

## BILAG 7

---

# Implementering & Test

---

19. december 2017

# Indholdsfortegnelse

---

<b>Kapitel 1</b>	<b>Indledning</b>	<b>3</b>
<b>Kapitel 2</b>	<b>Implementering</b>	<b>4</b>
2.1	Hardware . . . . .	4
2.1.1	Instrumentationsforstærker 1 . . . . .	4
2.1.2	Strømgenerator . . . . .	4
2.1.3	Instrumentationsforstærker 2 . . . . .	5
2.1.4	OP-AMP . . . . .	6
2.1.5	Spektrumanalyse . . . . .	6
2.1.6	AA filter . . . . .	7
2.2	Software . . . . .	8
2.2.1	Funktioner . . . . .	9
2.2.1.1	Synkerefleksmonitor_OpeningFcn . . . . .	9
2.2.1.2	Btn_Start_Measurements . . . . .	9
2.2.1.3	Btn_Save_Measuerments . . . . .	9
2.2.1.4	Start_GUI . . . . .	9
2.2.1.5	Generate_SineWave . . . . .	10
2.2.1.6	Read_Measurements . . . . .	10
2.2.1.7	Process_Measurements . . . . .	10
2.2.1.8	Show_Measurements . . . . .	11
2.2.1.9	Save_Measurments . . . . .	11
<b>Kapitel 3</b>	<b>Modultest</b>	<b>12</b>
3.1	Hardware . . . . .	12
3.1.1	Instrumentationsforstærker 1 . . . . .	12
3.1.2	Strømgenerator . . . . .	13
3.1.3	Instrumentationsforstærker 2 . . . . .	14
3.1.4	OP-AMP . . . . .	15
3.1.5	Spektrumanalyse . . . . .	16
3.1.6	AA filter . . . . .	17
3.2	Software . . . . .	18
3.2.1	Funktioner . . . . .	18
3.2.1.1	Synkerefleksmonitor_OpeningFcn . . . . .	18
3.2.1.2	Btn_Start_Measurment . . . . .	18
3.2.1.3	Btn_Save_Measurment . . . . .	18
3.2.1.4	Start_GUI . . . . .	18
3.2.1.5	Start_GUI . . . . .	19
3.2.1.6	Generate_SineWave . . . . .	19
3.2.1.7	Read_Measurements . . . . .	21

## Indholdsfortegnelse

3.2.1.8	Process_Measurements . . . . .	25
3.2.1.9	Show_Measurements . . . . .	28
3.2.1.10	Save_Measurments . . . . .	28
<b>Kapitel 4</b>	<b>Integrationstest</b>	<b>29</b>
<b>Kapitel 5</b>	<b>Accepttest</b>	<b>30</b>
<b>Litteratur</b>		<b>31</b>

# Indledning 1

---

I dette bilag beskrives, hvordan de designede hardware og software-komponenter er implementeret og modultestet. På baggrund af disse designede enheder i designfasen foretages der nu implementering og modultest. Til implementering af hardware komponenter er der valgt at opbygge testkredsløbet på et fumlebræt. Hver af disse komponenter skal igennem en modultest for at verificere om de kan bruges til det tiltænkte formål. For softwarens vedkommende dokumenteres, hvordan kodeimplementeringen er foregået, samt resultatet af disse implementeringer. Formålet med modultesten er at forberede produktet til integrationstest.

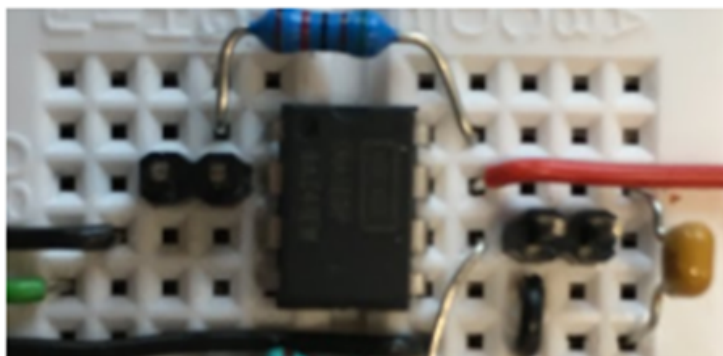
# Implementering 2

---

## 2.1 Hardware

### 2.1.1 Instrumentationsforstærker 1

Denne instrumentationsforstærker er tiltænkt til at forstærke et AC signal på 2V til 4V. Til dette formål er der anvendt instrumentationsforstærkeren INA128 med de begrundelser, som er angivet i designafsnittet. Figur 2.1 viser INA128 og en eksterne modstand, som bruges til at fastsætte den ønskede forstærkning.

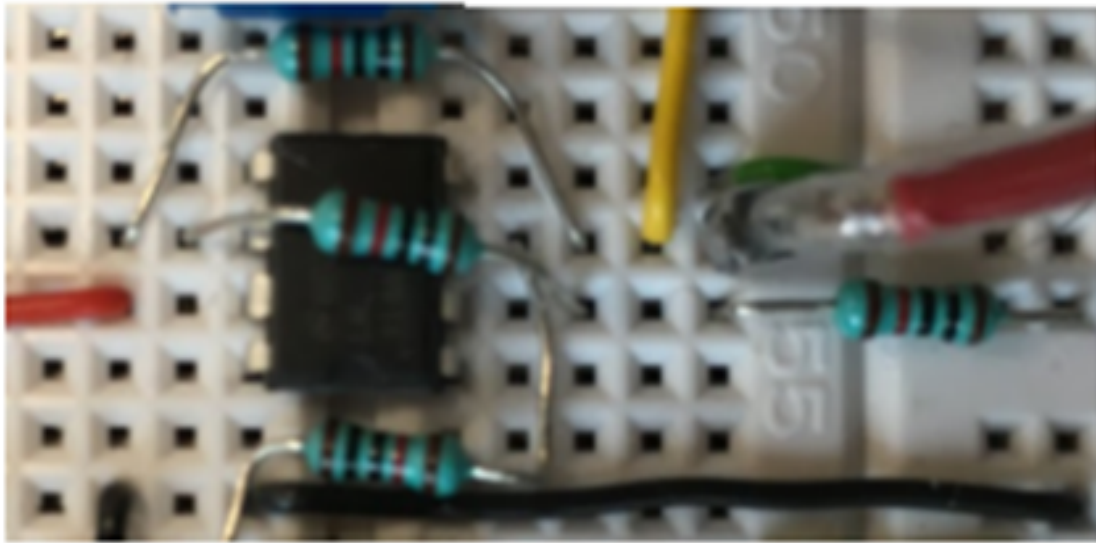


*Figur 2.1: Figuren viser, hvordan INA128 er implementeret på et fumlebræt*

### 2.1.2 Strømgenerator

Strømgeneratorens funktion er at levere en konstant strøm, som sendes til et måleobjektets væv. Til implementering af denne strømgenerator er der anvendt operationsforstærkeren LM318. Figur 2.2 viser komponenten LM318 med de tilhørende modstande.

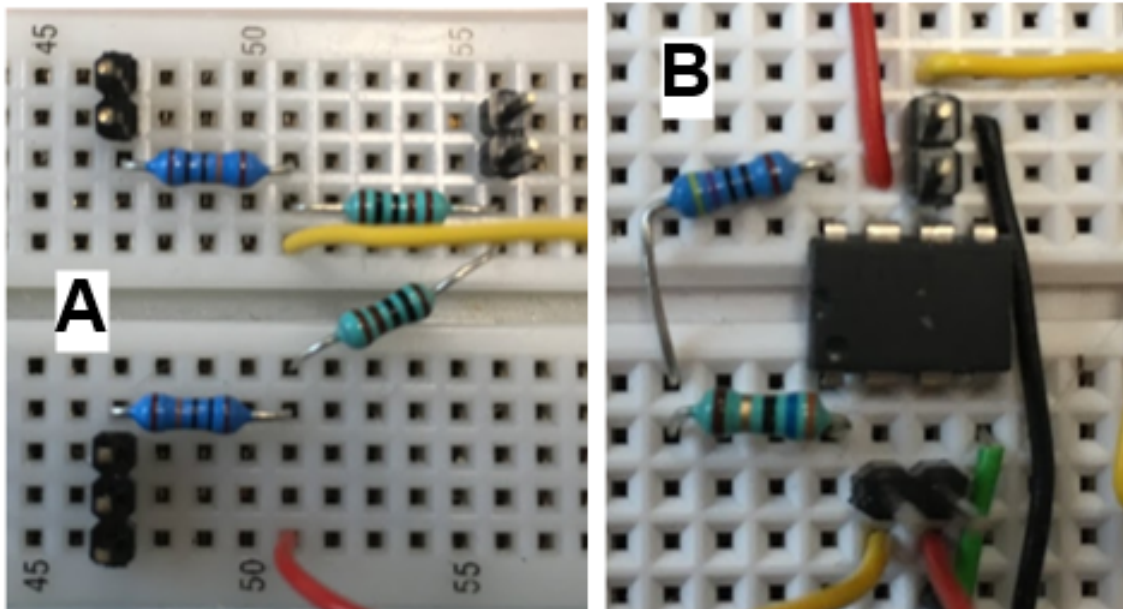
## 2.1. Hardware



*Figur 2.2: Figuren viser, hvordan LM318 er implementeret på et fumlebræt*

### 2.1.3 Instrumentationsforstærker 2

Instrumentationsforstærker 2 bliver brugt til at forstærke biosignal fra måleobjektets og undertrykkelse støj . Til implementering af denne Instrumentationsforstærker er der igen brugt INA128. Figur 2.3 B viser komponenten INA128 med dens tilhørende eksterne modstande. Figur 2.3 A viser en spændingsdeler kredsløb, der er benyttet til at teste instrumentationsforstærkeren. Hvorfor der anvendes en spændingsdeler til test af INA128 henvises der til "*bilag 6 - Design*".

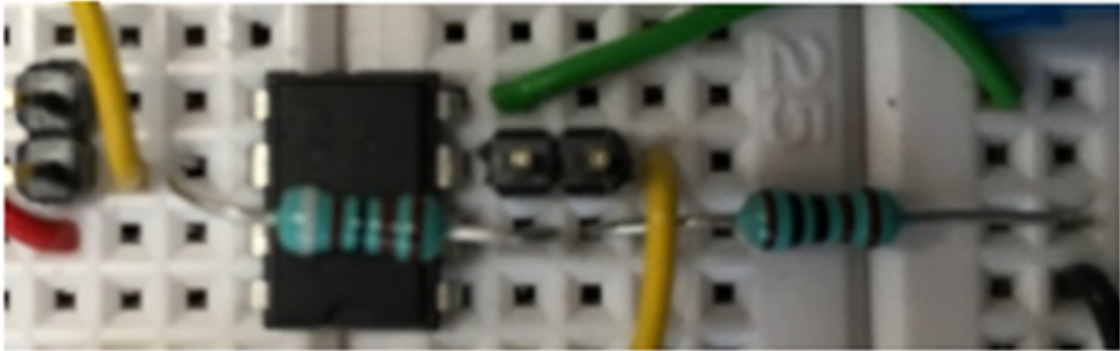


*Figur 2.3: Figuren viser, hvordan LM318 er implementeret på et fumlebræt*

## 2.1. Hardware

### 2.1.4 OP-AMP

Biosignalet fra måleobjektet forstærkes op i to trin. Det første trin benyttes Instrumentationsforstærker 2 og det andet trin anvendes operationsforstærkeren LM318, som forstærker signalet fra Instrumentationsforstærker 2 yderligere. Figur 2.4 viser implementering af LM318 og de to modstande, som fastsætter, hvor meget forstærkning man kan få ud af operationsforstærkeren.

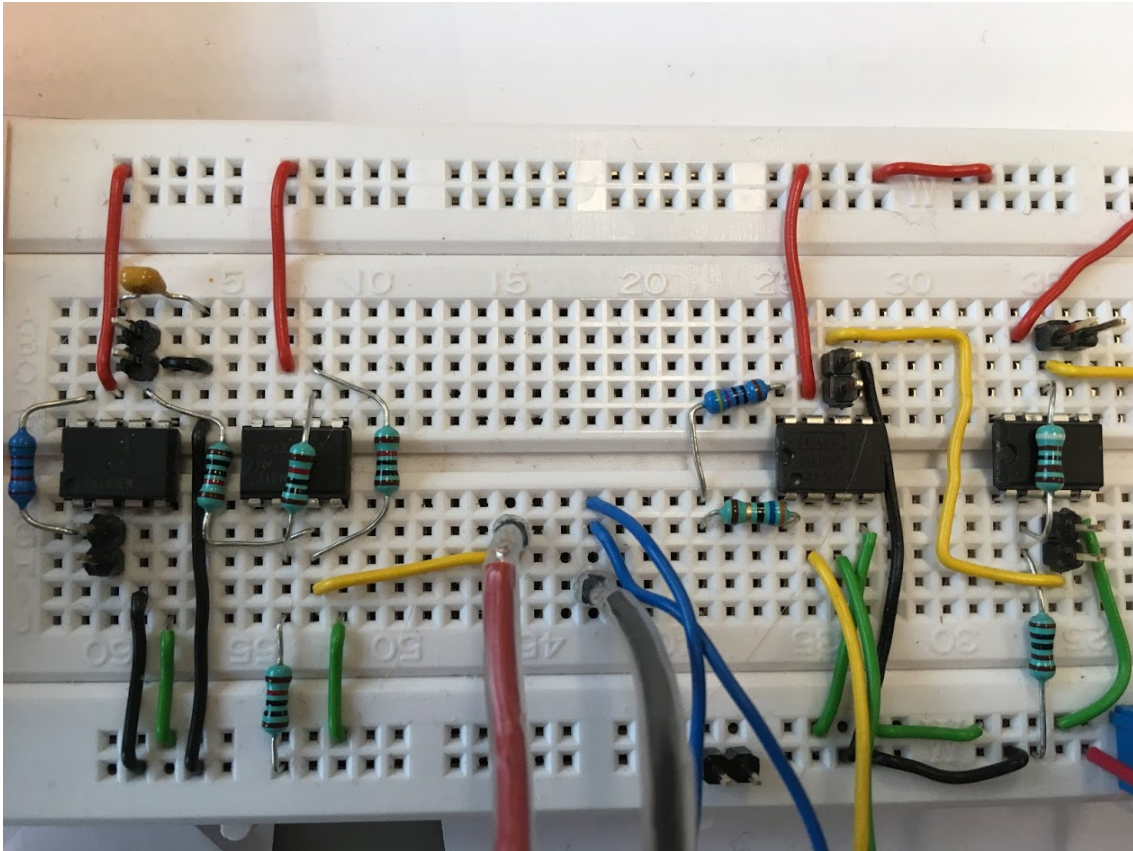


*Figur 2.4: Figuren viser, hvordan LM318 er implementeret på et fumlebræt*

### 2.1.5 Spektrumanalyse

Der er behov for at foretage en spektrumanalyse, med alle komponenter sat sammen, for at vurdere hvilket krav der skal være til AA filterets dæmpning.

## 2.1. Hardware



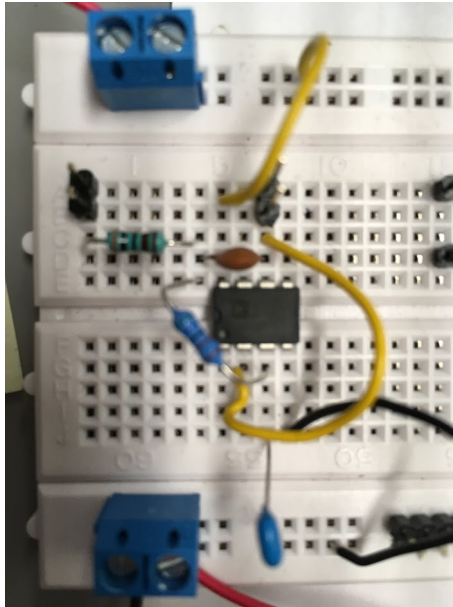
*Figur 2.5: Billede over den samlede opstilling med komponenterne: Instrumentationsforstærker 1 & 2, strømgenerator, OP-AP og monteret elektroder. Disse komponenter repræsenterer det samlede frekvensspektrum.*

### 2.1.6 AA filter

For at undgå aliasering i det optagede signal implementeres et anti-aliaseringsfilter foran Analog Discovery. AA filteret modtager et forstærkingsignal fra OPAMP, filtrerer det og sender det videre til Analog Discovery.



## 2.2. Software



*Figur 2.6: Implementering af AA filter.*

## 2.2 Software

I dette afsnit beskrives de vigtigste funktioner for systemet software-del. Til implementering af disse funktioner er der brugt udviklingsværktøjet Matlab pga. følgende fordele:

- Matlab understøtter styring af Analog Discovery
- Matlab forærer databehandlingsfunktioner som ligger klar til anvendelse
- Matlab er god til at indlæse store mængde data med få kodelinjer
- I Matlab kan man udvikle en brugergrænseflade nemt og hurtigt
- Ved brug af Matlab med Analog Discovery, kan man betjene funktionsgeneratoren, dataopsamlingsenhed og brugergrænsefladen fra ét sted

Alternativet til overnævnte fordele er at man bruger forskellige “single purpose” programmer og enheder, når man skal måle et biomedicinsk signal.

I det følgende gennemgås de kritiske funktioner for programmet.

## 2.2. Software

### 2.2.1 Funktioner

Funktionernes nærmere beskrivelser henvises der til "*bilag 6 - Design*".

#### 2.2.1.1 Synkerefleksmonitor\_OpeningFcn

```
1 function Synkerefleksmonitor_OpeningFcn(hObject, eventdata, handles, varargin)
2 % This function has no output args, see OutputFcn.
3 % hObject handle to figure
4 % eventdata reserved - to be defined in a future version of MATLAB
5 % handles structure with handles and user data (see GUIDATA)
6 % varargin command line arguments to Synkerefleksmonitor (see VARARGIN)
7
8 % Choose default command line output for Synkerefleksmonitor
9 handles = Start_GUI(handles);
10 % Update handles structure
11 guidata(hObject, handles);
```

#### 2.2.1.2 Btn\_Start\_Measurements

```
1 function Btn_Start_Measurements_Callback(hObject, eventdata, handles)
2 % hObject handle to Btn_Start_Measurements (see GCBO)
3 % eventdata reserved - to be defined in a future version of MATLAB
4 % handles structure with handles and user data (see GUIDATA)
5 handles = Generate_SineWave(handles);
6 handles = Read_Measurements(handles);
7 handles = Process_Measurements(handles);
8 guidata(hObject,handles);
9 handles = Show_Measurements(handles);
10 set(handles.Btn_Save_Measurements,'Visible','On');
11 guidata(hObject,handles);
```

#### 2.2.1.3 Btn\_Save\_Measuerments

```
1 function Btn_Save_Measurements_Callback(hObject, eventdata, handles)
2 % hObject handle to Btn_Save_Measurements (see GCBO)
3 % eventdata reserved - to be defined in a future version of MATLAB
4 % handles structure with handles and user data (see GUIDATA)
5 handles = Save_Measurements(handles);
6 guidata(hObject,handles);
```

#### 2.2.1.4 Start\_GUI

```
1 function handles = Start_GUI(handles)
2 %% Denne funktion har til opgave at saette tilfoeje objekter til GUI op naar programmet startes.
3 set(handles.txtDate, 'String', (datestr(datestr(now)))); %Dato og tid vises
4
5 axes(handles.axes3); %Visning af billede i GUI
6 myImage = imread('SynkBillede.PNG');
7 image(myImage);
8 axis off
9 axis image
10
11 set(handles.Btn_Save_Measurements,'Visible','Off'); %Save Measurements knappen er ikke synlig
12 end
```

## 2.2. Software

### 2.2.1.5 Generate\_SineWave

```
1 function handles = Generate_SineWave(handles)
2 % For at faa Analog Discovery til at fungere med Matlab skal denne saettes op med
3 % foelgende indstillinger:
4 s = daq.createSession('digilent'); % Opretter forbindelse til Analog Discovery
5
6 fgenBI = addFunctionGeneratorChannel(s, 'AD1', 1, 'Sine'); % Opsaetning af funktionsgenerator og
    waveform
7
8 fgenBI.Gain = 2; % Amplitude paa 2V
9
10 fgenBI.Frequency = 20000; % Frekvens paa 20 kHz
11
12 handles.GS = s; % Giver mulighed for starte en maaling vha. handles.GS
13 end
```

### 2.2.1.6 Read\_Measurements

```
1 function handles = Read_Measurements(handles)
2 % Her bliver maalingen oprettet og startet.
3 s = handles.GS; %Analog Discovery indstillinger fra "Generate_SineWave" funktionen
4 aiBI = addAnalogInputChannel(s, 'AD1', 1, 'Voltage'); %Oprettelse af analog ind 1 og 2
5 aiEMG = addAnalogInputChannel(s, 'AD1', 2, 'Voltage');
6 s.Rate = 500000; % Samplingrate
7 s.DurationInSeconds = 2; % Tid pr. maaling
8
9 % Visning af message boksen "Measurements running..."
10 h = figure('units','pixels','position',[300 300 300 80],'windowstyle','modal');
11 uicontrol('FontSize',12,'style','text','text','string','Measurements
    running...','units','pixels','position',[40 30 200 20]);
12
13 [data, timestamps] = startForeground(s); % Maaling startes og genererer data og tid
14 close(h); % Lukker message boksen "Measurements running..."
15
16 % Maalingerne gemmes i handles til videre brug
17 handles.data = data;
18 handles.timestamps = timestamps;
19
20 % Rydder op og slukker forbindelsen til Analog Discovery
21 s.release();
22 delete(s);
23 clear s;
24 end
```

### 2.2.1.7 Process\_Measurements

```
1 function handles = Process_Measurements(handles)
2 % Digital envelope af BI signalet.
3 RawBI = handles.data(:,1); % Det raa BI signal tages ud af data
4
5 % Dobbeltensret BI signal
6 BIabs = abs(RawBI);
7
8 % Udglattr BI signal med lavpas filter.
9 % Knaekfrekvens = 500 Hz
10 % Daemper en dekade frem til 5000 Hz
11 lpFilt = designfilt('lowpassfir', 'PassbandFrequency', 0.001, ...
12     'StopbandFrequency', 0.01, 'PassbandRipple', 0.2, ...
13     'StopbandAttenuation', 40, 'DesignMethod', ...
14     'kaiserwin');
15
```

## 2.2. Software

```
16 BIsignal = filter(lpFilt,BIabs); %BI signalet filteret
17 handles.BIsignal = BIsignal; %BI signalet gemmes i handles til senere visning
18 end
```

### 2.2.1.8 Show\_Measurements

```
1 function handles = Show_Measurements(handles)
2 % De optaget signaler BI og EMG plottes
3
4 axes(handles.axes1); % Vises i axes1
5 subplot(2,1,1)
6 plot(handles.timestamps, handles.BIsignal,'k'); % Det behandlet BI signal
7 title('Bioimpedance')
8 ylabel('Impedance (ohm)');
9
10 subplot(2,1,2)
11 plot(handles.timestamps, handles.data(:,2),'k'); % EMG signalet
12 title('EMG')
13 xlabel('Time in seconds')
14 ylabel('Voltage (V)');
15 end
```

### 2.2.1.9 Save\_Measurments

```
1 function handles = Save_Measurements(handles)
2 % Maalinger gemmes i csv-fil.
3 [FileName,PathName] = uiputfile('.csv'); % Oprettelse af prompt til brugeren, hvor filnavn og sti
   skal bestemmes
4
5 filepath=fullfile(PathName,FileName); % Sti og filnavn sammensættes
6
7 % Message boksen "Saving measurements..." vises
8 f = figure('units','pixels','position',[300 300 300 80],'windowstyle','modal');
9 uicontrol('FontSize',12,'style','text','text','Saving
   measurements...','units','pixels','position',[40 30 200 20]);
10
11 pause(1); % Pauser i 1 sek for at message boksen kan naa at blive vist under gemmeprocessen starter
12
13 csvwrite(filepath,([handles.BIsignal,handles.data(:,2),handles.timestamps])); % Behandlet BI
   signal, EMG signal og tid samles i en csv-fil.
14
15 close(f); %Lukker message boksen igen efter csv-filen er gemt
16
17 h = msgbox('Measurements saved!'); % Der informeres om at csv-filen er gemt
18 end
```

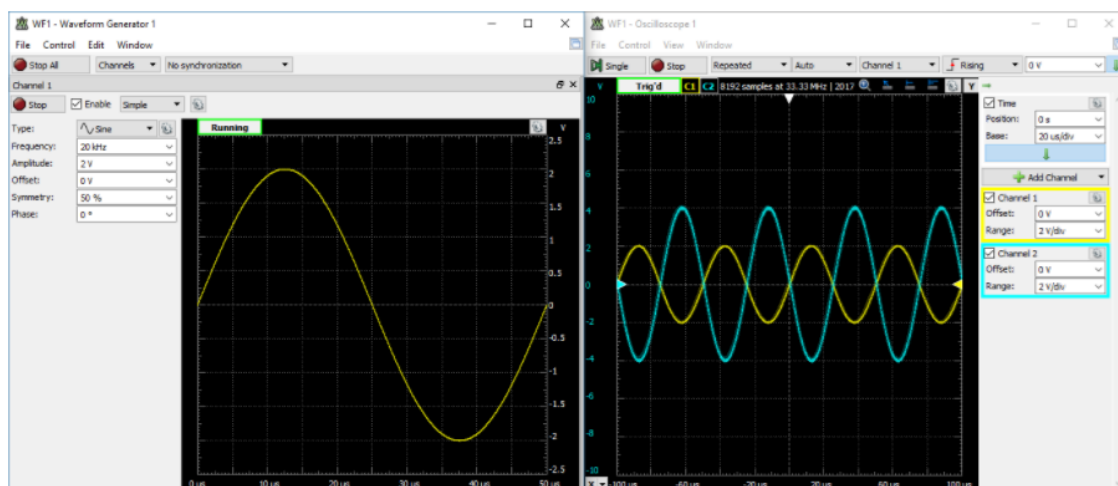
# Modultest 3

Modultest af hardware-delen består af en simuleret test vha. Multisim og en praktisk test. Nogle komponenterne eksister ikke i Multisim og kræver at blive oprettet, men det er valgt at ikke bruge tid på det, da det er tidskrævende. Derfor præsenteres kun praktiske resultater for disse komponenter. I det følgende præsenteres testresultaterne for instrumentationsforstærker 1, 2, strømgeneratoren, operationsforstærkeren og AA filteret.

## 3.1 Hardware

### 3.1.1 Instrumentationsforstærker 1

Dette modul er testet ved at sende 2V/20kHz fra Analog Discovery (den gule kurve) igennem INA128. Det ses på figur 3.1 at de 2V bliver forstærket til 4V (turkis kurve) ved udgangen af INA128. Dette resultat stemmer overens med det beregnede resultat i designfasen.

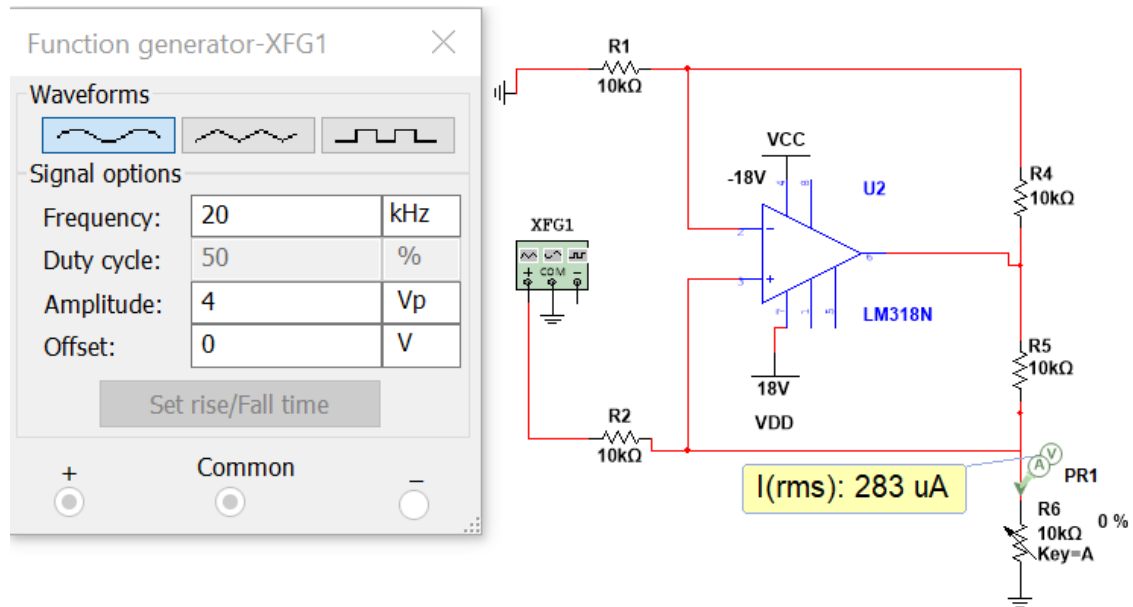


Figur 3.1: Figuren viser resultatet af INA128, som forstærker 2V til 4V

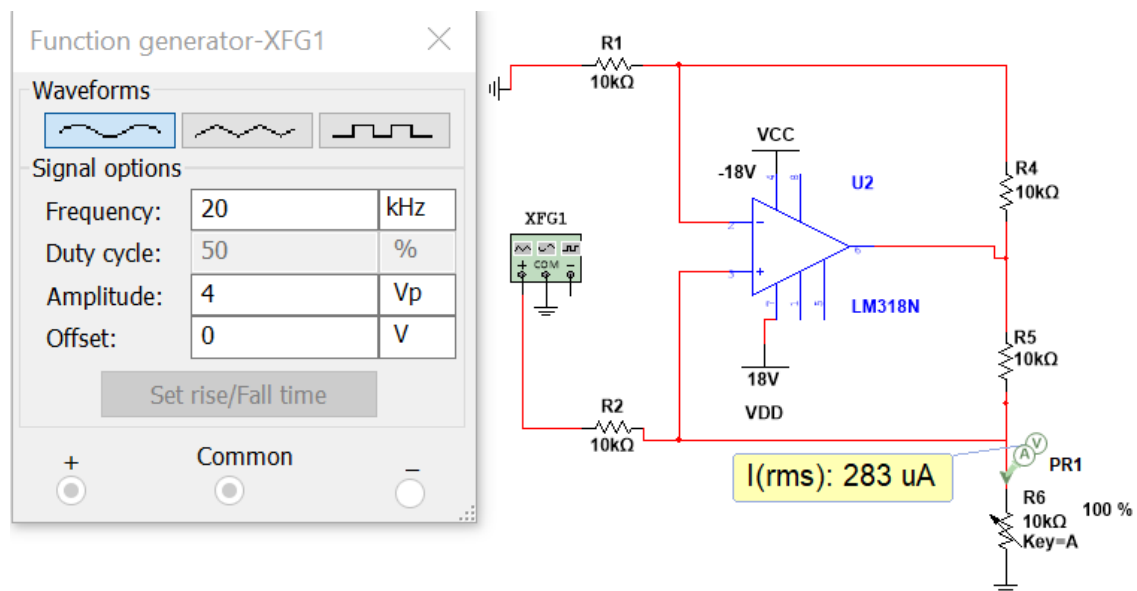
### 3.1. Hardware

#### 3.1.2 Strømgenerator

Strømgeneratoren er simuleret ved at den får 4V fra en funktionsgenerator. På baggrund af denne spænding genereres der 283uA ud af strømgeneratoren. Bemærk at figur 3.2 viser strømudgangen ved no-load, hvorimod figur 3.3 viser når man belaster strømgeneratoren med  $10k\Omega$ .



Figur 3.2: Figuren viser det simuleret resultat for strømgeneratoren ved no-load

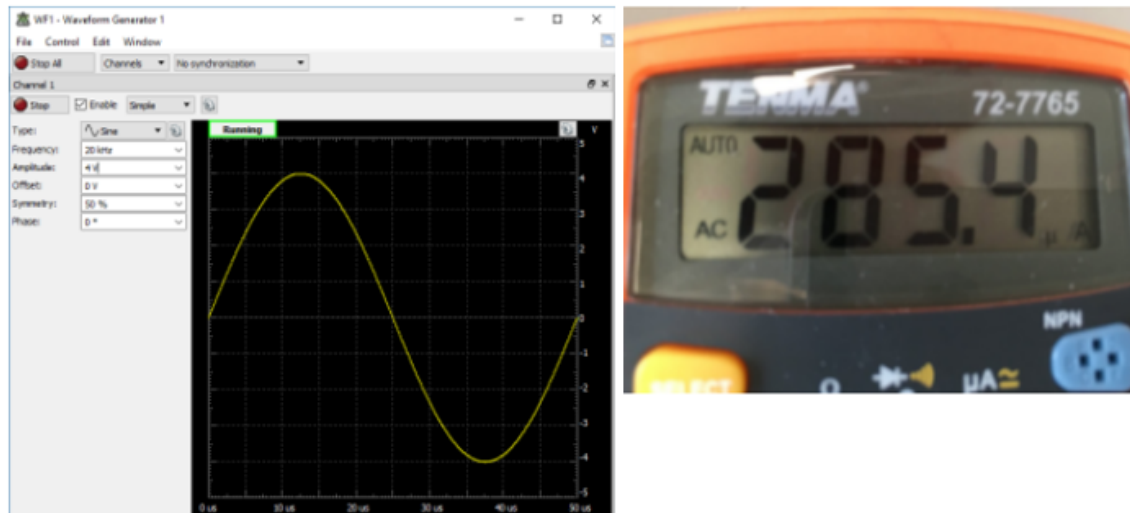


Figur 3.3: Figuren viser det simuleret resultat for strømgeneratoren med load på  $10k\Omega$

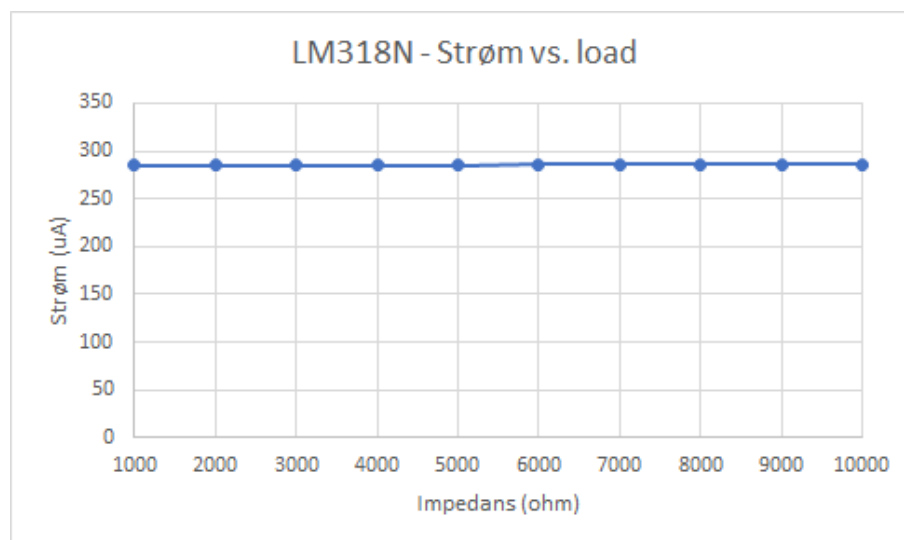
Det ses på figur 3.3 at strømmen ikke ændrer sig, selvom man belaster kredsløbet med  $10k\Omega$ .

### 3.1. Hardware

Tilsvarende sendes der 4V ind i strømgeneratoren, når der skal foretages den praktiske test på fumlebræt. De 4V genereres fra Analog Discovery. Den producerede strøm måles vha. et amperemeter i serie på udgangen af den anvendte operationsforstærker, LM318. Det ses på figur 3.4 at udgangsstrømmen, som genereres af strømgeneratoren er målt til 285uA. Dette resultat afviger lidt fra det beregnede og simulerede resultat. Det vurderes at afvigelsen er så lille at den ingen betydning har for måleobjektets sikkerhed. Figur 3.5 viser strømmens stabilitet fra 1 k $\Omega$  til 10 k $\Omega$ , hvilket svarer til området for load impedansen for et biologisk væv [1]. Derfor er det besluttet at arbejde videre med den målte strøm.



Figur 3.4: Figuren viser det praktisk strømresultat som er målt ved udgangen af strømgeneratoren.



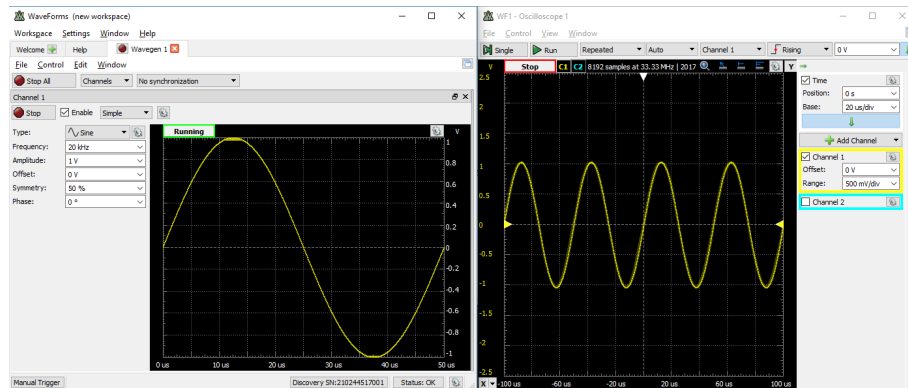
Figur 3.5

#### 3.1.3 Instrumentationsforstærker 2

Til test af instrumentationsforstærker 2 er der benyttet en spændingsdeler, som får 1V fra en funktionsgeneratoren, Analog Discovery. Spændingsdelerens funktion er at reducere 1V til 10mV, som efterfølgende forstærkes af instrumentationsforstærker 2 med faktor 100

### 3.1. Hardware

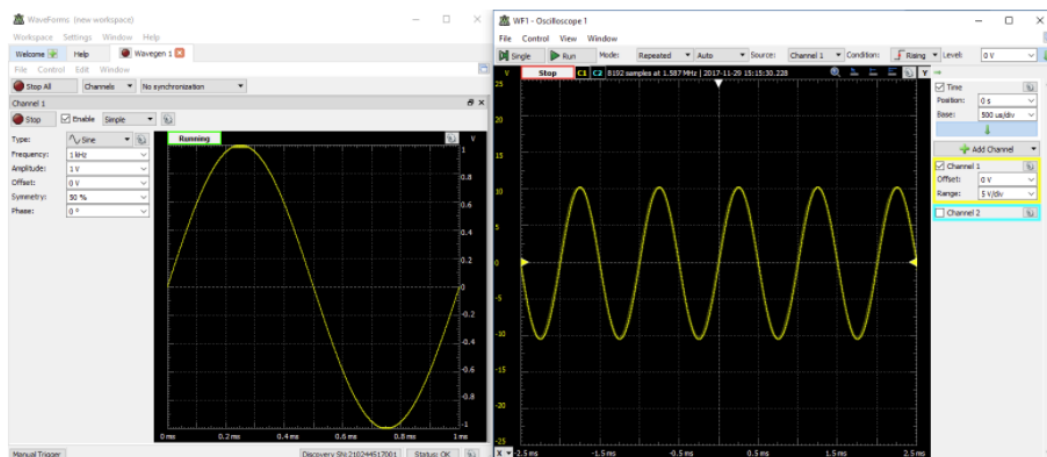
gange. Som figur 3.6 viser, kan man med instrumentationsforstærkeren INA128 opnå en forstærkning på 100. Dette resultat stemmer overens med det teoretiske resultat, som er beregnet i designfasen.



Figur 3.6: Figuren viser det praktiske udgangsspænding for INA128. Denne INA128 yder en forstærkning på 100 gange.

#### 3.1.4 OP-AMP

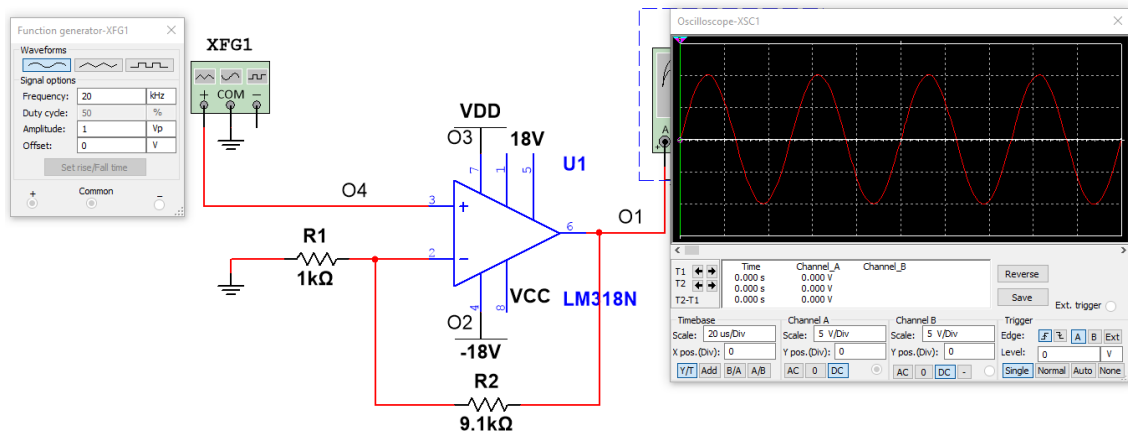
Denne operationsforstærkers opgave er at forstærke 1 V til 10 V, dvs. en forstærkning ved faktor 10. Der sendes 1V fra funktionsgeneratoren, Analog Discovery, som derefter bliver forstærket til 10 V. Figur 3.7 viser at signalet fra funktionsgeneratoren bliver forstærket 10 gange, dvs. den målte udgangsspænding er på 10V. Hermed stemmer den målte og den teoretiske udgangsspænding af operationsforstærkeren LM318 med hinanden.



Figur 3.7: Figuren viser det praktiske udgangsspænding som er målt ved udgangen af operationsforstærkeren.



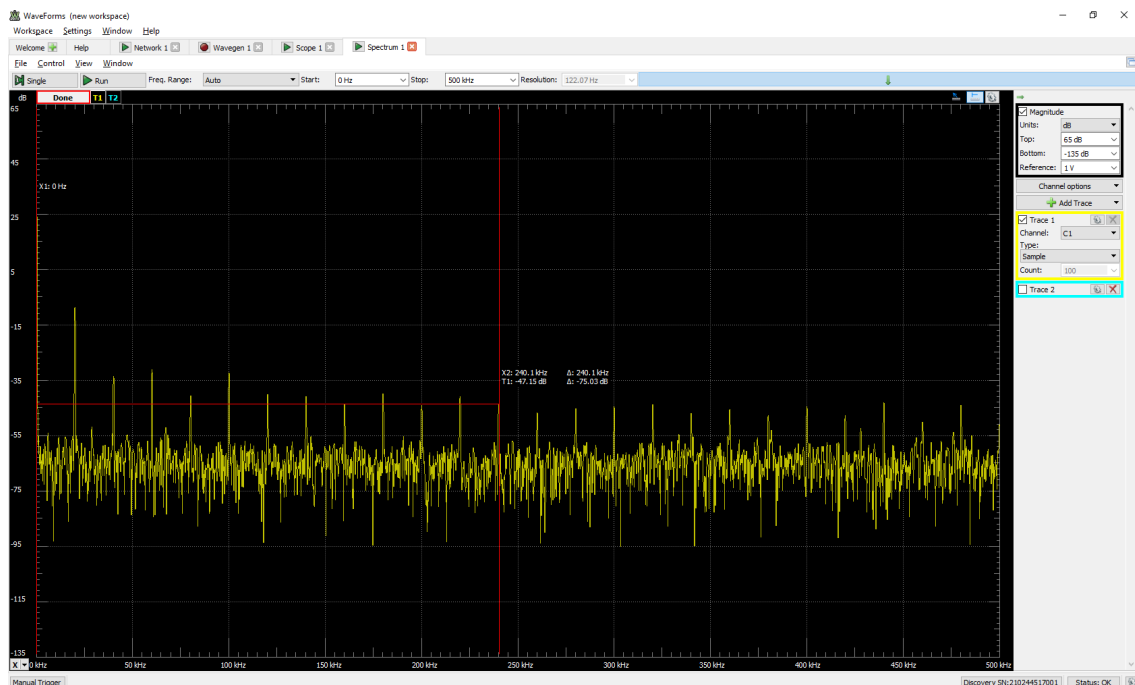
### 3.1. Hardware



Figur 3.8: Figuren viser det simuleret udgangsspænding som er målt ved udgangen af operationsforstærkeren.

#### 3.1.5 Spektrumanalyse

Ved at sammensætte alle komponenter inkl. elektroder, kan spektrumanalysen blive optaget vha. Analog Discovery. Som det kan aflæses på figur 3.9, er der en dæmpning fra det samlede kredsløb på -75 dB. Dette aflæses fra den højeste peak i passbåndet til ca. 250 kHz, som er den halve samplingfrekvens. Da Analog Discovery arbejder med 14bit (skal dæmpe ned til -90 dB) kræver det en yderligere dæmpning på ca. 15 dB. Dette kunne realiseres med et 1. ordens filter, men der vælges at benytte et 2. ordens filter, for at undgå de variationer der måtte være i passbåndet i det målte signal.

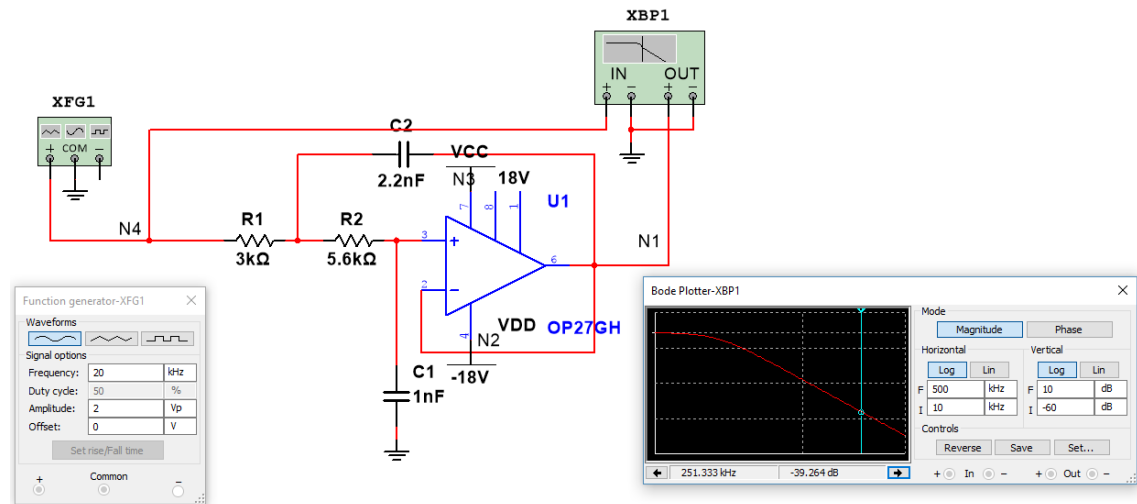


Figur 3.9: Det implementeret frekvensspektrum, som viser at der er en samlet dæmpning på ca. 75dB som skal yderligere dæmpes til 90dB med et 2.ordens lavpasfilter.

### 3.1. Hardware

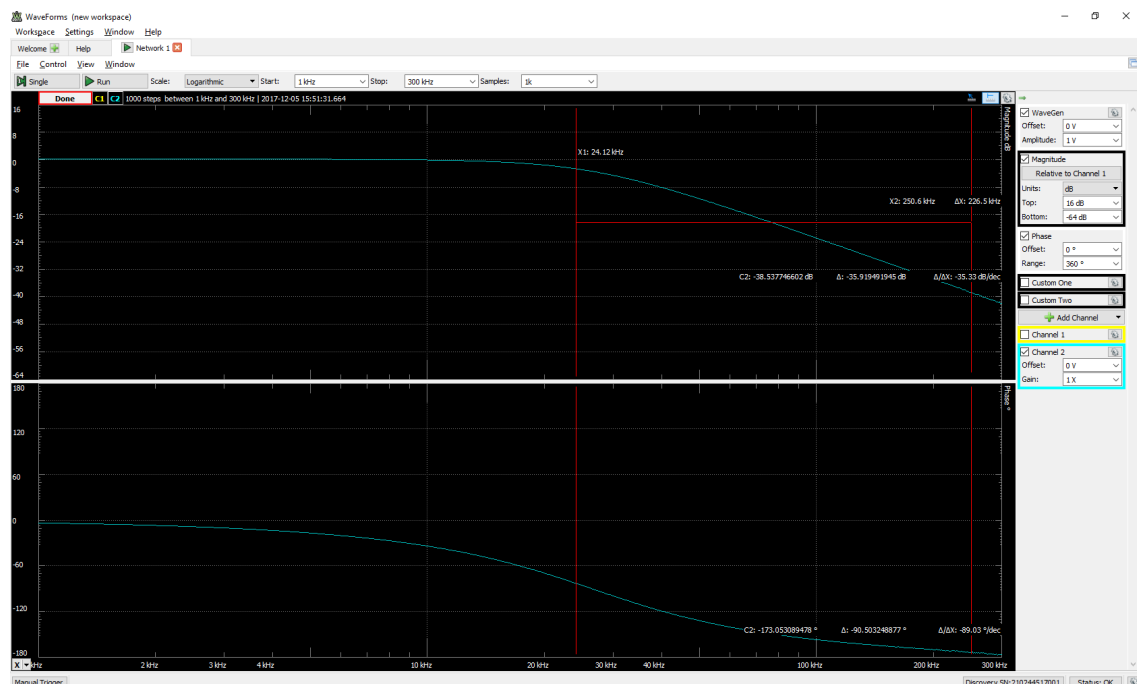
#### 3.1.6 AA filter

Først er AA filteret simuleret i Multisim. Ved hjælp af Bode Plotter i Multisim, kan det aflæses ved 250 kHz at er en dæmpning på 40 dB.



Figur 3.10: Resultat om filterets virkning simuleret i Multisim

Der er brugt værktøjet Network Analyzer i programmet Waveforms, for at teste filters virkning, se figur 3.15. Det kan konstateres at ved knæfrekvensen (25 kHz) er faseforskydningen ca. 90°. Efterfølgende kan det aflæses, at ved knæfrekvensen og en dekade frem er der en dæmpning på ca. 35 dB.



Figur 3.11: Resultat om filterets virkning fra Network Analyzer i waveforms.

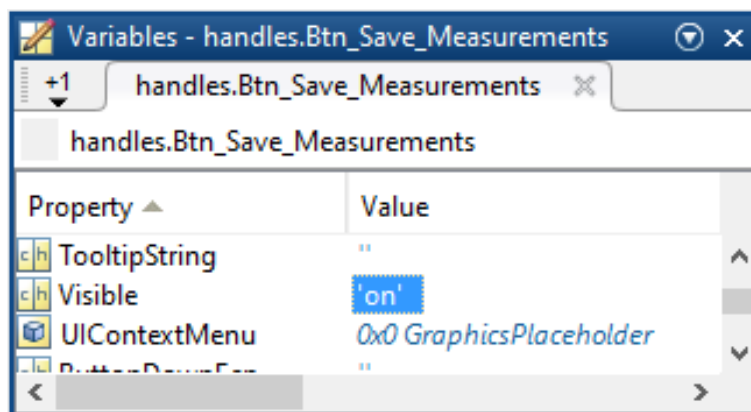
## 3.2. Software

### 3.2 Software

#### 3.2.1 Funktioner

##### 3.2.1.1 Synkerefleksmonitor\_OpeningFcn

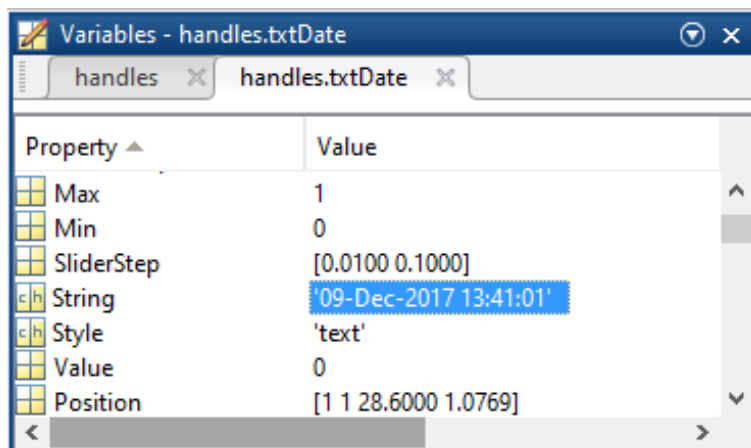
##### 3.2.1.2 Btn\_Start\_Measurment



Figur 3.12: Oprettelse af dato og tid.

##### 3.2.1.3 Btn\_Save\_Measurment

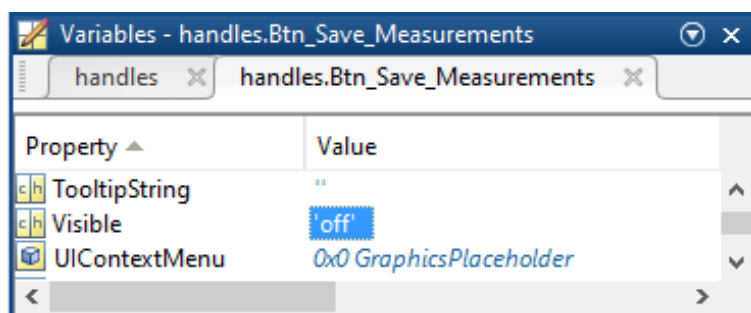
##### 3.2.1.4 Start\_GUI



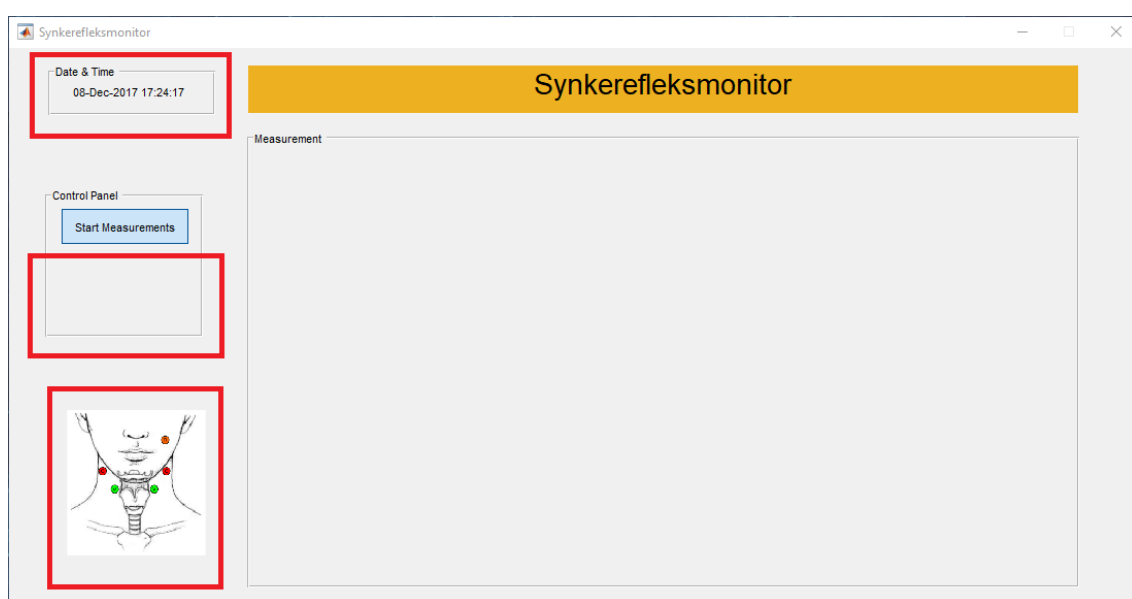
Figur 3.13: Oprettelse af dato og tid.

## 3.2. Software

### 3.2.1.5 Start\_GUI



Figur 3.14: Save\_Measurements knappen er gemt fra start.



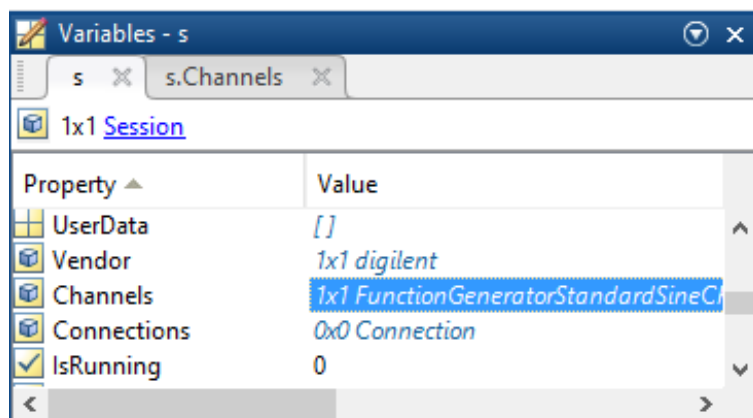
Figur 3.15: Resultat om koden i "Start\_GUI"

### 3.2.1.6 Generate\_SineWave

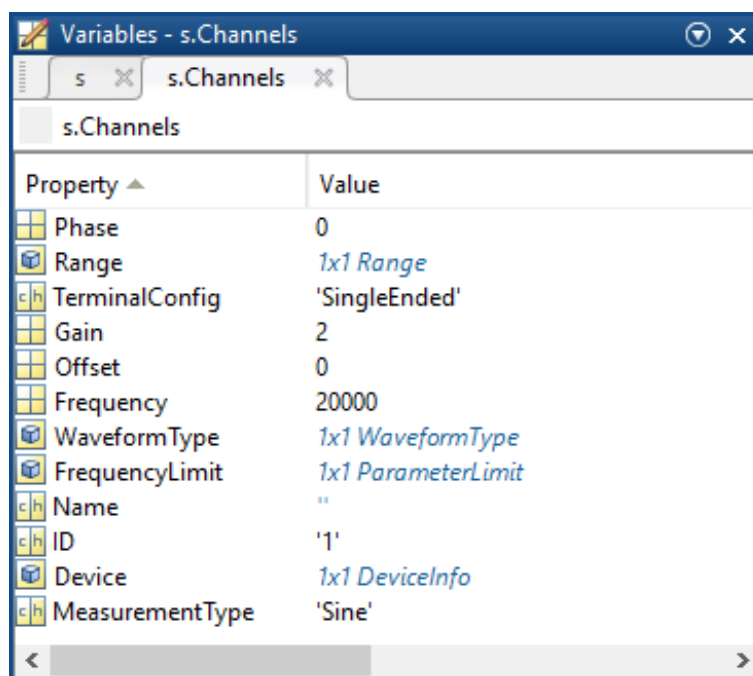


Figur 3.16: Resultat om koden i "Generate\_SineWave"

### 3.2. Software

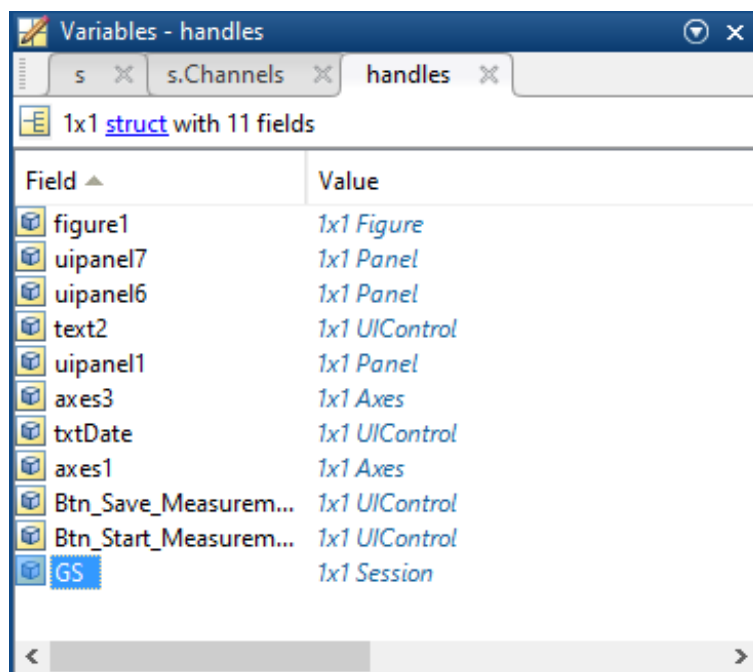


Figur 3.17: Oprettelsen til Analog Discovery

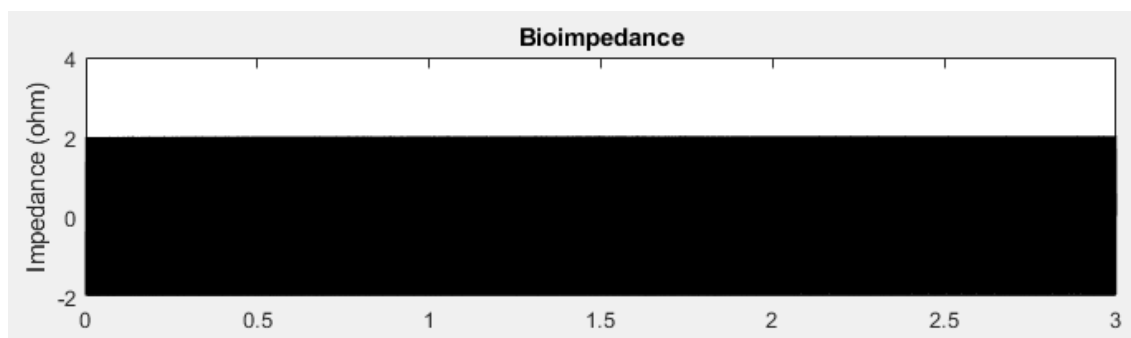


Figur 3.18: Funktionsgenerator tilføjet.

### 3.2. Software

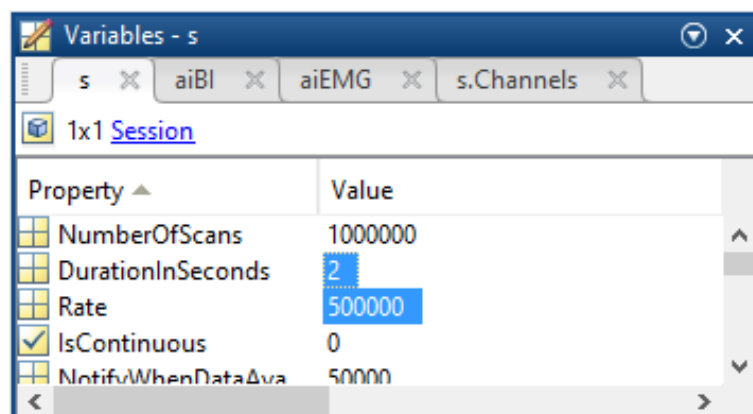


Figur 3.19: Variablen "handles.GS" er oprettet i handles til brug i funktionen "Read\_Measurements".



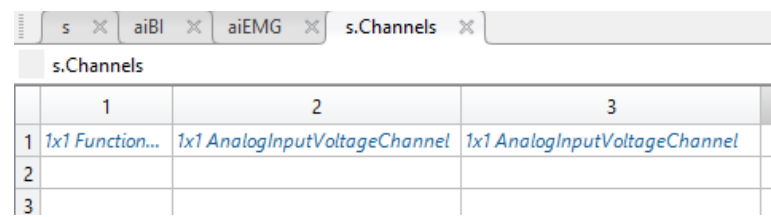
Figur 3.20: GUI resultat om koden i "Generate\_SineWave"

#### 3.2.1.7 Read\_Measurements



Figur 3.21

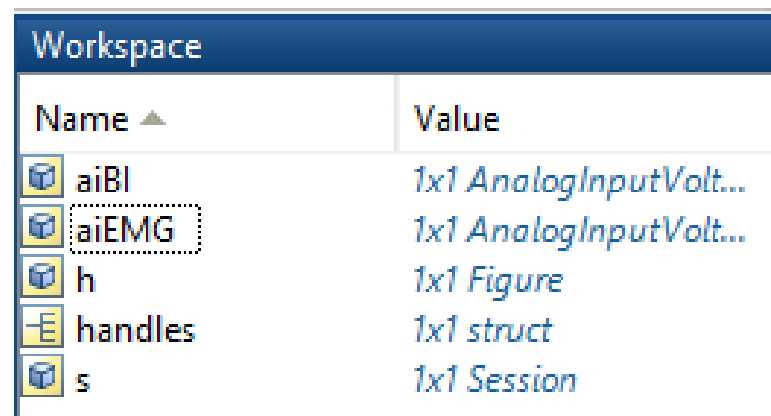
3.2. Software



The image shows a MATLAB window with four tabs: 's', 'aiBI', 'aiEMG', and 's.Channels'. The 's.Channels' tab is active, displaying a table with three columns labeled 1, 2, and 3. Row 1 contains the text '1x1 Function...', '1x1 AnalogInputVoltageChannel', and '1x1 AnalogInputVoltageChannel'. Rows 2 and 3 are empty.

	1	2	3
1	1x1 Function...	1x1 AnalogInputVoltageChannel	1x1 AnalogInputVoltageChannel
2			
3			


Figur 3.22



The image shows the MATLAB Workspace window. It has a title bar 'Workspace' and a table with two columns: 'Name' and 'Value'. The table lists several variables: 'aiBI' (1x1 AnalogInputVolt...), 'aiEMG' (1x1 AnalogInputVolt...), 'h' (1x1 Figure), 'handles' (1x1 struct), and 's' (1x1 Session). The 'aiEMG' variable is highlighted with a dashed border.

Name ▲	Value
aiBI	1x1 AnalogInputVolt...
aiEMG	1x1 AnalogInputVolt...
h	1x1 Figure
handles	1x1 struct
s	1x1 Session

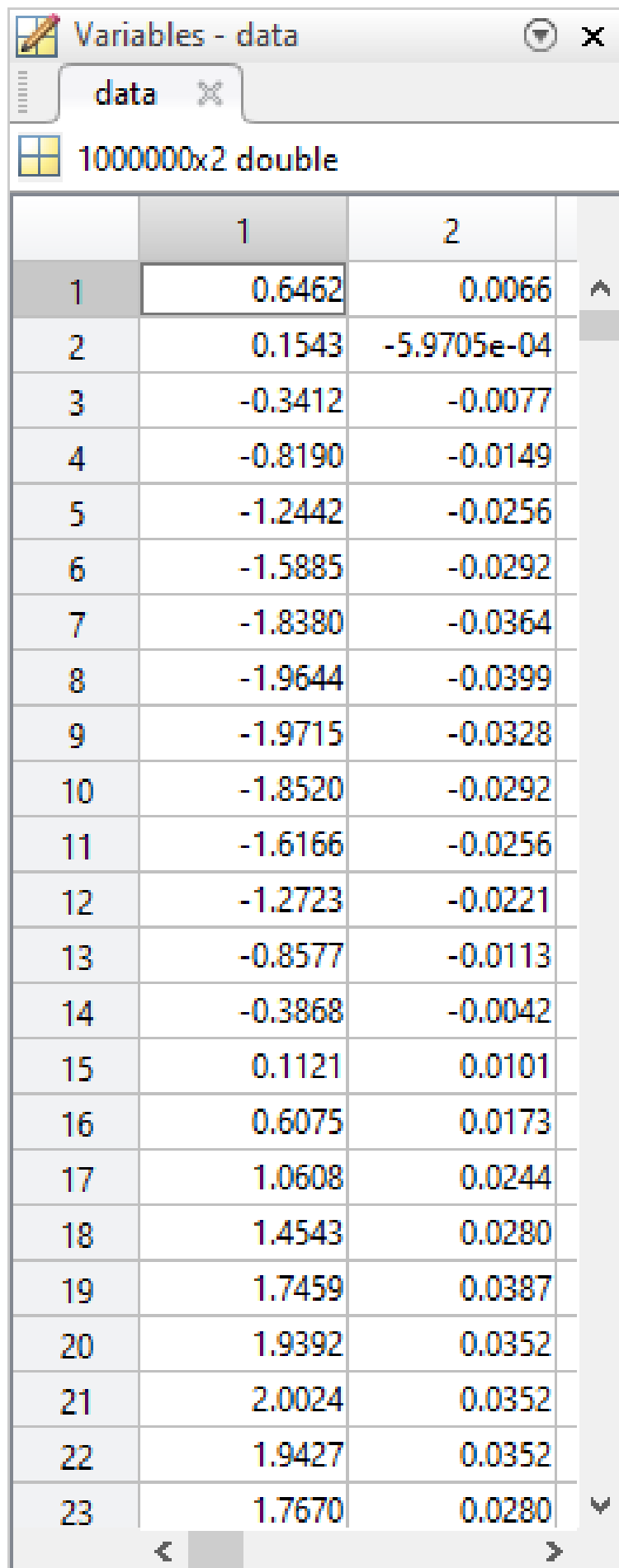
Figur 3.23



The image shows the MATLAB Workspace window with two variables: 'data' (1000000x2 double) and 'timestamps' (1000000x1 double). Both variables are represented by a 2x2 grid icon.

data	1000000x2 double
timestamps	1000000x1 double

Figur 3.24



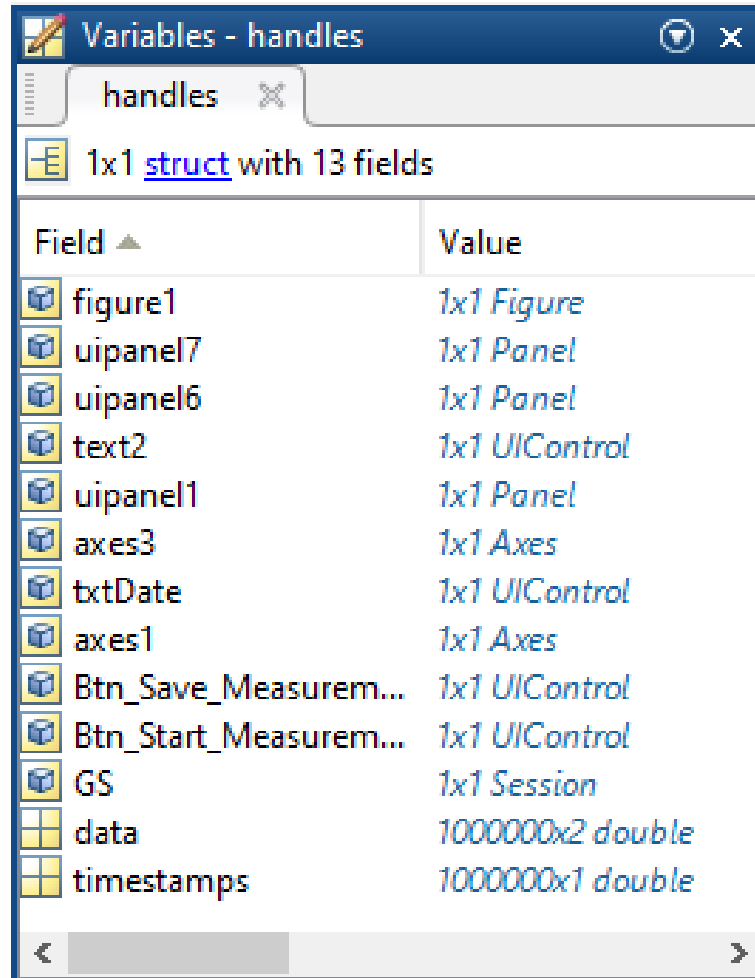
Variables - data

data

1000000x2 double

	1	2
1	0.6462	0.0066
2	0.1543	-5.9705e-04
3	-0.3412	-0.0077
4	-0.8190	-0.0149
5	-1.2442	-0.0256
6	-1.5885	-0.0292
7	-1.8380	-0.0364
8	-1.9644	-0.0399
9	-1.9715	-0.0328
10	-1.8520	-0.0292
11	-1.6166	-0.0256
12	-1.2723	-0.0221
13	-0.8577	-0.0113
14	-0.3868	-0.0042
15	0.1121	0.0101
16	0.6075	0.0173
17	1.0608	0.0244
18	1.4543	0.0280
19	1.7459	0.0387
20	1.9392	0.0352
21	2.0024	0.0352
22	1.9427	0.0352
23	1.7670	0.0280

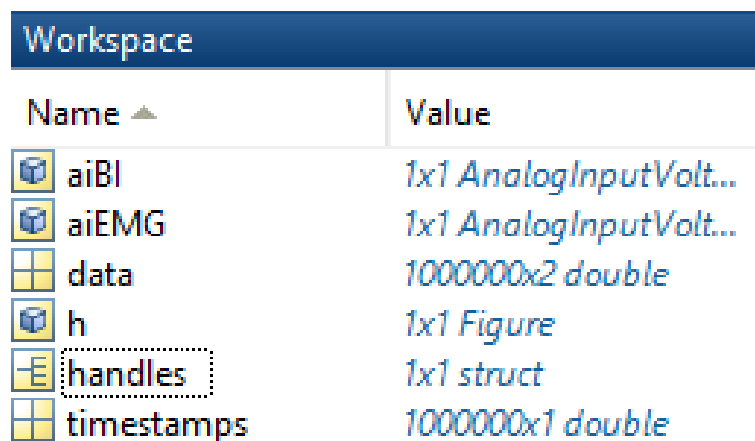




The image shows the 'Variables - handles' window in MATLAB. It displays a 1x1 struct with 13 fields. The fields and their values are listed in a table below.

Field ▲	Value
figure1	1x1 Figure
uipanel7	1x1 Panel
uipanel6	1x1 Panel
text2	1x1 UIControl
uipanel1	1x1 Panel
axes3	1x1 Axes
txtDate	1x1 UIControl
axes1	1x1 Axes
Btn_Save_Measurem...	1x1 UIControl
Btn_Start_Measurem...	1x1 UIControl
GS	1x1 Session
data	1000000x2 double
timestamps	1000000x1 double

Figur 3.26

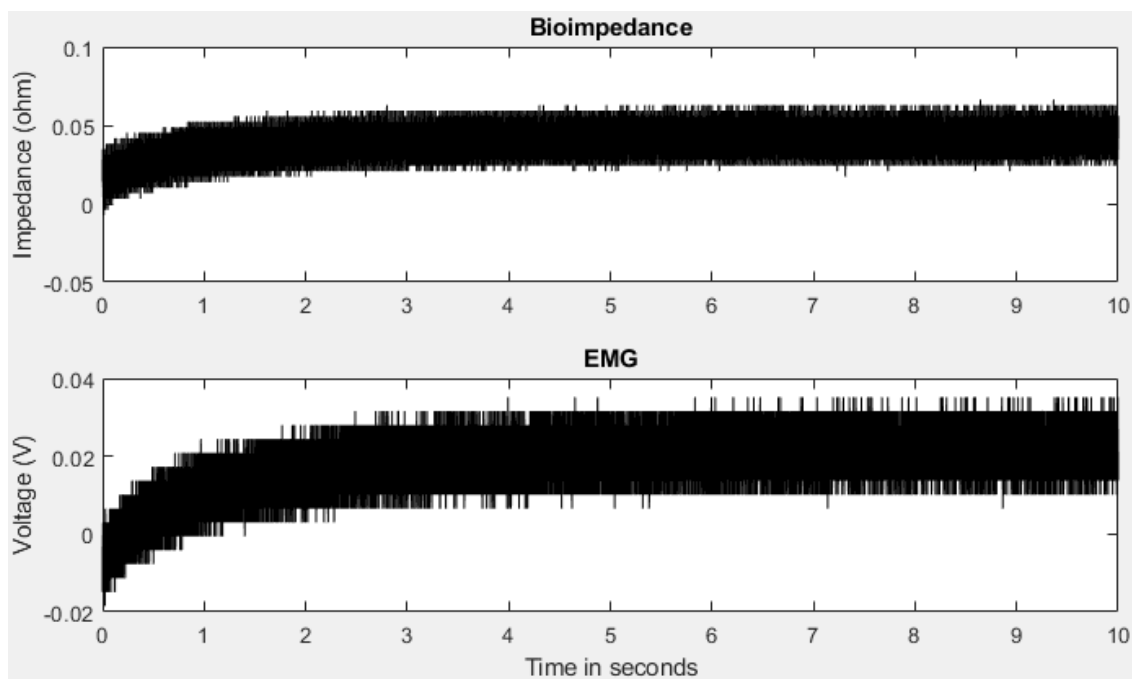


The image shows the 'Workspace' window in MATLAB. It displays a list of variables and their values. The variables are listed in a table below.

Name ▲	Value
aiBI	1x1 AnalogInputVolt...
aiEMG	1x1 AnalogInputVolt...
data	1000000x2 double
h	1x1 Figure
handles	1x1 struct
timestamps	1000000x1 double

Figur 3.27: Session "s"er stoppet og fjernet fra workspace.

### 3.2. Software



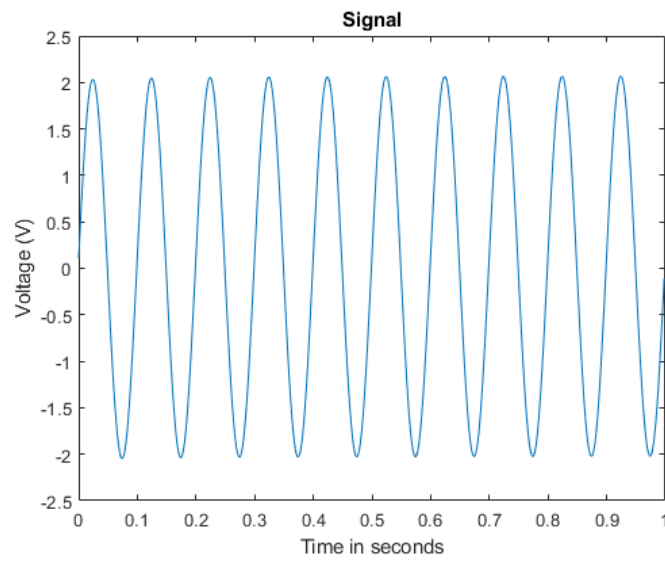
*Figur 3.28: Resultat om koden i "Read\_Measurements"*

#### 3.2.1.8 Process\_Measurements

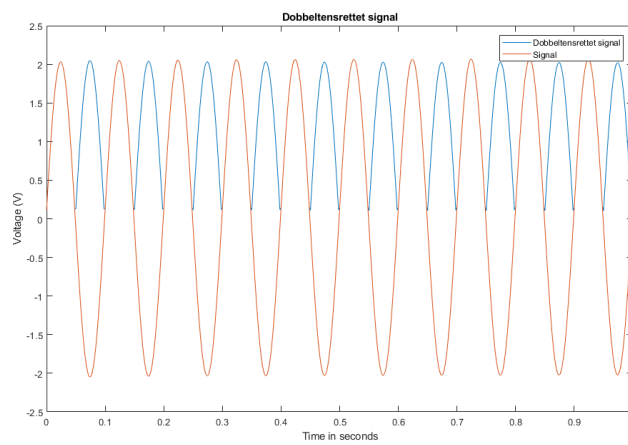


*Figur 3.29: Resultat om koden i "Generate\_SineWave"*

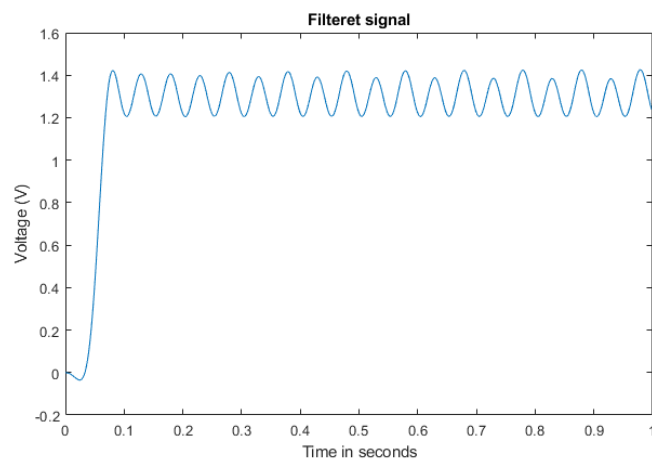
### 3.2. Software



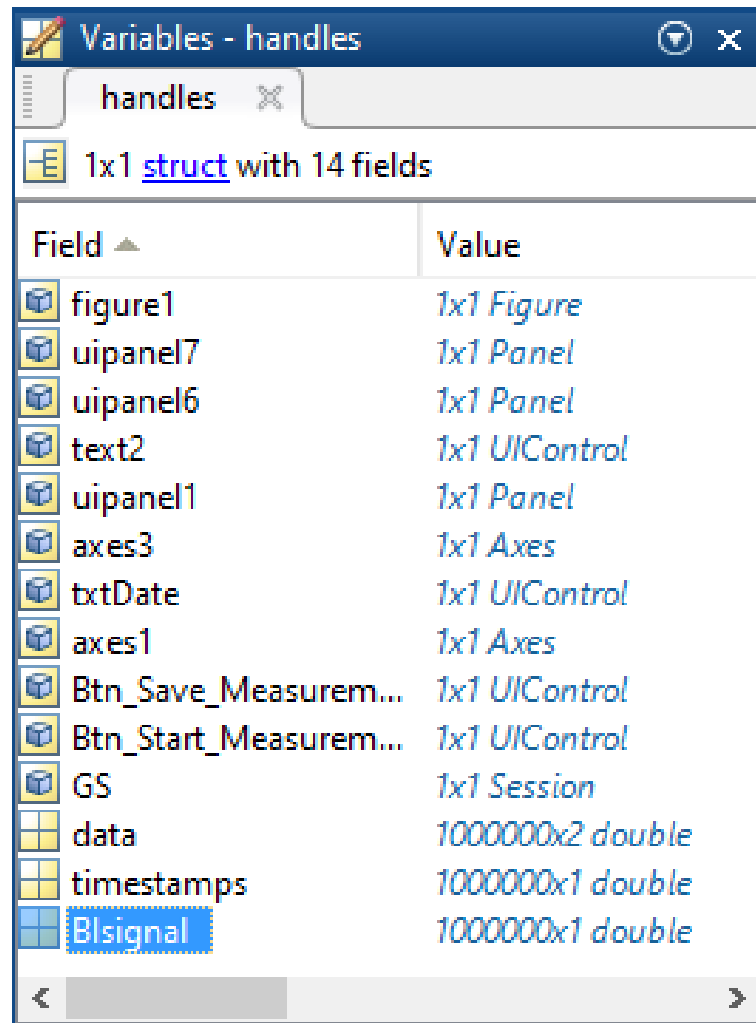
Figur 3.30: Resultat om koden i "Read\_Measurements"



Figur 3.31: Resultat om koden i "Read\_Measurements"



Figur 3.32: Resultat om koden i "Read\_Measurements"



Variables - handles

handles

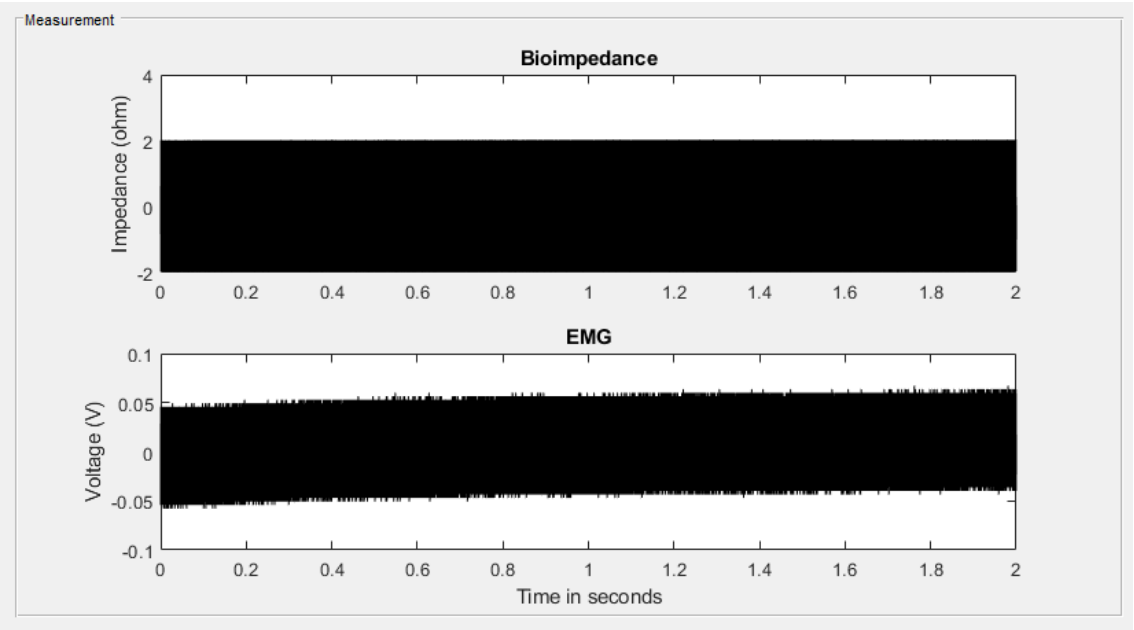
1x1 struct with 14 fields

Field ▲	Value
figure1	1x1 Figure
uipanel7	1x1 Panel
uipanel6	1x1 Panel
text2	1x1 UIControl
uipanel1	1x1 Panel
axes3	1x1 Axes
txtDate	1x1 UIControl
axes1	1x1 Axes
Btn_Save_Measurem...	1x1 UIControl
Btn_Start_Measurem...	1x1 UIControl
GS	1x1 Session
data	1000000x2 double
timestamps	1000000x1 double
Blsignal	1000000x1 double

Figur 3.33

3.2. Software

3.2.1.9 Show\_Measurements



Figur 3.34

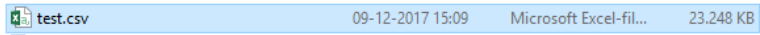
3.2.1.10 Save\_Measurments

Workspace	
Name ▲	Value
FileName	'test.csv'
filepath	'C:\Users\mbana\Desktop\test.csv'
handles	1x1 struct
PathName	'C:\Users\mbana\Desktop\'

Figur 3.35



Figur 3.36

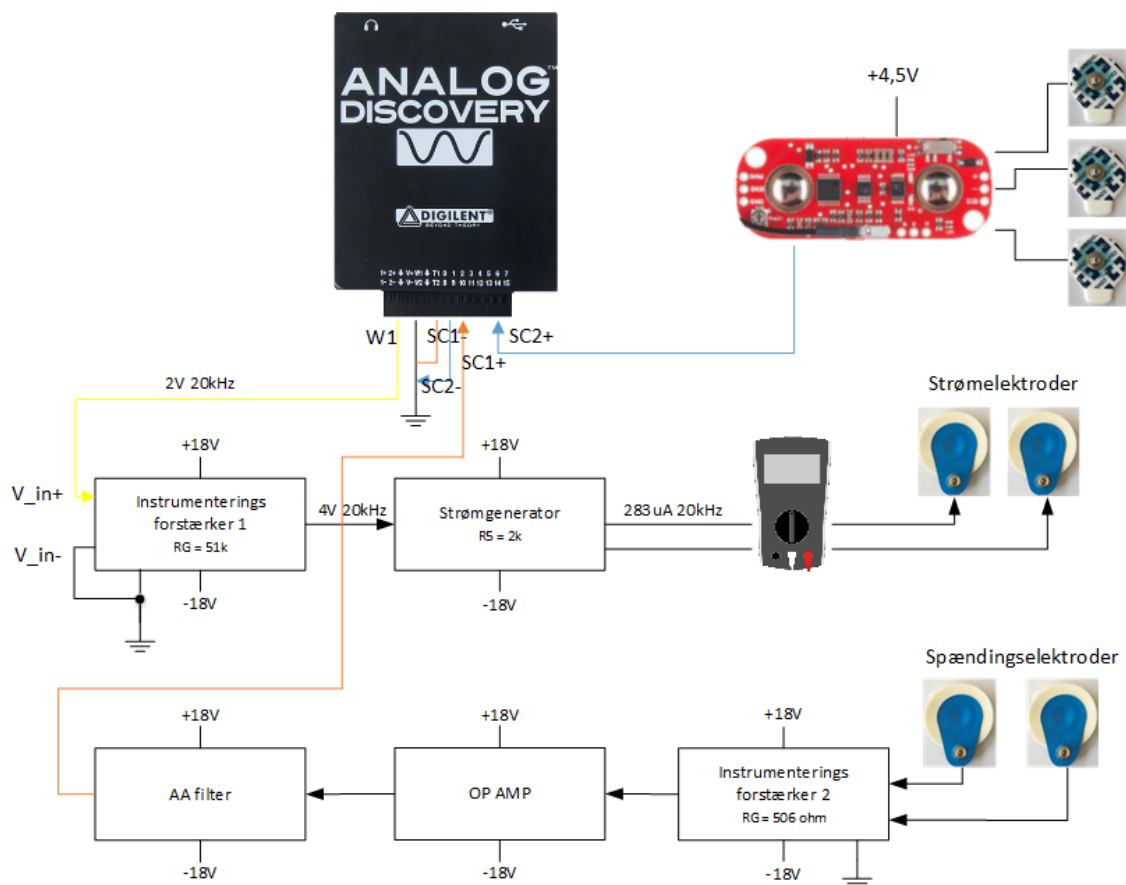


Figur 3.37



Figur 3.38

# Integrationstest 4



Figur 4.1

# Accepttest 5

---

[2]

# Litteratur

---

- [1] Christopher J Chester. Electrical-Impedance Biofeedback Instrument for Swallowing Rehabilitation. 2014.
- [2] Kevin R. Aroom, Matthew T. Harting, Charles S. Cox, Ravi S. Radharkrishnan, Carter Smith, and Brijesh S. Gill. Bioimpedance Analysis: A Guide to Simple Design and Implementation. *Journal of Surgical Research*, 153(1):23–30, 2009.