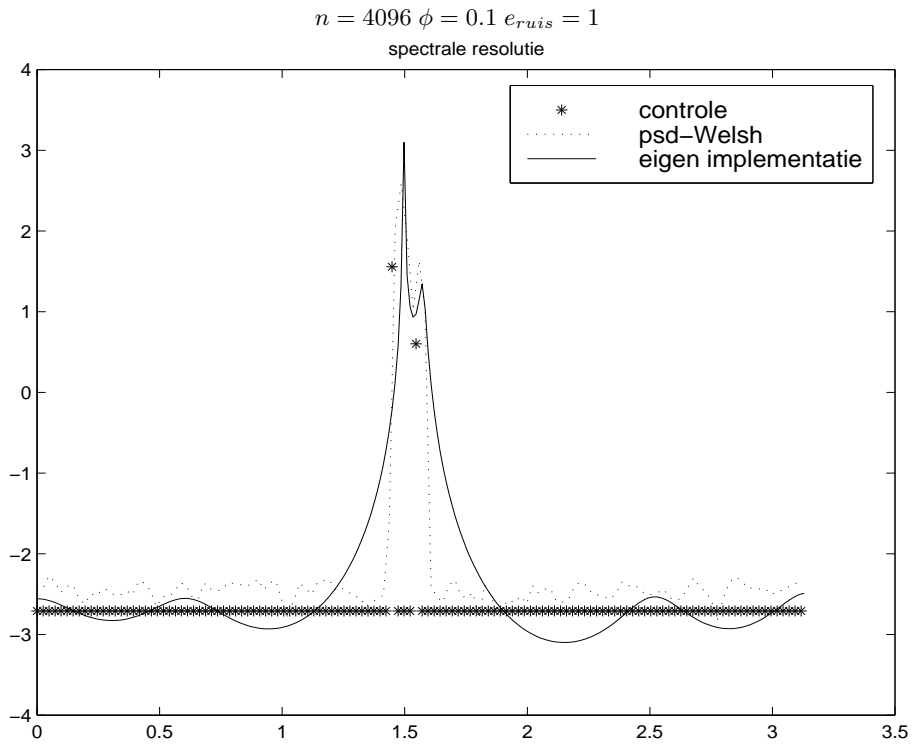
$n = 4096$ $\phi = 0.2$ $e_{ruis} = 1$



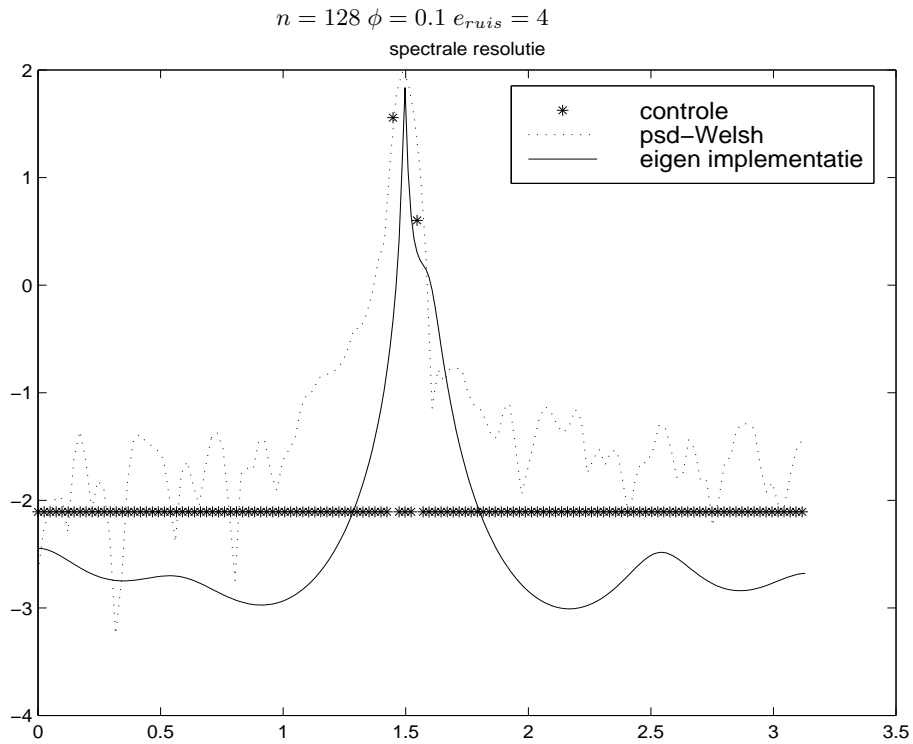
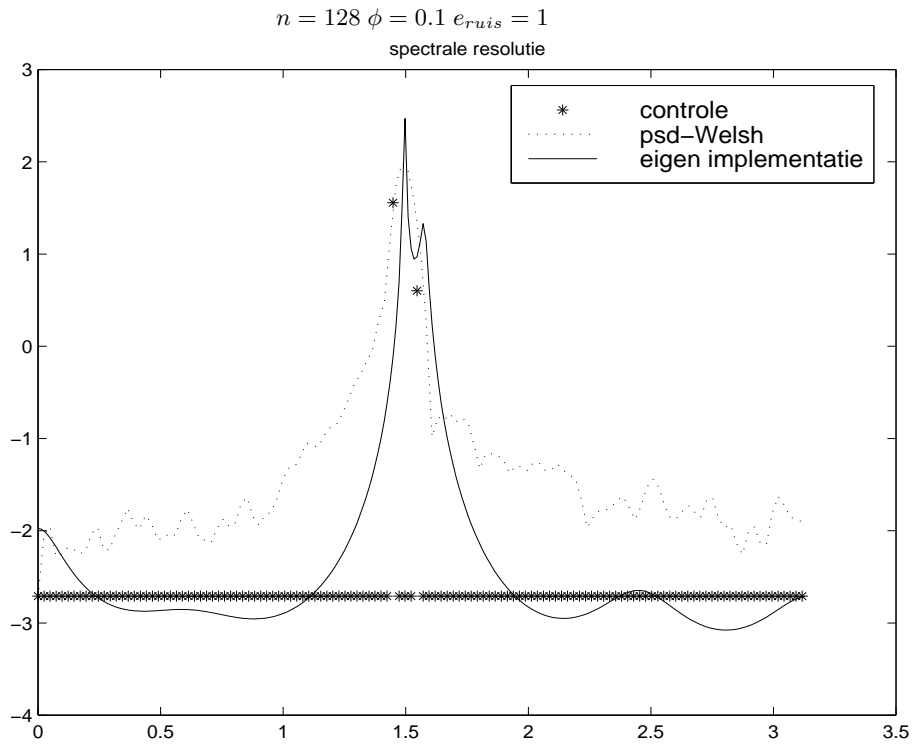
$n = 4096$ $\phi = 0.1$ $e_{ruis} = 1$



Figuur 6:
 spectrale resolutie met onze eigen implementatie, lange tijdreeks



Figuur 7:
spectrale resolutie met onze eigen implementatie, korte tijdreeks



demping zijn niet te zien. De overgangsgebieden gaan gewoon in elkaar over. Om stopband signalen te beschrijven is een puur autoregressieve parametrische methode dus zeker niet geschikt. Verder gaan we ons daarom ook alleen nog met pasband signalen bevatten.

Voor de spectrale resolutie geldt dat de AR-methode niet aan de verwachting voldoet. Welke methode, PSD of de parametrische methode, het beter doet hangt in principe af van de energie van de ruis. In de literatuur staat dan ook dat de hoge spectrale resolutie van de parametrische methode afhankelijk is van een goede signal-to-noise ratio in de data⁶.

5 Overdrachtsfuncties en Coherentie

Uit de vorige sectie blijkt dat de parametrische methode betrouwbare spectra levert, mits het geen stopband signalen zijn. Op zich moet het dus ook mogelijk zijn om deze spectra verder voor de analyse van signalen te gebruiken, bijvoorbeeld voor overdrachtsfuncties en coherentie. Interessant is daarbij of de blindheid van de parametrische methode voor stopbanden bij het berekenen van de overdrachtsfuncties en de coherentie versterkt of verzwakt wordt.

Stopbanden zijn in de signalen niet terug te vinden, de overgangsgebieden rechts en links gaan gewoon in elkaar over. Je zou daarom moeten verwachten dat stopbanden ook in de filter en de coherentie door de bijbehorende waarden uit de overgangsgebieden vervangen zijn.

Op zich zijn de resultaten al voorspelbaar en worden AR-modellen ook nooit voor stopbandsignalen gebruikt. Dit hoofdstuk is er dan ook alleen maar voor de volledigheid. We willen immers globaal vastleggen wat je met het AR-model wel dan niet kunt.

5.1 Implementatie

PYULE en PBURG leveren heelaas alleen powerspectra. We kiezen ervoor om de positieve wortel van de door PYULE berekende spectra in plaats van de fouriergetransformeerde te gebruiken. Hier is dit voldoende om de betrouwbaarheid van overdrachtsfunctie en coherentie te bekijken. Omdat PYULE een reël positief spectrum levert hoeven we in het vervolg geen rekening te houden met negatieve of imaginaire getallen.

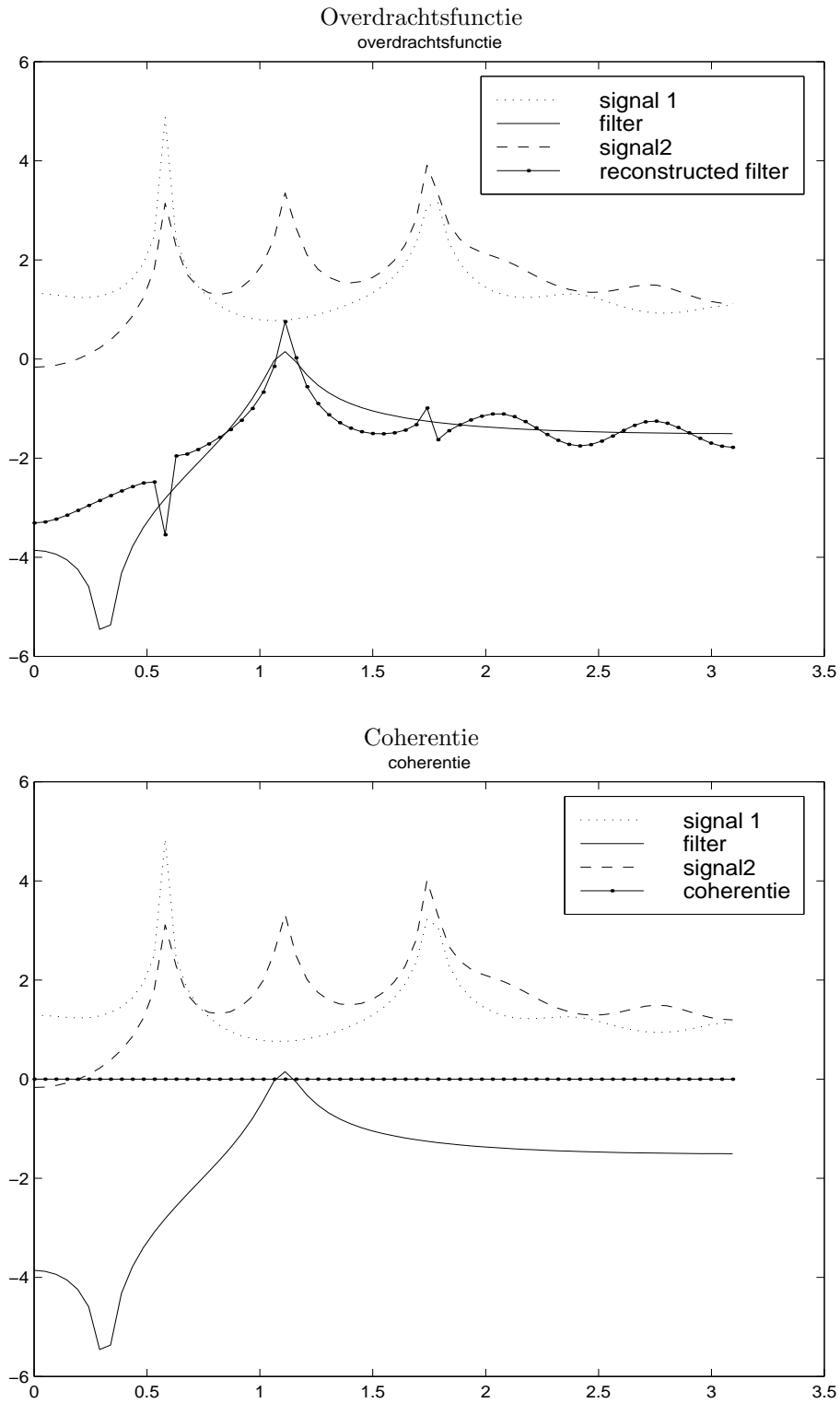
We nemen een willekeurig signaal x ⁷. Dit signaal wordt vervolgens gefilterd met de pas-stopband filter *filter* uit figuur 4 op pagina 10. Dit levert een nieuw signaal y . De erbijbehorende powerspectra zijn P_{xx} en P_{yy} . Omdat we de parameters van *filter* zelf hebben bepaald kunnen we makkelijk het controle filterspectrum P_{filter} berekenen. Het powerspectrum van de reconstrueerde overdrachtsfunctie $P_{overdracht}$ berekenen we nu als volgt (T is de lengte van de frequentie-as):

$$P_{overdracht}(\omega) = \frac{1}{T} * \frac{P_{yy}(\omega)}{P_{xx}(\omega)} \quad (11)$$

⁶[1], pagina 41

⁷door een willekeurige ARMA-filter op gausische ruis toe te passen

Figuur 8:
Overdrachtsfunctie en Coherentie



Voor de coherentie hebben we het kruisspectrum nodig (de wortels vervangen de fouriergetransformeerden en het * voor de geconjugerde van de tweede wortel staat er alleen maar voor de formaliteit):

$$P_{xy}(\omega) = \frac{1}{T} * \left| \sqrt{P_{xx}(\omega) * T} \right| * \left| \sqrt{P_{yy}(\omega) * T} \right|^* \quad (12)$$

En de Coherentie is:

$$Coh(\omega) = \frac{|P_{xy}(\omega)|^2}{|P_{xx}(\omega)| * |P_{yy}(\omega)|} \quad (13)$$

In figuur 8 op pagina 16 is het resultaat te zien. Op de linker kant heeft de gereconstrueerde overdrachtsfunctie de piek in de originele filter goed herkend. Maar van het stopband is niets te zien. In plaats daarvan wordt op de hoogte van de eerste piek een stopband gepostuleerd. Het niet zien van het stopband leidt dus tot nieuwe fouten in de verdere analyse. Maar de fout is beperkt tot de stopband.

De coherentie beneden in figuur 8 op pagina 16 is misschien een teleurstelling. Met de formules 12 en 13 valt de coherentie niet te berekenen. Het resultaat moet constant 1 zijn. De redenen waarom we via de conventionele methode toch iets kunnen zien is de fout die tijdens het smoothen wordt gemaakt. Deze fout geeft in feite pas betekenis aan de coherentie.

P_{xy} kan ook direkt via de parametrische methode uit de kruiscorrelatie berekend worden, PYULE gebruikt immers intern de autocorrelatie. Die zou vervangen moeten worden door de kruiscorrelatie. Uit een zo verkregen spectrum moet dan de coherentie af te leiden zijn. Helaas weten we niet hoe dit in Matlab kan.

5.2 Conclusie

De overdrachtsfunctie die met de AR-parametrische methode kan worden berekend is niet erg betrouwbaar. De meeste signalen zijn weliswaar pasband- en geen stopbandsignalen, enkele pieken en een ruisachtige achtergrond. De stopbandblindheid van deze methode speelt dan geen grote rol. Overdrachtsfuncties echter worden wel degelijk door hun stopband-kwaliteiten gekarakteriseerd. Een ontbrekend of fout gelokaliseerde stopband is niet acceptabel.

Maar een ARMA-parametrische methode zou dit kunnen veranderen. In het niet-stopband gedeelte is de AR-methode namelijk tamelijk betrouwbaar en MA-parameters zouden dit ook voor het stopband waar maken.

6 Evaluatie

6.1 Implementatie

De implementatie van een AR-schatter is op zich makkelijk. De tijd-complexiteit van de implementaties is niet direct te schatten. In het ongunstige geval wordt de least-square vergelijking $P = M \setminus Y$ in opgave 6 in kwadratische tijd opgelost. Door gebruik te maken van een één-laags neural netwerk kan met een echt kleine fout het zelfde in lineaire tijd (evenredig aan de lengte van het signaal) bereikt

worden. Het berekenen van het spectrum uit de parameters gebeurt ook in lineaire tijd (evenredig aan het aantal parameters). De parametrische methode kan dus in lineaire tijd worden geïmplementeerd.^x

Dit is beter dan de $n * \log(n)$ tijd van de fft. Maar door het beperken van de lengte van de fft bij Welsh is ook PSD in feite van lineaire complexiteit.

6.2 Energie

Een AR-parametrisch model kan de form van een (niet stopband) signaal heel goed beschrijven. Maar er is minimaal één MA-parameter nodig om ook informatie over de energie te bevatten. Hoe dit in PYULE geïmplementeerd is is ons niet duidelijk. En hoe wij dat kunnen implementeren is niet goed uit de literatuur te halen.

6.3 Nauwkeurigheid

Zoals boven op pagina 11 gezegd is de parametrische methode tenminste een evengoede schatter als de meest gangbare conventionele schatter, de Welsh-methode.

De fout is in beide gevallen even groot. Er is geen verschil te zien bij de pieken, de dalen en de overgangsgebieden. Voor kleine samples geldt bovendien dat de parametrische schatter PYULE veel betrouwbaarder is dan de Welsh-implementatie PSD. En PYULE heeft duidelijk minder variatie en is beter te lezen.

Voor stopbanden geldt dit niet. De overgangsgebieden kan PYULE met een hoog aantal parameters nog even beschrijven. De stopband zelf en de maximale demping zijn niet te zien. De overgangsgebieden gaan gewoon in elkaar over. Om stopband signalen te beschrijven is een puur autoregressieve parametrische methode zeker niet geschikt. De verwachting is dat zelfs een eenvoudige implementatie van een ARMA schatter dit nadeel kan overkomen.

Voor spectrale resolutie geldt dat de parametrische methode alleen bij een lage ruiscomponente beter is. Bij lage ruis kan de parametrische methode (althans onze eigen implementatie) twee pieken beter scheiden dan PSD, bij sterke ruis doet PSD het beter. In de praktijk zou dit betekenen dat je beide methoden toe moet passen en dan afwegen welke betrouwbaarder is.

6.4 Overdracht en Coherentie

De overdrachtsfunctie die met de AR-parametrische methode kan worden berekend is niet erg betrouwbaar. De meeste signalen zijn weliswaar pasband- en geen stopbandsignalen, enkele pieken en een ruisachtige achtergrond. De stopband-blindheid van deze methode speelt dan geen grote rol. Overdrachtsfuncties echter worden degelijk door hun stopband-kwaliteiten gekarakteriseerd. Een ontbrekend of een fout gelokaliseerde stopband is soms niet acceptabel.

Maar een ARMA-parametrische methode zou dit kunnen veranderen. In het niet-stopband gedeelte is de AR-methode namelijk tamelijk betrouwbaar en MA-parameters zouden dit ook voor het stopband waar maken.

6.5 Mogelijkheden voor verdere analyse

De analyse in dit verslag is beknopt. Deze opdracht heeft de omvang van maar één studiepunt en we moesten onze aandacht op de algemene punten beperken. De punten die ontbreken maar die je graag nog verder wilt onderzoeken zijn:

- de prestaties van een algoritme met lineaire complexiteit,
- mogelijkheden om ook de MA-parameters te schatten,
- prestaties van een ARMA schatter bij het schatten van de overdrachtsfunctie,
- prestaties van een AR-gebaseerde coherentie, die via de kruiscorrelatie is berekend,
- prestaties van een ARMA-gebaseerde coherentie,
- toepassing op b.v. de data van hartslagsignaal.

Referenties

- [1] Techniques for Studying Short-Term Changes
in Cardio-Respiratory data, II, from
Computer Analysis of Cardiovascular Signals
M.Die Rienzo (Ed.)
IOS Press, 1995
- [2] Parametric Methods for Rational Spectra, from
Nonparametric Methods,
Chapter 3