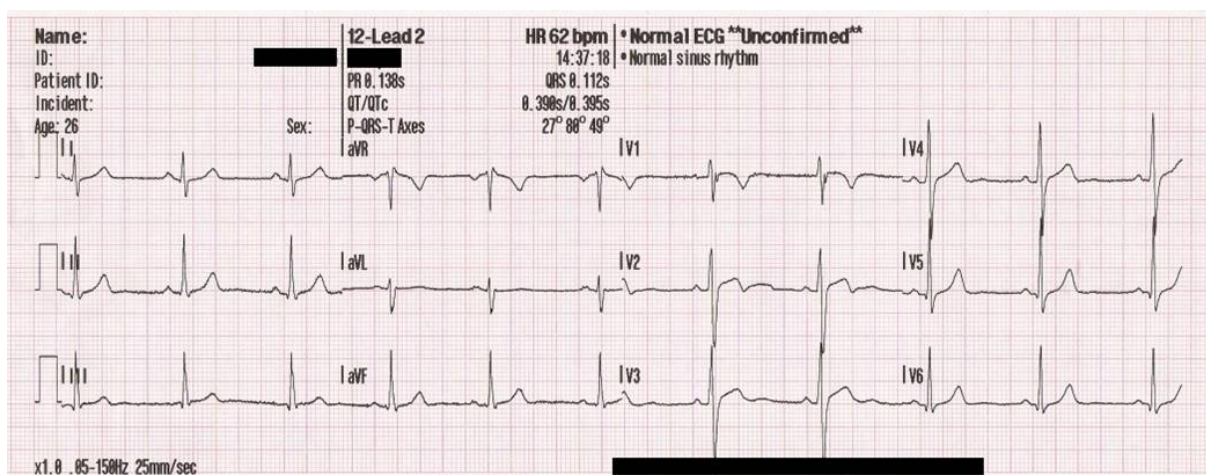


E3DSB miniprojekt 4 - Digital signalbehandling af elektrokardiogrammer

Janus Bo Andersen ¹

11. december 2019



¹ja67494@post.au.dk

Indhold

1	Indledning	1
1.1	Valgt emne	1
1.2	Formål, metode og struktur	1
1.3	Datakilder	2
1.4	Software, data og kildekode	2
2	Analyse	2
2.1	EKG-måling	2
2.2	EKG-signalet	3
2.2.1	Hjerterytme, frekvenser og filtre	3
2.3	Indikationer på hjerteproblemer og diagnostiske analyser	4
2.3.1	Arytmi	4
2.3.2	Analyse 1: Powerspektrum	4
2.3.3	Analyse 2: Heart-rate variability	5
2.4	Signalbehandling på EKG-data	5
3	Design	6
3.1	Filterdesign og pre-processering	6
3.2	Test af filter og pre-processering	7
4	Implementering og test	9
4.1	Analyse 1: Powerspektrum	9
4.2	Analyse 2: Heart-rate variability	11
5	Konklusion	14
6	Implementerede hjælpefunktioner	15
6.1	ecg_noise_filter	15
6.2	show_filter_response	15
6.3	setlatexstuff	16
6.4	ecg_load	16
6.5	ecg_filtered_cropped_moved	17
6.6	ecg_show_processing	17
6.7	ecg_powerspectrum	18
6.8	ecg_plot4_powerspectrum	18
6.9	ecg_find_r_ibi	19
6.10	ecg_plot4_peaks	19
6.11	ecg_plot4_tachogram	20
6.12	ecg_ibi_resample	21
6.13	ecg_show_resampling	21
6.14	ecg_hrv_powerspectrum	22
6.15	ecg_plot4_hrv	22
6.16	smoothMag (KPL)	23

1. Indledning

Fjerde miniprojekt i E3DSB er frit og udarbejdet individuelt med teori og metoder fra hele kurset. Ud over E3DSB er der bl.a. fundet inspiration til projektet i ST2PRJ2 fra ASE [1].

1.1 Valgt emne

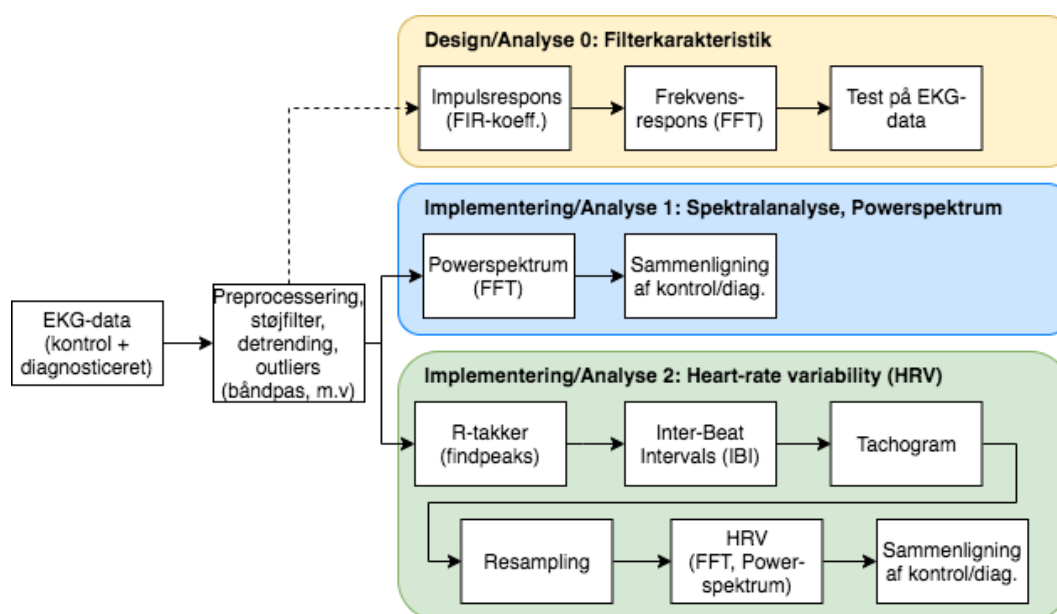
Elektrokardiografi er medicinsk signalbehandling til at måle et hjertes elektriske aktivitet. Signalet behandles først analogt og så digitalt for at danne et elektrokardiogram (EKG). EKG'er bruges til at screene, diagnosticere og monitorere hjertepatienter, og fx til råd og vejledning i hjertestartere.

EKG'er er interessante, fordi analysemetoderne fra E3DSB er velegnede. Der er meget information i både tids- og frekvensdomænet for et EKG-signal. Der er desuden støj og andre signalkomponenter, der skal filtreres bort, samt udfordrende tekniske problemstillinger.

1.2 Formål, metode og struktur

Projektet demonstrerer, hvordan værktøjer fra E3DSB i MATLAB kan anvendes på EKG-signaler. Dvs. hvordan information udtrækkes fra EKG-signalet i både tids- og frekvensdomænet. Metoden er at sammenligne resultater fra raske (kontrolgruppe) og diagnosticerede patienter (ekspertvurdering). Fokus er på arytmier.

Analyse redegør kort for EKG-signalet og anvendelser. **Design** udarbejder og karakteriserer filtre. **Implementering og test** demonstrerer analyser på EKG-signaler fra raske og sammenligner med diagnosticerede patienter. En del kode er implementeret i hjælpefunktioner, som ses til sidst i **Implementerede hjælpefunktioner**. Følgende figur opsummerer flow for signalbehandlingen.



Figur 1.1: Design og implementering

1.3 Datakilder

PhysioNet (<https://physionet.org/>) er et arkiv med medicinske, fysiologiske og biologiske signaler. Der findes også gode værktøjer, og bl.a. ATM (<https://archive.physionet.org/cgi-bin/atm/ATM>) er nyttig til at finde og downloade data. Projektet benytter derfra serier fra to datasæt. Det simulerede signal er benyttet til eksperimenter og algoritmeudvikling ej inkluderet i rapporten.

Kilde	Navn	Beskrivelse	Benyttede datasæt
[2]	PTB Diagnostic ECG Database	Klinisk data fra 268 patienter i forskellige diagnostiske klasser	Pt.104, 116 (kontrol); Pt.113 (atrieflim.); Pt.126 (palpitation)
[3]	PhysioNet Challenge 2011	Simuleret EKG-data til udvikling af computerbaseret diagnostik	Sim. #2: Normal 80 BPM. Kun til udvikling.

Tabel 1.1: Datasæt benyttet i projektet

1.4 Software, data og kildekode

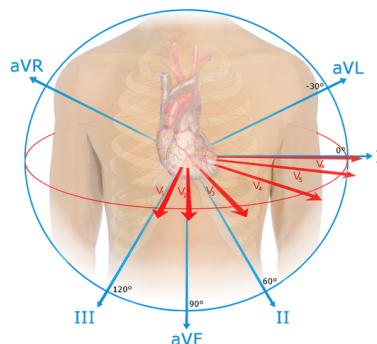
Rapport, databehandling og grafer er udarbejdet i MATLAB R2018b update 5. T_EX-kode er dannet fra MATLAB-kode vha. en tilpasset XSL-template. LuaL^AT_EX er benyttet til at compile en PDF. MATLAB-kode, data og template findes på <https://github.com/janusboandersen/E3DSB>.

2. Analyse

Først opridses (kort) baggrund for EKG-signalet. Så gennemgås et par indikationer og analysemetoder på relevante kardiovaskulære lidelser. Endelig diskuteres problemer ifm. signalbehandling på EKG-data.

2.1 EKG-måling

EKG'et tages med 10 elektroder på patientens krop: 4 placeres på arme og ben (ekstremiteter), 6 på brystkassen (thorax) [4]. Elektrisk potentiale måles mellem 12 forskellige kombinationer af elektroderne. Hver kombination kaldes en **afledning**.

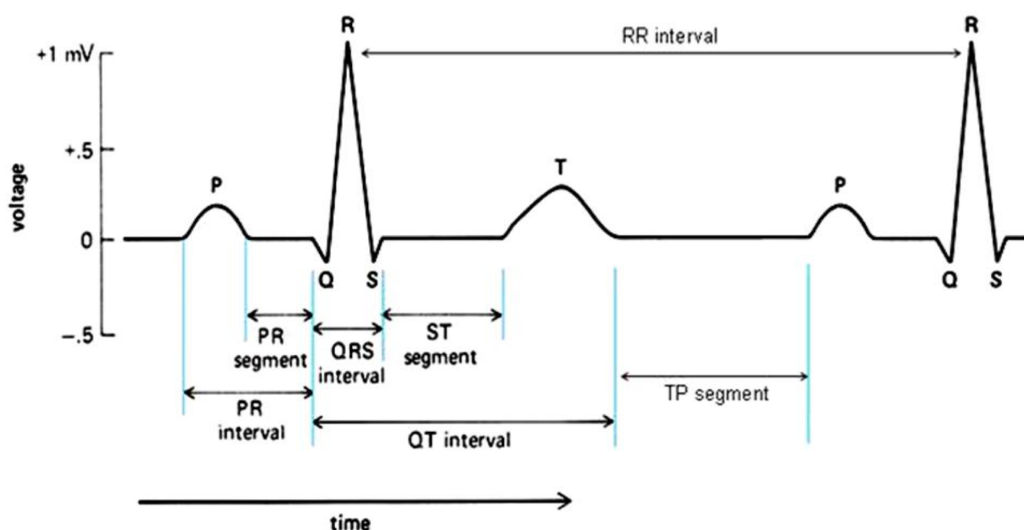


Figur 2.1: De 12 afledninger fra et EKG

En afledning er et vektorsignal, som viser hjertets aktivitet i bestemt retning/vinkel. Ovenstående figur illustrerer de 12 afledninger [5]. I dette projekt benyttes kun **afledning-II**. II giver det største signal, da måleretningen vektorielt er alignet med den længste akse i hjertet. Dette er også udbredelsesretning for depolariseringsimpulsen, der giver hjertemuskulaturens sammentrækning, samt for den følgende repolariseringsimpuls [1, ca. 13m36s].

2.2 EKG-signalet

Den elektriske impuls, der starter en hjertecyklus, udgår fra sinusknuden i højre forkammer (atrium). Den normale hjerterytme kaldes derfor en sinusrytme. En periode af signalet repræsenterer en hjertecyklus. EKG'et viser flere “takker” og “intervaller”, som har fysiologisk relevans. Afstanden mellem flere følgende perioder afgør hjertets rytme. Dette er illustreret i følgende figur [6].



Figur 2.2: EKG-signalets “takker” og “intervaller”

Hjertets sammentrækning (systole) begynder ved P-takken, hvor forkamrene (atrierne) trækker sig sammen. Hjerterkamrene (ventriklerne) trækker sig derefter sammen ved QRS-komplekset. Her pumpes blod ud i kroppen. R-takken er det højeste positive udslag i denne sammentrækning (for afledning-II). T-takken viser ventriklernes repolarisering og start på hjertets afslapning og udvidelse (diastole) [4]. RR-intervallet måler afstanden i tid mellem to R-takker og kan bruges til at måle pulsen. De flade linjesegmenter mellem takkerne kaldes baseline. “Afstande” og “højder” for forskellige komponenter har relevans i forskellige kliniske analyser.

2.2.1 Hjerterytme, frekvenser og filtre

EKG'er tages i hviletilstand, hvor pulsen ligger fra 50 min^{-1} til 100 min^{-1} [7]. Alle signalkomponenter inden for en cyklus ligger derfor over 0.8 Hz (50 min^{-1} omregnet). Forventeligt ligger hvilepuls omkring 60 min^{-1} , dvs. første komponent ved 1 Hz . Harmoniske heraf vil også optræde. Sygdomme kan tilføje mere højfrekvent flimren i signalet (se beskrivelse næste afsnit). Klinisk udstyr bør kunne måle frekvenser i intervallet fra 0.5 Hz til 150 Hz for at opnå “full fidelity” [8]. Til dette projekt er der ikke behov for frekvenser over 40 Hz .

Sygdom kan give meget lav puls (bradykardi) eller meget høj puls (takykardi). Pulsen kan da ligge fra

30 min^{-1} til 300 min^{-1} . Det ligger dog udenfor analyserne i projektet.

Filtre kan nemt ødelægge information, der har klinisk relevans. Så principielt bør alle filtre have lineær fase af diagnostisk grund (konstant group delay, ingen faseforvrænging).

Analyseformål og situation bestemmer de relevante frekvensbånd til analyse og støjfiltrering. EKG-maskiner tilbyder forskellige filtre og indstillinger (monitorering, diagnostik, 50/60 Hz, osv.).

2.3 Indikationer på hjerteproblemer og diagnostiske analyser

Et EKG giver indikation på en række hjerteproblemer. Her omtales kort hjerteproblemer relateret til arytmi (uregelmæssig rytme), som kan detekteres i et EKG-signal. Afsnittet opridser to anvendte analyser med baggrund i metoder fra E3DSB.

2.3.1 Arytmi

Arytmi er uregelmæssig hjerterytm. Det er normalt, at puls varierer med åndedræt (og fysiologiske processer). Men det er unormalt, når pulsen varierer *uregelmæssigt, for meget* eller *for lidt*. Årsagerne til arytmi er mange, fx stress eller hjerteproblemer. Konsekvenser varierer fra minimale til livstruende.

Palpitationer

En type arytmi er hjertebanken (palpitationer), der kan føles som om at hjertet springer slag over, eller slår for hurtigt. I tidsdomænet er en indikation på arytmi, at det længste RR-interval er minimum 0.16 s længere end det korteste [9][10]. I en spektralanalyse vil arytmi give en bredere fordeling af energi i spektret, end tilfældet for en normal hjerterytm.

Atrieflimren

Atrieflimren (AF) (forkammerflimren, eng.: atrial fibrillation) er hyppigt, især hos ældre. AF skyldes forstyrrelser i hjertets ledningsnet, og kan være følgelidelse til andre sygdomme. Impulsen fra sinusknuden forlader ikke atriet, men cirkulerer rundt som en "impulskarrusel" [7]. Så atriemuskulaturen laver mange sammentrækninger per pulsslag. Atriefrekvensen er kraftigt forhøjet og uregelmæssig på mellem 350 min^{-1} og 600 min^{-1} [11]. Pulsen svinger uregelmæssig mellem 50 min^{-1} og 100 min^{-1} [12].

Højfrekvente fluktuationer (flimrelinje) på baseline på EKG'et er et tegn [12]. Som ved palpitationer vil AF vise en bredere fordeling af energi i spektrum. Det forventes, at der er mere energi ved højere frekvenser end er tilfældet ved palpitationer.

2.3.2 Analyse 1: Powerspektrum

Første analyse viser, hvordan energien i signalet fordeler sig som en funktion af frekvens. Det relaterer til indikationer på arytmier fra forrige afsnit.

Power-spektrum: Power-spektrum beregnes via FFT'en. Iflg. Parsevals sætning kan energi og effekt i tidsdomænet regnes fra Fourier-koefficienter i frekvensdomænet. Gennemsnitseffekt for alle N bins er:

$$P = \frac{1}{N} \frac{2}{N} \sum_{k=0}^{N/2} X(k) \cdot X^*(k)$$

Her konjugeres frem for at kvadrere og tage modulus. For at skalere FFT'en divideres med N . For at regne effekt (ligesom $P = \frac{\Delta E}{\Delta t}$) divideres med N igen. Der skaleres med 2 for at gå fra dobbeltsidet til enkeltsidet analyse (summationsgrænse $N/2$).

2.3.3 Analyse 2: Heart-rate variability

Heart-rate variability (HRV) giver et mål på pulsens variabilitet. I litteraturen beskrives, at HRV bruges som risikomarkør for kardiovaskulære sygdomme og for nervesystemets funktion [13]. HRV-analyser regnes klinisk fra EKG'et [14]. Fitness-entusiaster kan se HRV på nyere pulsure og fitness-trackers. HRV kan anskues både i tidsdomænet (statistisk) og frekvensdomænet. Projektet fokuserer på frekvensdomænet.

HRV beregnes via spektralanalyse på længden af RR-intervallerne = IBI (inter-beat interval). Visning af udviklingen af IBI over tid kaldes et tachogram. Med et Powerspektrum på IBI (normaliseret) vises, hvor energien (variansen) er.

Resampling og detrending: Pointen er at analysere den uregelmæssige rytme, men metodisk er det en udfordring, at IBI-signalet er samlet uregelmæssigt (en "sample" per pulsslæg). MATLAB-funktionen `resample` bruges til at resample og interpolere IBI-signalet til et regelmæssigt samlet signal med en højere samplingsfrekvens [15]. Der vælges en ny samplingsfrekvens på 10 gange forventet pulsfrekvens. Resampling stemmer overens med metodeanbefalinger [13]. IBI normaliseres til middelværdi på 0 s.

Litteraturen nævner mange mulige diagnostiske mål. Et eksempel på et diagnostisk mål, der går igen flere steder, er **LF/HF**-ratioen for relativ energi i 2 frekvensbånd, hvor **LF**-båndet er 0.04 Hz til 0.15 Hz, og **HF** er 0.15 Hz til 0.4 Hz.

2.4 Signalbehandling på EKG-data

Påvirkning fra støj under målingerne giver uønsket påvirkning af EKG'et og det spektrale indhold. Fysiologiske signaler har "af natur" variabilitet; dog er kun noget af variabiliteten brugbar.

Støjklender og støjfilter

Der forventes en række støjartefakter ved opmåling af EKG:

- Lavfrekvent støj fra patientens bevægelser og vejrtrækning ($< 50 \text{ min}^{-1} = 0.8 \text{ Hz}$).
- DC-støj fra kontaktpotentiale ved elektrode/hud-forbindelsen (op til 200-300 mV). Forværres af bevægelser og perspiration. [8].
- Baseline vandrer (driver) sfa. ovenstående el.lign. (ukendt frekvens).
- Muskelstøj i frekvensområdet 5-50 Hz. Svært at filtrere [8].
- AC-støj fra elforsyning (50 Hz i europæisk data og 60 Hz i amer.) samt harmoniske.
- Andre instrumenter og apparater i lokalet, fx mobiltelefoner.

Der burde være common-mode-rejection via A/D-converteren, men elektroder benyttet til opmåling er muligvis ikke ens, hvilket reducerer CMRR.

I projektet ignoreres frekvenser over 40 Hz, da der ikke er relevante fysiologiske komponenter. Det tillader en simplificering af filteret til en lavere orden med bredere transitionsområde. Dermed undgås også et båndstop-filter (notch), der skal flyttes alt efter datasættets oprindelsesland.

Alt i alt: Støjfilteret skal fungere som et **båndpasfilter**, der lukker frekvenser fra 0.8 Hz op til 45 Hz igennem (regnet så der er 5 Hz transition op til powerline-støj, og 5 Hz ned til relevante frekvenser). På den måde kan ovennævnte støjkloder frasorteres, og “interessante” frekvenser nævnt i tidligere afsnit stadig passere.

Periodicitet og stationaritet

DFT'en (FFT'en) er non-parametrisk, og forudsætter at signalet er én periode af et periodisk signal. EKG-signalet er en *ikke*-stationær proces og er ikke periodisk. Fordelingsegenskaber er ikke stabile over tid, og amplitude, frekvens og fase vil ikke være konstante over tid.

Ikke-periodicitet håndteres med **vinduesfunktioner** i filterdesign og FFT'er for at få bedre kontrol over leakage og ripples.

Ikke-stationaritet gør fortolkning sværere, når en fysiologisk komponent optræder i flere bins pga. variation over tid. I dette projekt gøres ikke noget for at håndtere dette. En mulig metode er et glidende vindue, hvor signalsektioner analyseres separat. Short-Time Fourier Transform (STFT) er et bud.

3. Design

3.1 Filterdesign og pre-processering

Filterbehov blev diskuteret tidligere. Her designes og karakteriseres filter og pre-processering.

Støjfilteret er FIR-typen og har cut-off-frekvenser ved hhv. 0.8 Hz og 45 Hz. Filteret skal have gain på 1 V/V i pasbåndet. Der benyttes et Hann-vindue for at få hurtig transition og smal main lobe. “Prisen” er, at sidelobe-dæmpning “kun” er omkring -50 dB for de første par bins.

For at få et hurtigt roll-off, og dæmpning på 6 dB ved 0.8 Hz, skal filterorden for et højpasfilter være omkring 2000. Så båndpas er designet som 2000.-ordens højpas foldet med et 200.-ordens lavpas.

Modsat et realtidssystem *behøver* filtrene i projektet ikke at være kausale. FIR-filtre designet efter metoder fra E3DSB er dog kausale. For at undgå høj filterorden kunne “forward-backward-filtering” (IIR) give samme eller bedre respons med lavere filterorden og med zero-phase. Et alternativ kunne være at fjerne lavfrekvente støjkomponenter med cubic-splines i tidsdomænet i stedet for at filtrere. Begge dele dog udenfor projektets omfang.

Filteret fjerner gennemsnitligt DC-niv., men baseline er på medianen. Så efter filtrering justeres

DC-niveauet yderligere så baseline rykkes til 0 mV.

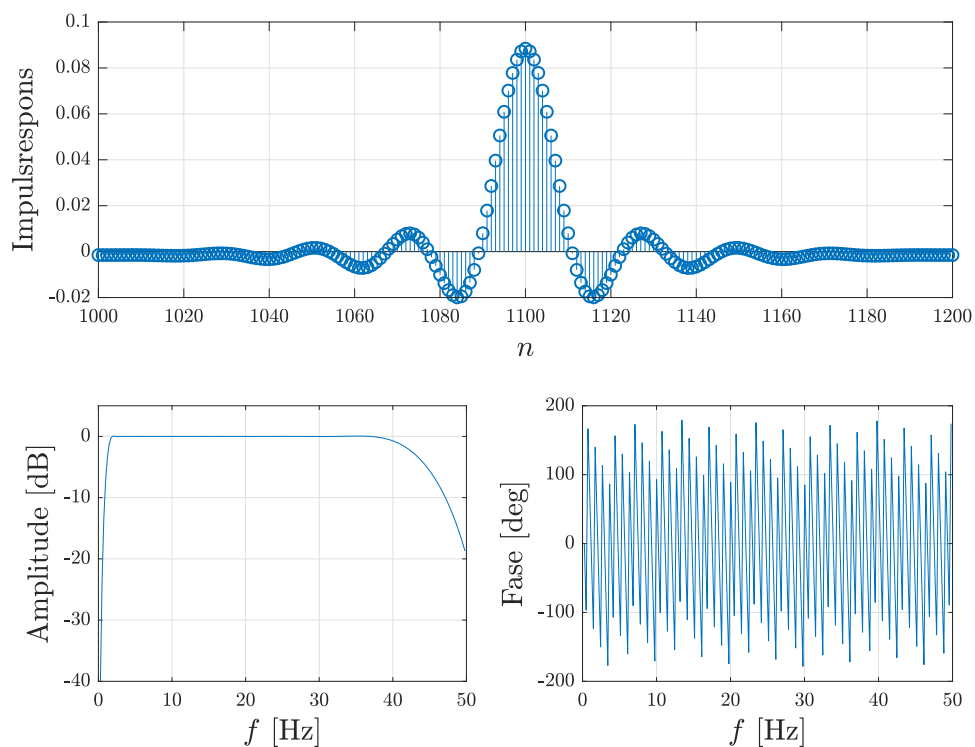
Endelig fjernes outliers, her primært indsvingning (transiente) indtil delay-line er fyldt op.

```
1 clc; clear all; close all;
```

Specificér og byg filteret og se impuls- og frekvensrespons. Alle dataserier har 1000 Hz samplingsfrekvens.

```
1 Fs = 1000; % Hz ord. Fc ord. Fc
2 h = ecg_noise_filter(Fs, [2000 0.8], [200 45]); % Som specificeret
3
4 % Respons i 4096-punkts FFT, 0-50 Hz på frek. akse, impulsresp. 1000-1200
5 show_filter_response(h, 4096, Fs, [0 50], [1000 1200]);
```

Filterkarakteristik



Filterets pasbånd er som ønsket, og der er lineær fase. Impulsrespons viser, at efter ca. 1200 samples vil indsvingning af filter næsten være færdig (resterende koefficienter “tæt på” 0).

3.2 Test af filter og pre-processering

Filteret testes på kontrolgruppen. Data loades og behandles, bl.a. justeres til rå værdier (data er lagret med gain 2000 V/V). Der oprettes også hjælpeserier i objektet (ecg er en struct).

```
1 A = 2000; B = 0;
2
3 % Patient 104: Kontrol (rask), mand 58, 60 s data, Fs=1000, A=2000, DC=0
4 ecg_ktr1 = ecg_load('data/kontrol/ktr1/s0306lrem.mat', 'Ktrl.1 (m/58)', ...
```

```

5      'II', 'mV', A, B, Fs);
6
7  % Patient 116: Kontrol (rask), mand 54, 60 s data, Fs=1000, A=2000, DC=0
8  ecg_ktr2 = ecg_load('data/kontrol/ktr2/s03021rem.mat', 'Ktrl.2 (m/54)', ...
9      'II', 'mV', A, B, Fs);
10
11 d = {ecg_ktr1, ecg_ktr2};           % Gem på en smart måde til senere

```

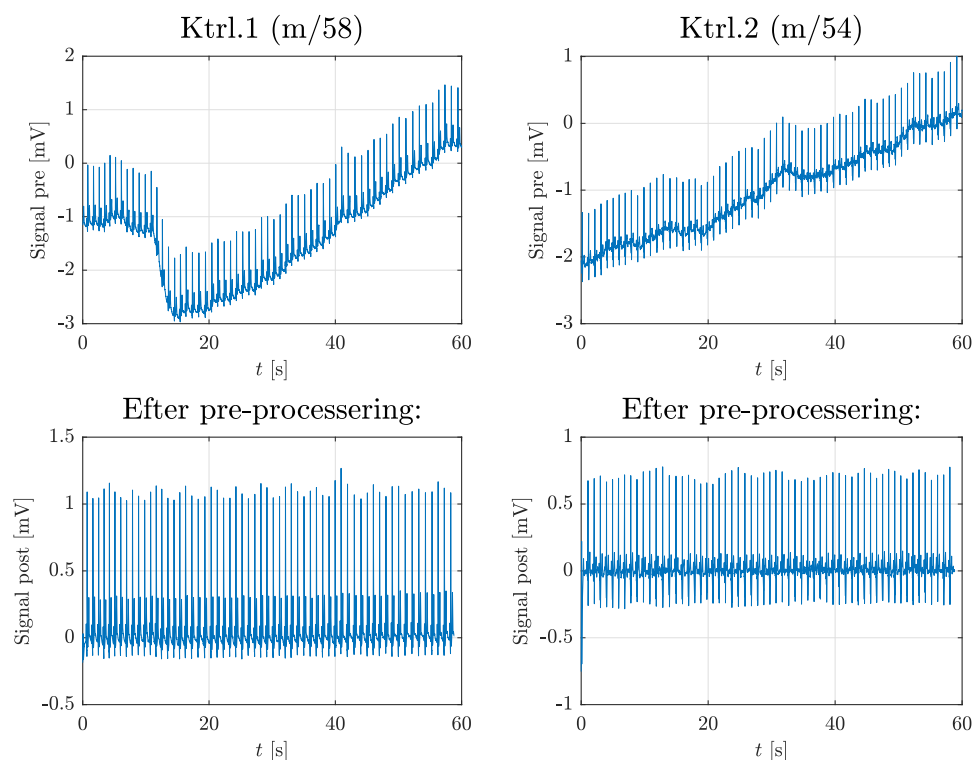
Pre-processér de to dataserier: Filtrér med støjfilter. Fjern indsvingning (fjernelse af første 1200 samples koster 1.2 sek. data). Justér baseline til 0 mV. Sammenlign før/efter-plots.

```

1  crop_samples = 1200;               % de første 1200 samples fjernes (indsv.)
2
3  for e=1:2
4      d{e} = ecg_filtered_cropped_moved(d{e}, h, crop_samples);
5  end
6  ecg_show_processing( d(1:2) );     % Sammenlign før/efter

```

Effekt af pre-processering



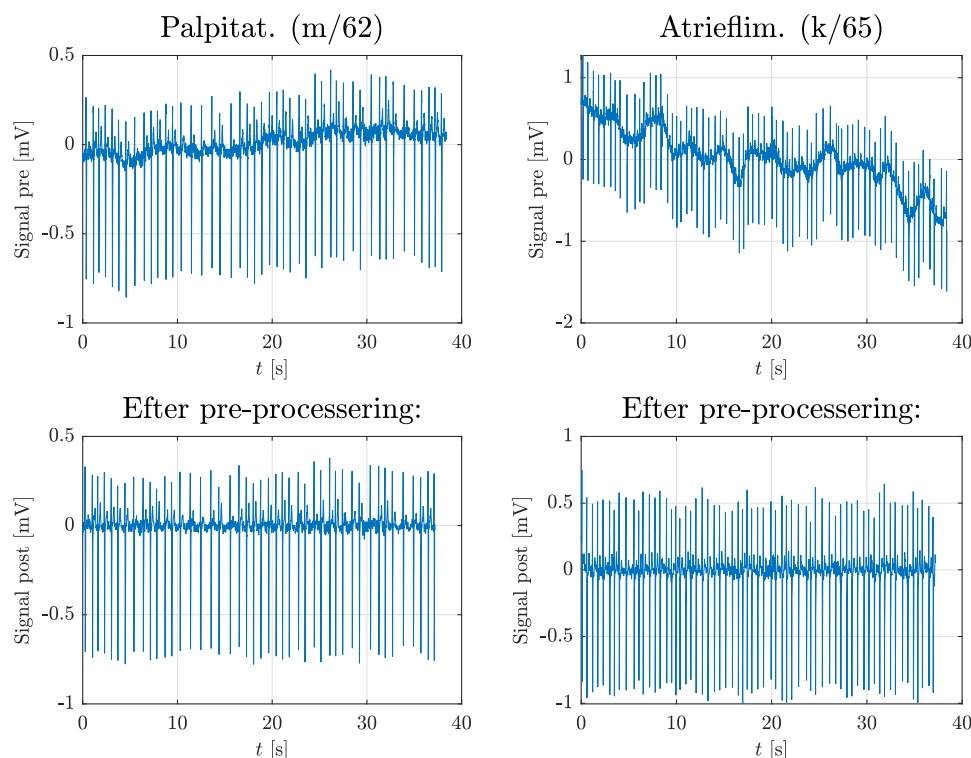
Pre-processering virker som ønsket. Lavfrekvent støj og vandring på baseline fjernes. Signalets baseline ligger på 0 mV efter justeringer.

4. Implementering og test

I dette afsnit implementeres Analyse 1 og Analyse 2. Data for diagnosticerede patienter loades og pre-processeres først.

```
1 % Pt. 126: Sinusarytmi: Hjertebanken, mand 62, 38 s, Fs=1000, A=2000, DC=0
2 d{3} = ecg_load('data/arytmi/sa/s0154_rem.mat', 'Palpitat. (m/62)', ...
3         'II', 'mV', A, B, Fs);
4
5 % Pt. 113: Arytmi: Atrieflimren, kvinde 65, 38 s, Fs=1000, A=2000, DC=0
6 d{4} = ecg_load('data/arytmi/af/s0018crem.mat', 'Atrieflim. (k/65)', ...
7         'II', 'mV', A, B, Fs);
8
9 for e = 3:4      % Pre-processer signaler for de to diagnosticerede pat.
10     d{e} = ecg_filtered_cropped_moved(d{e}, h, crop_samples);
11 end
12 ecg_show_processing( d(3:4) );           % Sammenlign før/efter
```

Effekt af pre-processering



Igen bekræftes grafisk, at filtrering og anden pre-processeringen er gået OK.

4.1 Analyse 1: Powerspektrum

Analyse 1 udføres ved at regne powerspektra på de 4 serier, som beskrevet i analyseafsnittet. FFT'en regnes med et Hann-vindue. Med smoothing fås et lidt pænere spektrum (tak for funktionen, KPL!).

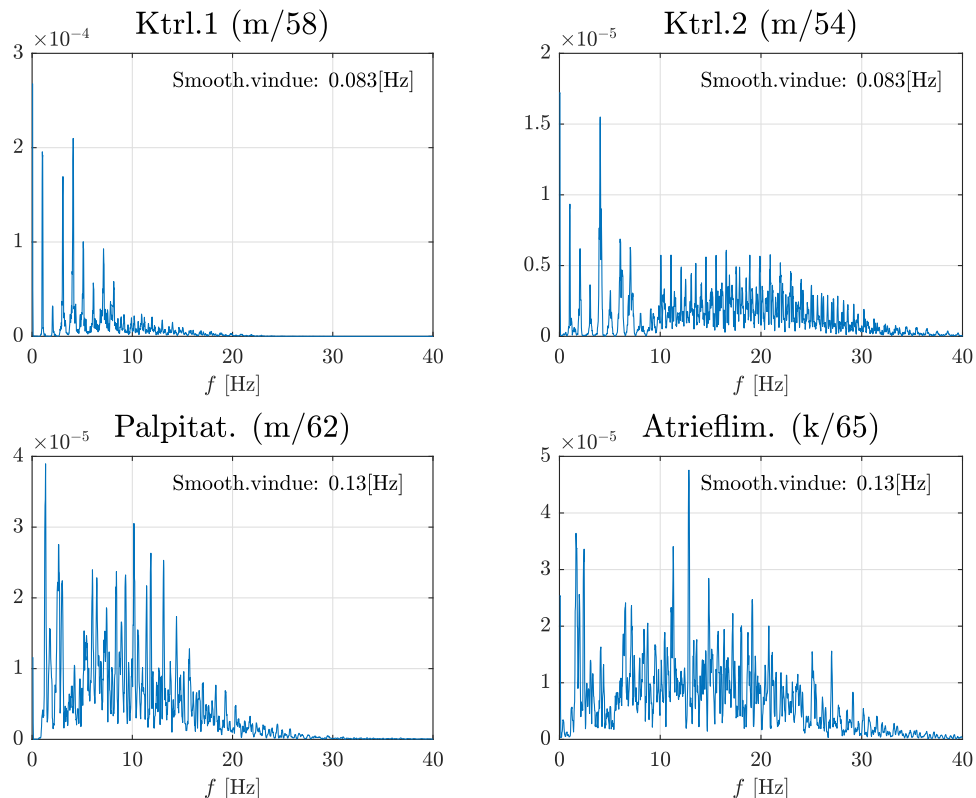
Endelig sammenlignes de resulterende spektra grafisk.

```

1 MA = 5; % Smoothing af Powerspektrum glidende over MA bins
2 for e = 1:4 % Udfør for alle 4 serier
3     d{e} = ecg_powerspectrum(d{e}, MA);
4 end
5 ecg_plot4_powerspectrum(d, [0 40]); % Vis Powerspectrum fra 0-40 Hz

```

Powerspektra [$\propto V^2 \text{Hz}^{-1}$]



Den først spike i spektret repræsenterer patientens puls (for de raske, i hvert fald). For Ktrl.1 ligger den omkring 1.05 Hz. Det svarer til en normal hvilepuls på 63 min^{-1} . Ktrl.2 har puls omkring 61 min^{-1} .

Det ses, at spektra for kontrolgruppe og diagnoser er væsentligt forskellige. For raske patienter ligger det meste af energien under 10 Hz i et pænt mønster af puls og harmoniske. For Ktrl.2 ses der dog antageligt noget muskelstøj i området 10-30 Hz.

Spektra for palpitationer og atrieflimmer svarer til forventningen fra teori afsnittet om arytmier: Der er *væsentligt* bredere fordeling af energi i spektra for patienter med diagnose end for kontrolgruppen. Der er intet pænt mønster af puls og harmoniske. Atrieflimren har, som forventet, den bredeste fordeling.

Analysen demonstrerer, hvordan powerspektra kan bruges til give indikationer på hjertelidelser. Spektra viser også, at der ikke er meget energi i båndet 30-40 Hz.

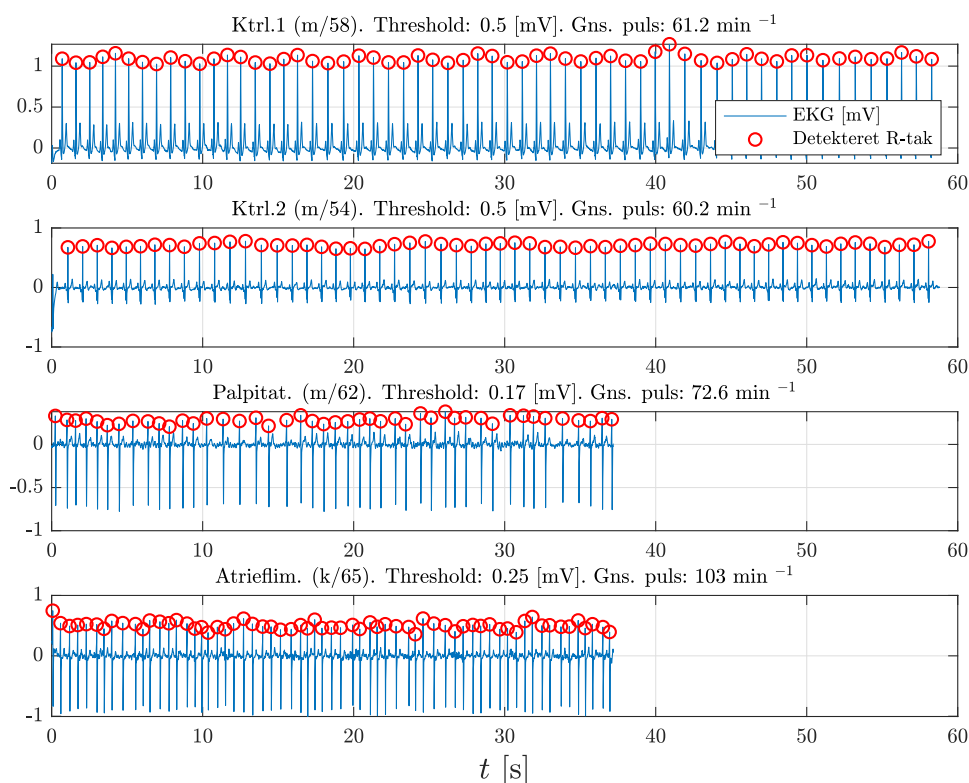
4.2 Analyse 2: Heart-rate variability

HRV-analyse kræver i princippet mere end 4 minutters observationer for at kunne give valide data på relevante dele af spektre [16]. EKG-signalerne valgt til dette projekt (mellem 38-60 s lange) er principielt ikke lange nok til at give håndfaste estimater. Men det er ok til at demonstrere analysen.

Med det “in mente” udføres Analyse 2 med baggrund i teoriafsnittet. Først detekteres R-takkerne, og resultatet inspiceres grafisk. For at detektere R-takke algoritmisk, sættes der en tærskelværdi manuelt. Den er sat ved inspektion af tidsserierne (ca. hvilket niveau i signalet overstiges kun af R-takke). Der kunne også findes automatiske metoder. MATLAB’s `findpeaks` benyttes til at lokalisere peaks. Kriteriet er: En peak er en sample med højere værdi end sine to naboer og over tærskelværdien. RR-intervallerne og IBI (normaliseret, i sek.) beregnes derefter.

```
1 thresholds = [0.5, 0.5, 0.17, 0.25]; % Tærskelværdier per inspektion, mV
2
3 for e = 1:4 % R-takke og IBI-intervaller for alle 4 serier
4     d{e} = ecg_find_r_ibi(d{e}, thresholds(e));
5 end
6 ecg_plot4_peaks(d, [0 60]); % Vis R-takke for serierne, 0-60s
```

Detektion af R-takke i EKG-signaler i [mV]

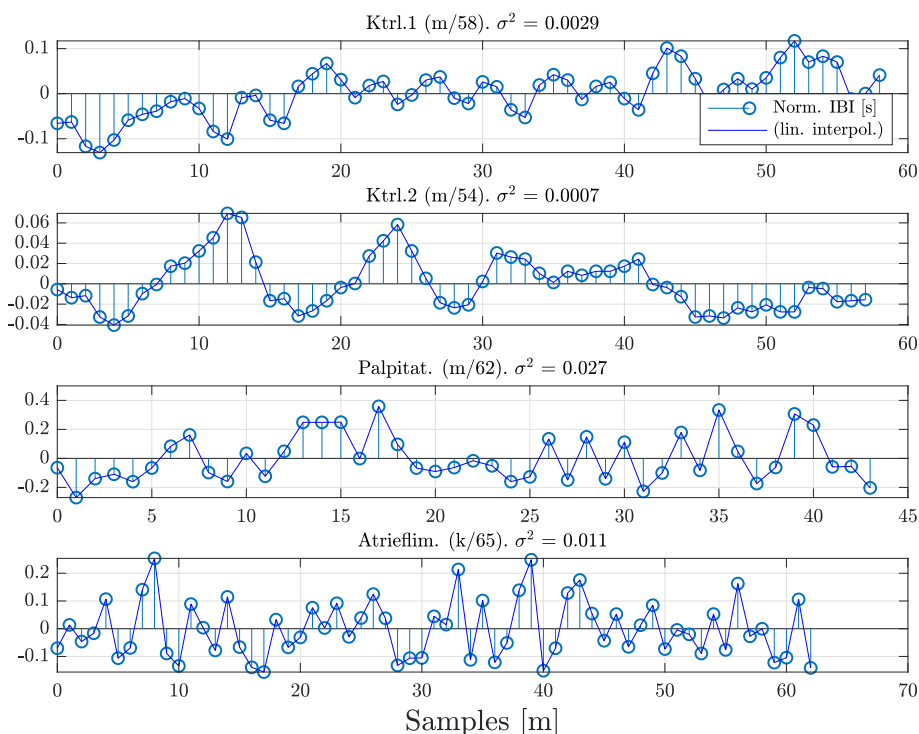


Diagrammerne bekræfter, at algoritmen har detekteret alle R-takkerne ud fra de givne tærskelværdier. Der vises også beregnet puls over hele serien, som svarer fint til beregninger baseret på frekvenser i Powerspektrum. Hjerterytmen for diagnosticerede patienter virker, helt forventeligt, ud fra diagrammet mere variabel og “kaotisk” end for de raske kontroller. Bemærk, at patienten med atrieflimren har en forhøjet puls.

Tachogrammerne vises for at illustrere udsving ift. det gennemsnitlige inter-beat interval.

```
1 ecg_plot4_tachogram(d);
```

Tachogram viser normaliseret IBI i [s] per R-tak sample



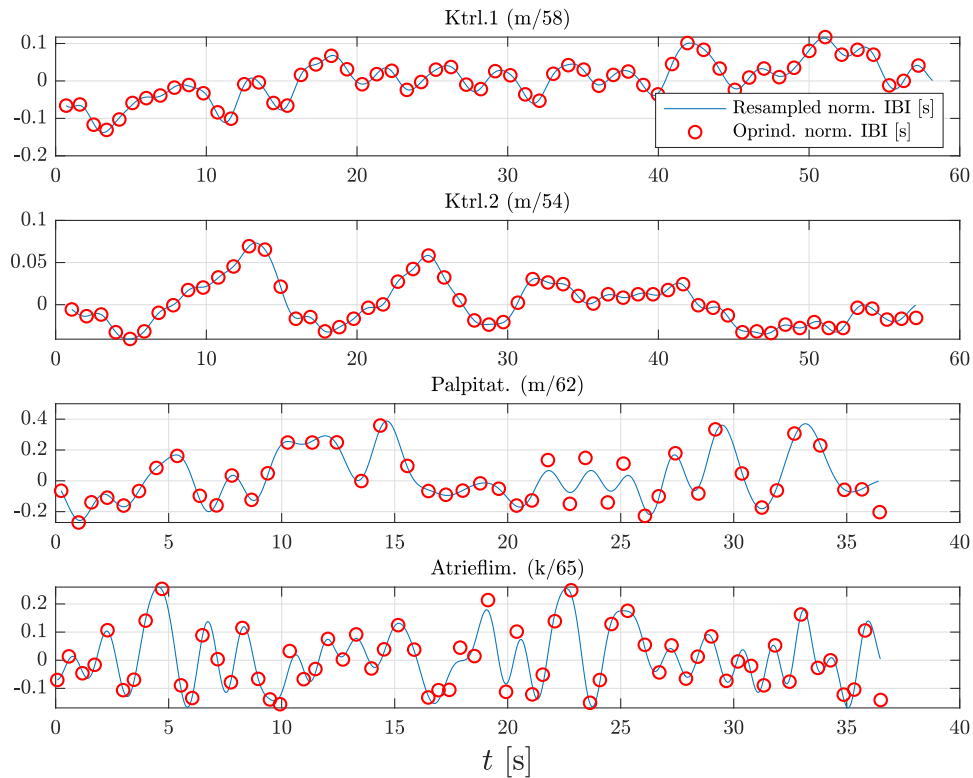
Tachogrammerne viser ret tydeligt, at der er forskel på variabilitet i hjerterytmen hos raske versus diagnosticerede, især for patienten diagnosticeret med atrieflimren. Sammenlign fx mindste og højeste værdi for IBI - i serien for palpitationer, er der mere end 0.6s i forskel mellem korteste og længste interval. For de raske patienter ses også variabilitet. Her er det overvejende mest sandsynligt, at størstedelen er drevet af vejrtrækningen, hvilket er helt normalt. Hjerterytmen har høj korrelation med respirationen, hvilket kaldes respiratorisk sinusarytmi (RSA). Variansen (σ^2) for hver serie er angivet i diagrammet. Igen bekræftes, at de diagnosticerede har højest varians i hjerterytmen.

Fra tachogrammet “føles” det nærliggende at lave en spektralanalyse. Det er tydeligt fra figurer, at samplingen har været uregelmæssig. Først skal IBI-serien resamples [15].

Afstanden i tid mellem hver sample er givet fra `findpeaks`-algoritmen. Splines-metoden benyttes til at interpolere mellem samples, fordi standardmetoden slet ikke kan fange al variabiliteten i IBI-signalet.

```
1 %           Gns. puls for      K1 K2 SA AF
2 desiredFs = round((10/60)*[61 60 72 103]); % 10 x gennemsnitspuls i Hz
3                                     % som omtalt i teoriafsnit
4 for e = 1:4
5     d{e} = ecg_ibi_resample(d{e}, desiredFs(e)); % Resample hvert IBI
6 end
7 ecg_show_resampling(d); % Vis resultater af resampling
```

Resultat af resampling af normaliseret IBI, i [s]



Resultatet af resamplingen virker overordnet fornuftigt, bortset fra at den resamplede serie for palpitationer har lidt afvigelse omkring 21-25 s.

HRV-analysen regnes med de nye regelmæssigt samplede serier. Powerspectrum beregnes, og samtidig regnes LF/HF-ratioen. Der benyttes *ikke* et Hann-vindue i denne FFT. Powerspectrum udglattes til brug i grafer ved et glidende gennemsnit.

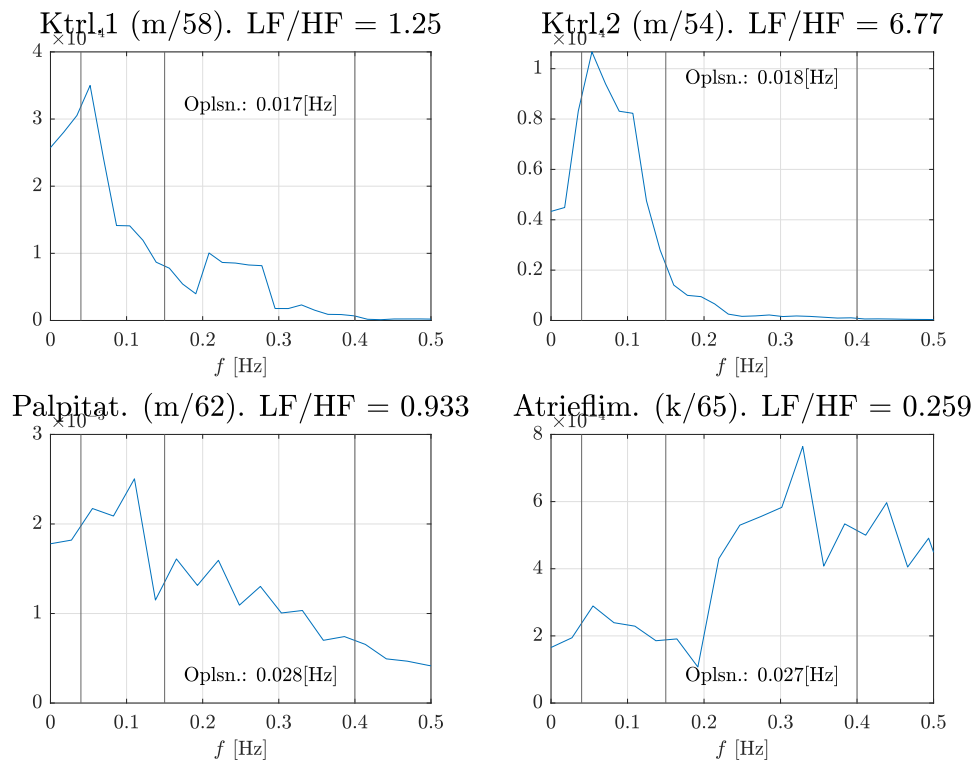
HRV-analysen er primært relevant i frekvensområden 0-0.5 Hz. Specifikt kigges på intervallerne LF: 0.04-0.15 Hz og HF: 0.15-0.4 Hz. Disse intervaller er markeret i diagrammet. Et udglattet powerspectrum plottes, og beregnede LF/HF-ratioer er vist for hver patient.

```

1 MA = 5; % Smoothing af Powerspektrum glidende over MA bins
2 for e = 1:4
3     d{e} = ecg_hrv_powerspectrum(d{e}, MA); % Beregn HRV for hver patient
4 end
5 ecg_plot4_hrv(d, [0 0.5]); % Vis HRV for f op til 0.5Hz

```

HRV Powerspektra [$\propto \text{s}^2\text{Hz}^{-1}$]



Som nævnt i indledningen til analysen lider resultatet under at dataserierne er relativt korte. Desuden er der mange forskellige mulige diagnostiske grænser for LF/HF. Ikke desto mindre er det interessant at observere, at begge kontroller har $\text{LF}/\text{HF} > 1$, mens de to diagnosticerede patienter har $\text{LF}/\text{HF} < 1$. Det fremgår således, at energien i nervesystemets impulser er mere højfrekvent for de to diagnosticerede patienter, relativt til kontrollerne.

Denne analyse har vist, hvordan Fourieranalyse kan benyttes til at implementere en nyere diagnostisk teknik. Længere dataserier ville give analysen mere kraft.

5. Konklusion

Denne rapport har demonstreret metoder fra E3DSB anvendt på EKG-signaler. Det er vist, hvordan information, som kan have diagnostisk relevans for sundhedspersonale, fx kardiologer, udtrækkes fra EKG-signalet i tids- og frekvensdomænet. Forbedringsforslag til metoder modtages meget gerne!

Projeket har også demonstreret, hvordan en lidt større datanalyse kan gribes an i MATLAB. Gode råd, tips og tricks modtages meget gerne. Det fremgår nok af rapporten, at det var nødvendigt at bruge en del tid på at oparbejde viden omkring fagområdet *og* bruge en del tid på at forbehandle data, før der egentlig kunne udarbejdes analyser.

6. Implementerede hjælpefunktioner

Der er til projektet implementeret en række hjælpefunktioner.

6.1 ecg_noise_filter

```
1 function [h] = ecg_noise_filter(Fs, highpass, lowpass)
2 % Lav et båndpasfilter. Specificér hp, lp koeff. som [orden Fc]
3 % Janus Bo Andersen, 2019
4     flag = 'scale'; % Normaliser koeff. til pasbånd på 0 dB
5     win_hp = hann(highpass(1)+1); % Vinduesfkt. af Hann-typen
6     win_lp = hann(lowpass(1)+1); % Ditto
7     hp = fir1(highpass(1), highpass(2)/(Fs/2), 'high', win_hp, flag);
8     lp = fir1(lowpass(1), lowpass(2)/(Fs/2), 'low', win_lp, flag);
9     h = conv(lp, hp); % Fold lavpas og højpas sammen
10 end
```

6.2 show_filter_response

```
1 function [] = show_filter_response(h, N, Fs, flim, hlim)
2 % Viser filterrespons i dB
3 % Janus Bo Andersen, 2019
4     setlatexstuff('Latex'); figure % Figurindstillinger
5     H = fft(h, N); % Frekvensrespons
6     k = floor( flim(2)*(N/Fs) ); % Højeste bin vi vil se
7
8     subplot(2,2,1:2) % Impulsrespons
9     stem(hlim(1):hlim(2), h( hlim(1)+1:hlim(2)+1 ))
10    xlabel('$n$', 'Interpreter','Latex', 'FontSize', 15);
11    ylabel('Impulsrespons', 'Interpreter','Latex', 'FontSize', 15);
12    grid on;
13
14    subplot(2,2,3) % Amplituderrespons
15    plot((1:k)*(Fs/N), mag2db(abs(H(1:k)))));
16    xlabel('$f$ [Hz]', 'Interpreter','Latex', 'FontSize', 15);
17    ylabel('Amplitude [dB]', 'Interpreter','Latex', 'FontSize', 15);
18    ylim([-40 5]);
19    grid on;
20
21    subplot(2,2,4) % Faserrespons
22    plot((1:k)*(Fs/N), angle(H(1:k))*180/pi );
23    xlabel('$f$ [Hz]', 'Interpreter','Latex', 'FontSize', 15);
```

```

24     ylabel('Fase [deg]', 'Interpreter','Latex', 'FontSize', 15);
25     grid on;
26
27     sgttitle('Filterkarakteristik', 'Interpreter', 'Latex', 'FontSize', 20);
28 end

```

6.3 setlatexstuff

```

1 function [] = setlatexstuff(intpr)
2 % Sæt indstillinger til LaTeX layout på figurer: 'Latex' eller 'none'
3 % Janus Bo Andersen, 2019
4     set(groot, 'defaultAxesTickLabelInterpreter',intpr);
5     set(groot, 'defaultLegendInterpreter',intpr);
6     set(groot, 'defaultTextInterpreter',intpr);
7 end

```

6.4 ecg_load

```

1 function [ecg] = ecg_load ( filename, name, lead, unit, gain, offset, Fs )
2 %load_ecg læser en fil og returnerer en datastruktur med ECG-data
3 %   filename: filnavn med (type .mat)
4 %   name: navn på dataserien (bruges i plots)
5 %   lead: afledning ('I', 'II', 'III', osv.)
6 %   unit: enhed ('mV')
7 %   gain: forstærkning i data (typisk 200-2000)
8 %   offset: 0 medmindre der er benyttet anden base end 0
9 %   Fs: samplingsfrekvens
10 % Janus Bo Andersen, 2019
11
12     ecg = load(filename, 'val');
13     ecg.meta.name = name;                % Dataseriens navn
14     ecg.meta.lead = lead;                % Benyttet afledning (I, II, ...)
15     ecg.meta.unit = unit;                % Værdienhed for ECG-måling
16     ecg.meta.gain = gain;                % Gain i data (.info-fil)
17     ecg.meta.offset = offset;            % Base i date (.info-fil)
18     ecg.Fs = Fs;                        % Samplingsfrekvens (.info-fil)
19
20     % Afledte beregninger
21     ecg.Ts = 1/ecg.Fs;                  % Sampleafst. på 1/Fs [s]
22     ecg.N = length(ecg.val);             % Antal samples
23     ecg.resolution = ecg.Fs / ecg.N;     % Frekvensopløsning [Hz/bin]
24     ecg.T = ecg.N * ecg.Ts;              % Samlet tid i sek
25     ecg.n = (0:ecg.N - 1);              % Vektor til samplenumre
26     ecg.t = ecg.n * ecg.Ts;              % Vektor til tidsakse

```

```

27
28 % Rådata (rå mV før justeringer i lagret datasæt)
29 ecg.raw = (ecg.val - ecg.meta.offset) ./ ecg.meta.gain;
30 end

```

6.5 ecg_filtered_cropped_moved

```

1 function [ecg] = ecg_filtered_cropped_moved(ecg, h, crop)
2 % Returnerer processeret ECG-objekt, samples der fjernes er 1:crop.
3 % Baseline flyttes fra median til 0 V.
4 % Janus Bo Andersen, 2019
5
6     ecg.filt.filtered = filter(h, 1, ecg.raw);           % Filtring
7
8     ecg.filt.filtered = ecg.filt.filtered(crop+1:end); % Cropping
9     ecg.filt.N = length(ecg.filt.filtered);
10    ecg.filt.T = ecg.filt.N * ecg.Ts;
11    ecg.filt.n = (0:ecg.filt.N - 1);
12    ecg.filt.t = ecg.filt.n * ecg.Ts;
13    ecg.h = h;                                           % Gem filteret
14
15    m = median(ecg.filt.filtered);                       % Flyt baseline
16    ecg.filt.filtered = ecg.filt.filtered - m;
17 end

```

6.6 ecg_show_processing

```

1 function [] = ecg_show_processing(d)
2 % Plotter 2x2 plot med før/efter pre-processering
3 % d indeholder 2 ECG-objekter
4 % Janus Bo Andersen, 2019
5     setlatexstuff('Latex'); figure                     % Figurindstillinger
6     for p = 1:2
7         subplot(2,2,p)                                % Ubehandlet EKG
8         plot(d{p}.t, d{p}.raw);
9         xlabel('$t$ [s]', 'Interpreter','Latex', 'FontSize', 10);
10        ylabel(['Signal pre', ' [' , d{p}.meta.unit, ' ]'], ...
11            'Interpreter','Latex', 'FontSize', 10);
12        grid on;
13        title(d{p}.meta.name, 'Interpreter','Latex', 'FontSize', 15)
14
15        subplot(2,2,p+2)                                % Processeret EKG
16        plot(d{p}.filt.t, d{p}.filt.filtered);
17        xlabel('$t$ [s]', 'Interpreter','Latex', 'FontSize', 10);

```

```

18     ylabel(['Signal post', ' [' , d{p}.meta.unit, ']' ], ...
19           'Interpreter','Latex', 'FontSize', 10);
20     grid on;
21     title('Efter pre-processering:', ...
22           'Interpreter','Latex', 'FontSize', 15)
23 end
24 sgtitle('Effekt af pre-processering', ...
25         'Interpreter', 'Latex', 'FontSize', 20);
26 end

```

6.7 ecg_powerspectrum

```

1 function [ecg] = ecg_powerspectrum(ecg, smooth_bins)
2 % Regner Powerspectrum på pre-processeret ECG-data
3 % Benytter Hann-vinduesfkt. og smoothing af Powerspektrum
4 % smooth_bins: Antal bins, i glidende gennemsnit (skal være ulige)
5 % Janus Bo Andersen, 2019
6
7     x = ecg.filt.filtered;
8     N = length(x);
9     w = hann(N);                                % Hann-vinduesfkt.
10    X = fft(x.*w');                              % Windowed FFT
11    P = 2*(X.*conj(X))/(N^2);                    % Power for hver frekv.sample
12    ecg.filt.Ps = smoothMag(P, smooth_bins);      % Glidende gennemsnit
13    ecg.filt.Ps_sw = ecg.resolution*smooth_bins; % Bredde af smoothing, Hz
14 end

```

6.8 ecg_plot4_powerspectrum

```

1 function [] = ecg_plot4_powerspectrum(d, flim)
2 % Plotter 2x2 figur med powerspektra, og sætter xlim = flim
3 % d indeholder 4 ECG-objekter
4 % Janus Bo Andersen, 2019
5     setlatextstuff('Latex'); figure              % Figurindstillinger
6
7     for p = 1:4
8         subplot(2,2,p)                          % Plot figur nr. p
9         plot((1:d{p}.filt.N/2)*(d{p}.Fs/d{p}.filt.N), ...
10             d{p}.filt.Ps(1:d{p}.filt.N/2))
11         xlim(flim)                              % Begræns udbredning i x
12         xlabel('$f$ [Hz]', 'Interpreter','Latex', 'FontSize', 10);
13         title(d{p}.meta.name, 'Interpreter','Latex', 'FontSize', 15)
14         grid on;
15         txt = ['Smooth.vindue: ', num2str(d{p}.filt.Ps_sw,2), '[Hz]'];

```

```

16     yLimits = get(gca, 'YLim');
17     text(flim(2)*0.35, yLimits(2)*0.9, txt)
18 end
19 sgtitle('Powerspektra [$\propto$ V$^2$Hz$^{-1}$]', ...
20         'Interpreter', 'Latex', 'FontSize', 20);
21 end

```

6.9 ecg_find_r_ibi

```

1 function [ecg] = ecg_find_r_ibi(ecg, threshold)
2 % Finder R-takker vha. findpeaks og tærskelværdi
3 % Janus Bo Andersen, 2019
4     ecg.filt.R.threshold = threshold;           % Gem tærskelværdi
5     [ecg.filt.R.pk, ecg.filt.R.lk] = ...        % R Peakværdi og Lokation
6         findpeaks(ecg.filt.filtered, ...        % Find i filtreret data
7         ecg.Fs, 'MinPeakHeight', threshold);
8
9     ecg.filt.ibi = diff(ecg.filt.R.lk);         % IBI i sekunder
10    ecg.filt.ibi_norm = ecg.filt.ibi - mean(ecg.filt.ibi); % normaliseret
11
12    ecg.filt.HR = (60 / ecg.filt.T)*length(ecg.filt.R.pk); % puls i BPM
13 end

```

6.10 ecg_plot4_peaks

```

1 function [] = ecg_plot4_peaks(d, tlim)
2 % Plotter 4x1 figur med R-takker
3 % d indeholder 4 ECG-objekter
4 % Janus Bo Andersen, 2019
5     setlatexstuff('Latex'); figure             % Figurindstillinger
6
7     for p = 1:4
8         subplot(4,1,p)                         % Plot figur nr. p
9         plot(d{p}.filt.t, d{p}.filt.filtered, ...
10             d{p}.filt.R.lk, d{p}.filt.R.pk, 'ro')
11         xlabel('$t$ [s]', 'Interpreter','Latex', 'FontSize', 10);
12         ylabel(['Signal', ' [' , d{p}.meta.unit, ' ]'], ...
13             'Interpreter','Latex', 'FontSize', 10);
14         txt = [d{p}.meta.name, ' . ', ...
15             'Threshold: ', num2str(d{p}.filt.R.threshold,2), ...
16             ' [' , d{p}.meta.unit, ' ]~', ...
17             'Gns. puls:~', num2str(d{p}.filt.HR,3), ' min~$^{-1}$'];
18         title(txt, 'Interpreter','Latex', 'FontSize', 10)
19     grid on;

```

```

20     xlim(tlim);
21     if p == 1
22         legend('EKG [mV]', 'Detekteret R-tak', ...
23             'Location', 'SouthEast')
24     end
25 end
26 xlabel('$t$ [s]', 'Interpreter','Latex', 'FontSize', 15);
27 sgtitle('Detektion af R-takker i EKG-signaler i [mV]', ...
28     'Interpreter', 'Latex', 'FontSize', 20);
29 end

```

6.11 ecg_plot4_tachogram

```

1 function [] = ecg_plot4_tachogram(d)
2 % Plotter 4x1 figur med tachogrammer
3 % d indeholder 4 ECG-objekter
4 % Janus Bo Andersen, 2019
5     setlatexstuff('Latex'); figure % Figurindstillinger
6
7     for p = 1:4
8         subplot(4,1,p) % Plot figur nr. p
9         N = length(d{p}.filt.ibi_norm);
10        stem(0:N-1, d{p}.filt.ibi_norm)
11        hold on
12        plot(0:N-1, d{p}.filt.ibi_norm, 'b-')
13        hold off
14        xlabel('$t$ [s]', 'Interpreter','Latex', 'FontSize', 10);
15        ylabel(['Signal', ' [' , d{p}.meta.unit, ']'], ...
16            'Interpreter','Latex', 'FontSize', 10);
17        txt = [d{p}.meta.name, '.~', ...
18            '$\sigma^2$~=', num2str(var(d{p}.filt.ibi_norm),2)];
19        title(txt, 'Interpreter','Latex', 'FontSize', 10)
20        grid on;
21        if p == 1
22            legend('Norm. IBI [s]', '(lin. interpol.)', ...
23                'Location', 'SouthEast')
24        end
25    end
26    xlabel('Samples [m]', 'Interpreter','Latex', 'FontSize', 15);
27    sgtitle('Tachogram viser normaliseret IBI i [s] per R-tak sample', ...
28        'Interpreter', 'Latex', 'FontSize', 20);
29 end

```

6.12 ecg_ibi_resample

```
1 function [ecg] = ecg_ibi_resample(ecg, desiredFs)
2 % Resampler et uregelmæssigt samplet IBI-signal til jævnt samplede værdier
3 % Janus Bo Andersen, 2019
4
5     x = ecg.filt.ibi_norm;           % Vi vil resample normaliseret IBI
6     Tx = ecg.filt.R.lk(1:end-1);     % Regner øjeblikkelig puls fra m=0
7                                     % og -1 fordider er regnet diff.
8
9     [y, Ty] = resample(x, Tx, desiredFs, 'spline'); % Resampling med
10                                           % splines
11
12     ecg.filt.resample.ibi_norm = y;   % Resamplet tidsserie
13     ecg.filt.resample.t = Ty;         % Nye samplingstidspunkter
14     ecg.filt.resample.Fs = desiredFs; % Gem ny Fs
15 end
```

6.13 ecg_show_resampling

```
1 function [] = ecg_show_resampling(d)
2 % Viser 4x1 diagram over resultater af resampling
3 % Janus Bo Andersen, 2019
4     setlatextstuff('Latex'); figure % Figurindstillinger
5
6     for e = 1:4
7         subplot(4,1,e)
8         x = d{e}.filt.ibi_norm;       % Oprindelig IBI
9         Tx = d{e}.filt.R.lk(1:end-1); % Oprindelige tidsp.
10        y = d{e}.filt.resample.ibi_norm; % Resamplet IBI
11        Ty = d{e}.filt.resample.t;     % Nye samplingstidspunkter
12
13        plot(Ty,y,Tx,x, 'ro')          % Plot sammenligning
14        title(d{e}.meta.name, 'Interpreter','Latex', 'FontSize', 10)
15        grid on;
16        if e == 1
17            legend('Resampled norm. IBI [s]', 'Oprind. norm. IBI [s]', ...
18                  'Location', 'SouthEast')
19        end
20    end
21    xlabel('$t$ [s]', 'Interpreter','Latex', 'FontSize', 15);
22    sgtitle('Resultat af resampling af normaliseret IBI, i [s]', ...
23            'Interpreter', 'Latex', 'FontSize', 20);
24 end
```

6.14 ecg_hrv_powerspectrum

```
1 function [ecg] = ecg_hrv_powerspectrum(ecg, smooth_bins)
2 % Regner Powerspectrum på resamplet IBI-data, regner LF/HF-ratio.
3 % Benytter smoothing af Powerspektrum
4 % smooth_bins: Antal bins, i glidende gennemsnit (skal være ulige)
5 % Janus Bo Andersen, 2019
6
7     x = ecg.filt.resample.ibi_norm;
8     N = length(x);
9     w = ones(N,1); % Mulighed for at udskifte til vinduesfkt
10    X = fft(x.*w'); % Windowed FFT
11    P = 2*(X.*conj(X))/(N^2); % Power for hver frekv.sample
12    ecg.filt.resample.ibi_Ps = smoothMag(P, smooth_bins); % Gl. gennemsnit
13    ecg.filt.resample.resolution = ecg.filt.resample.Fs / N; % Ny frekv.opl
14    ecg.filt.resample.N = N;
15
16    f = (0:N-1) * ecg.filt.resample.resolution; % Frekvensakse
17    ecg.filt.resample.f = f;
18
19    % Beregning af LF og HF tager en genvej: Skaleringsfaktorer, der er ens
20    % for HF og LF udelades (intervallængde, skalering til Hz, osv.) da
21    % disse faktorer udgår i ratioen.
22    lfidx = [f > 0.04 & f <= 0.15]; % Frekvenser i LF
23    hfidx = [f > 0.15 & f <= 0.4]; % Frekvenser i HF
24    LF = sum(P(lfidx)); % LF Energy
25    HF = sum(P(hfidx)); % HF Energy
26    ecg.filt.resample.LF = LF; % Gem værdi
27    ecg.filt.resample.HF = HF; % Gem værdi
28    ecg.filt.resample.LFHF = LF / HF; % Beregn LF/HF-ratio
29 end
```

6.15 ecg_plot4_hrv

```
1 function [] = ecg_plot4_hrv(d, flim)
2 % Plotter 2x2 figur med HRV powerspektra, og sætter xlim = flim
3 % d indeholder 4 ECG-objekter
4 % Janus Bo Andersen, 2019
5     setlatexstuff('Latex'); figure % Figurindstillinger
6
7     for p = 1:4
8         subplot(2,2,p) % Plot figur nr. p
9
10        N = d{p}.filt.resample.N;
11        Nmax = floor(N/2);
```



```

12     X = d{p}.filt.resample.ibi_Ps(1:Nmax);
13     f = (0:Nmax-1) * d{p}.filt.resample.resolution;
14     LFHF = d{p}.filt.resample.LFHF;
15
16     plot(f, X)
17     xline(0.04); xline(0.15); xline(0.4);
18     xlim(flim) % Begræns udbredning i x
19     xlabel('$f$ [Hz]', 'Interpreter','Latex', 'FontSize', 10);
20     ylabel('Power [$\propto s^2$Hz$^{-1}$]', ...
21     'Interpreter','Latex', 'FontSize', 10);
22     title([d{p}.meta.name, ' . LF/HF~~', num2str(LFHF,3)], ...
23     'Interpreter','Latex', 'FontSize', 15)
24     grid on;
25     txt = ['0plsn.: ', ...
26     num2str(d{p}.filt.resample.resolution, 2), '[Hz]'];
27     yLimits = get(gca,'YLim');
28     sc = 0.8;
29     if p > 2
30         sc = 0.1; % Placering på diagr. for diagnosticerede
31     end
32     text(flim(2)*0.35, yLimits(1)+(yLimits(2)-yLimits(1))*sc, txt)
33 end
34 sgtitle('HRV Powerspektra [$\propto s^2$Hz$^{-1}$]', ...
35 'Interpreter', 'Latex', 'FontSize', 20);
36 end

```

6.16 smoothMag (KPL)

```

1 function Y = smoothMag(X,M)
2 % Smoothing of signal. Eg. frequency magnitude spectrum.
3 % X must be a row vector, and M must be odd.
4 % KPL 2016-09-19
5     N=length(X);
6     K=(M-1)/2;
7     Xz=[zeros(1,K) X zeros(1,K)];
8     Yz=zeros(1,2*K+N);
9     for n=1+K:N+K
10         Yz(n)=mean(Xz(n-K:n+K));
11     end
12     Y=Yz(K+1:N+K);
13 end

```

Bibliografi

- [1] Samuel Thrysoe. *ST2PRJ2 - Introduktion til projektet*. 2018. URL: <https://www.youtube.com/watch?v=UhSmKs8xrrk&list=PLRqTBecQ3YQkl-j21eProysBn50qeKkg0&index=1> (sidst set 04.12.2019).
- [2] Physikalisch-Technische Bundesanstalt. *PTB Diagnostic ECG Database*. 2004. URL: <https://physionet.org/content/ptbdb/1.0.0/> (sidst set 05.12.2019).
- [3] PhysioNet. *Computing in Cardiology Challenge 2011*. 2011. URL: <https://archive.physionet.org/challenge/2011/> (sidst set 05.12.2019).
- [4] Københavns Universitet. *Basalt: Introduktion til EKG*. ukendt årstal. URL: <https://absalon.instructure.com/courses/15408/pages/basalt-introduktion-til-ekg> (sidst set 04.12.2019).
- [5] Travis Van Slooten. *Understanding the basics of an ECG/EKG*. 2017. URL: <https://www.livingwithatrialfibrillation.com/2853/ecg-ekg-basics/> (sidst set 04.12.2019).
- [6] MathWorks. *Real-Time ECG QRS Detection*. 2016. URL: <https://www.mathworks.com/help/dsp/examples/real-time-ecg-qrs-detection.html> (sidst set 07.12.2019).
- [7] Københavns Universitet. *Basalt: Rytme*. ukendt årstal. URL: <https://absalon.instructure.com/courses/15408/pages/basalt-rytme> (sidst set 04.12.2019).
- [8] Christopher Watford. *Understanding ECG Filtering*. 2014. URL: <http://ems12lead.com/2014/03/10/understanding-ecg-filtering/#gref> (sidst set 04.12.2019).
- [9] Samuel Thrysoe. *ST2PRJ2 - Gennemgang af sygdomme*. 2018. URL: <https://www.youtube.com/watch?v=8mXRJHkGcwI&list=PLRqTBecQ3YQkl-j21eProysBn50qeKkg0&index=3> (sidst set 04.12.2019).
- [10] Jaakko Malmivuo og Robert Plonsey. *Ch. 19 - Basis of ECG Diagnosis*. 1995. URL: <http://www.bem.fi/book/19/19.htm> (sidst set 04.12.2019).
- [11] Sundhed.dk. *Atrieflimren og -flagren*. 2017. URL: <https://www.sundhed.dk/sundhedsfaglig/laegehaandbogen/hjerte-kar/tilstande-og-sygdomme/arytmier/atrieflimren-og-flagren/> (sidst set 04.12.2019).
- [12] Københavns Universitet. *Atrieflimren*. ukendt årstal. URL: <https://absalon.instructure.com/courses/15408/pages/atrieflimren> (sidst set 04.12.2019).
- [13] Task Force of the European Society of Cardiology the North American Society of Pacing Electrophysiology. *Heart Rate Variability - Standards of Measurement, Physiological Interpretation, and Clinical Use*. 1996. URL: <https://www.ahajournals.org/doi/10.1161/01.CIR.93.5.1043> (sidst set 08.12.2019).
- [14] George E. Billman. *Heart Rate Variability - A Historical Perspective*. 2011. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3225923/> (sidst set 08.12.2019).
- [15] Mathworks. *Resampling Nonuniformly Sampled Signals*. 2019. URL: <https://www.mathworks.com/help/signal/examples/resampling-nonuniformly-sampled-signals.html> (sidst set 08.12.2019).
- [16] Wikipedia.org. *Duration and circumstances of ECG recording*. 2019. URL: https://en.wikipedia.org/wiki/Heart_rate_variability#Duration_and_circumstances_of_ECG_recording (sidst set 09.12.2019).