



# Sensorer og Instrumentering

## Systemoppsett

### Labrapport 1

### Gruppe 49

Sander Bakke  
544012

Kolbjørn Bølgen  
564466

Vår 2024

## Innhold

<b>1</b>	<b>Innledning</b>	<b>1</b>
<b>2</b>	<b>Teori</b>	<b>2</b>
2.1	Grunnmodell for sensorsystem . . . . .	2
2.2	Analog-til-Digital Omvandler (ADC) . . . . .	2
2.3	Støy demping . . . . .	4
2.4	Direct Memory Access (DMA) . . . . .	5
2.5	Digital Signalbehandling . . . . .	6
<b>3</b>	<b>Eksperimentelt Oppsett</b>	<b>10</b>
3.1	Blokk- og kretsskjema . . . . .	10
3.2	Digital styring, dataoverføring og databehandling . . . . .	11
<b>4</b>	<b>Realisering, Test og Resultater</b>	<b>12</b>
<b>5</b>	<b>Diskusjon</b>	<b>18</b>
<b>6</b>	<b>Konklusjon</b>	<b>20</b>
	<b>Referanser</b>	<b>21</b>
<b>A</b>	<b>Vedlegg</b>	<b>22</b>
A.1	Github . . . . .	22
A.2	Utstyrsliste . . . . .	22
A.3	Programvare . . . . .	22
A.4	Ekstra Figurer/Grafer . . . . .	23

---

---

## Forord

Vi ønsker å rette en spesiell takk til Instituttet for Elektroniske Systemer ved NTNU for deres generøse støtte gjennom utlån av Raspberry Pi, samt den grundige opplæringen og tilgjengeliggjøringen av nødvendig utstyr. Videre uttrykker vi vår dype takknemlighet overfor Lånekassen, som har gjort det mulig å finansiere anskaffelsen av andre essensielle komponenter og måleutstyr for gjennomføringen av dette prosjektet.

Vi ønsker også å rette en hjertelig takk til professorer, studentassistenter og øvrige ansatte ved Institutt for Elektroniske Systemer for deres uvurderlige støtte og veiledning gjennom hele prosessen med å utarbeide denne labrapporten. Uten deres ekspertise, dedikasjon og tålmodighet ville denne rapporten ikke ha vært mulig å fullføre. Våre medstudenter fortjener også anerkjennelse for deres oppmuntrende ord og konstruktive tilbakemeldinger som har bidratt til å forbedre vårt arbeid.

I tillegg ønsker vi å uttrykke vår takknemlighet overfor våre familier og venner for deres kontinuerlige støtte og forståelse gjennom den utfordrende studieperioden. Til slutt vil vi takke alle andre som har bidratt indirekte til dette prosjektet gjennom deling av kunnskap, inspirerende samtaler og verdifulle diskusjoner. Deres bidrag har medført både faglig og personlig vekst.

---

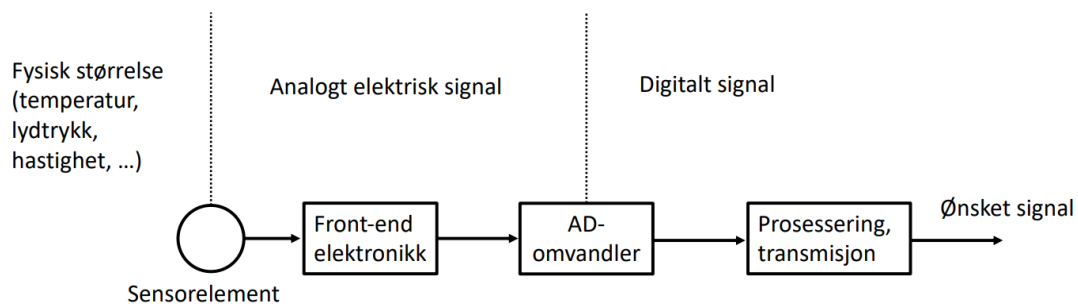
## Sammendrag

Denne rapporten beskriver design og testing av et målesystem, der hensikten er å sample analoge signaler for å analysere og prosessere dem digitalt. Systemet tar i bruk fem ADC-er til konvertering av analoge signaler til digitale. En Raspberry Pi blir brukt med SSH fra Matlab for å sample data fra ADC-ene, for så å prosessere dem. Målet med systemet er å få konvertert og sample data med så lite støy som mulig. Systemet er testet med et sinussignal med frekvens  $f = 1\text{kHz}$  og 1V amplitude på inngangen til ADC-ene. Det digitaliserte og samplede signalet hadde et signal-til-støy forhold på ca.  $56.6 \pm 0.2\text{ dB}$  fra fire av ADC-ene, mens den siste hadde på 42.4 dB, med maks oppnåelig på 74.0 dB. Det er også testet med og sammenliknet for to ulike vindusfunksjoner og zeropadding. Resultatene tyder på at målesystemet fungerer som forventet.

## 1 Innledning

I alt av moderne teknologi er det behov for å gjøre målinger av fysiske størrelser. En bil som måler avstand til omgivelsene, mobiler som skal vite hvilken vei du holder mobilen, smarte hjem som stiller inn temperaturen i rommet automatisk, medisinsk utstyr som måler alt fra hjerterefrekvens til oksygenmetning, og satellitter for orientering, er alle eksempler på teknologi som krever sensorer og behandling av data.

Det er selvsagt ulikt hvor krevende og omfattende disse systemene er, og i denne rapporten tar for seg et enklere målesystem, hvor hovedhensikten er å sample analoge signaler, og prosessere dem digitalt. Systemet skal baseres på en Raspberry Pi 3B single-board computer som prosesseringsenhet, som får digitale signaler fra fem eksterne analog-til-digital omvandlere (ADC) som sampler hvert sitt analoge signal. Målet for prosjektet er å utvikle et stabilt og fleksibelt målesystem, som kan brukes til å hente og prosessere data fra ulike typer sensorer. Videre er det ønskelig å oppnå et system mest mulig resistent mot støykomponenter og feil, som gir et tilstrekkelig presist og nøyaktig utgangssignal. Et blokkdiagram for et slikt grunnleggende system er vist i Figur 1.

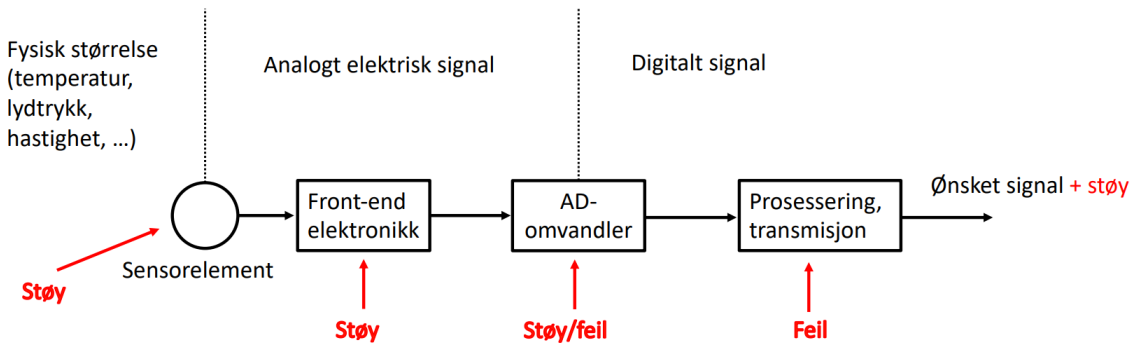


**Figur 1:** Grunnleggende blokkdiagram for målesystem med bruk av sensor. Tatt fra [1][s.2].

## 2 Teori

### 2.1 Grunnmodell for sensorsystem

Figur 2 illustrerer et blokkdiagram som beskriver en grunnleggende modell for et målesystem med introdusert støy.



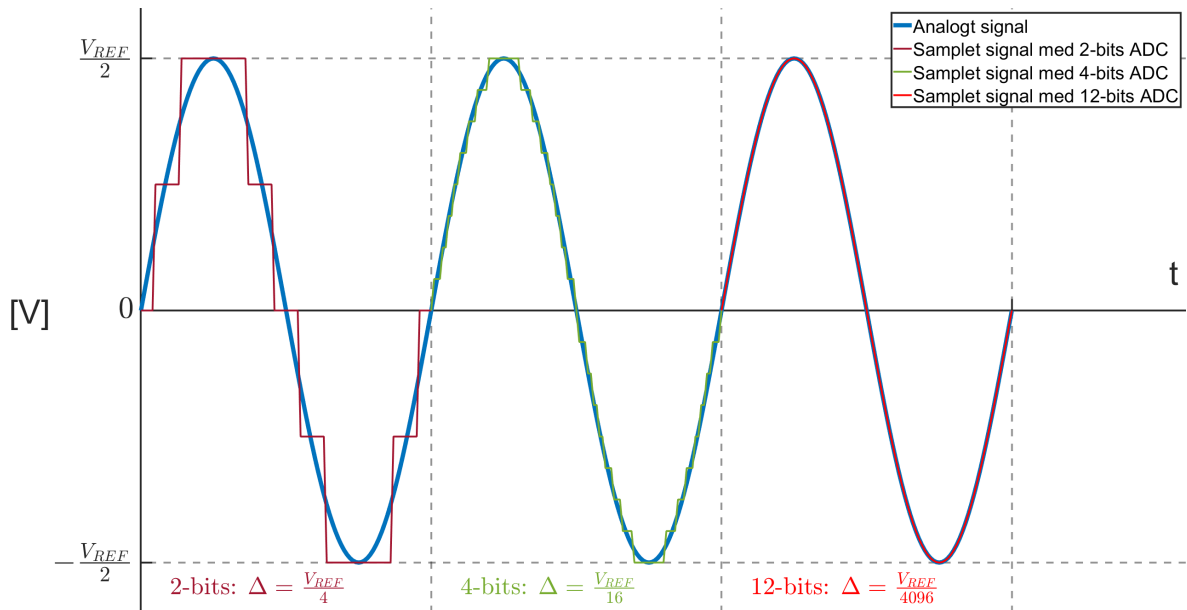
**Figur 2:** Grunnleggende blokkdiagram for målesystem med støy. Tatt fra[1][s. 3].

I ethvert system vil det opptre noe form for støy. Den målte utgangsverdien vil være det ønskede signalet, pluss den totale støyen i systemet. Totale støy er en sum av støy fra alle komponenter av systemet som er utsatt for støy. I et målesystem som det illustrert i figur 2 vil *sensorelementet* måle en gitt fysisk størrelse og gjøre dette til et elektrisk analogt signal. Via denne målingen vil det gjerne komme med en form for fysisk støy i det analoge signalet. Videre vil det den fysiske *front-end elektronikken* også være sensitiv for støy som kommer fra alle fysiske komponenter eller fra en ustabil spenningskilde. I Analog-til-Digital omvandler vil det analoge signalet bli samplet og omgjort til et digitalt signal, her kan det, som vist i figuren, opptre både støy i signalet ved sampling, og feil, såkalte *kvantiseringsfeil/-støy* (forklart i seksjon 2.2), ved prosessering av det analoge signalet. Det digitale signalet vil så bli samplet av en prosesseringsenhet, gjerne en datamaskin, som prosesserer det digitale signalet og gjør dataen lesbar. Her kan det også oppstå kvantiseringsfeil/-støy.

### 2.2 Analog-til-Digital Omvandler (ADC)

En  $N$ -bits ADC,  $N \in \mathbb{N}$ , som har en referansespenning  $V_{REF}$  vil ha en oppløsning,  $\Delta$ , som bestemmes av spenningsnivået ved den minst betydningsfulle biten (LSB),  $V_{LSB}$ , som beskrevet i likning (1), basert på [2][s. 17, Eq. 4-1], og illustrert i Figur 3.

$$\Delta = V_{LSB} = \frac{V_{REF}}{2^N} \quad (1)$$



**Figur 3:** Kvantisering i en 2-, 4- og 12-bits ADC. Matlab simulering *Kvantisering.mlx* i Vedlegg A.1.

Kvantiseringsstøy,  $\epsilon \in \mathbf{U}[-\frac{\Delta}{2}, \frac{\Delta}{2}]$ , er et resultat av avrundingsfeil som skjer ved digitalisering av analoge signaler, og vil ha en *rms*-verdi (root-mean-square) som er lik standardavviket til støyen, som beskrevet i likning (2), [1][s. 80-81].

$$\epsilon_{rms} = \sigma_{\epsilon} = \frac{\Delta}{\sqrt{12}} \quad (2)$$

Ved kvantisering av et sinussignal, vil den maksimale rms-verdien,  $\epsilon_{rms,signal} = \frac{\Delta \cdot 2^N}{2\sqrt{2}}$ , medføre en maksimal signal-støy-forhold  $\text{SNR}_{\text{maks}}$ , som beskrevet i likning (3), når det er ingen andre støykilder, [1][s. 81].

$$\text{SNR}_{\text{maks}} = 20 \cdot \log\left(\frac{\epsilon_{rms,signal}}{\epsilon_{rms}}\right) [\text{dB}] = 20 \cdot \log\left(\frac{\sqrt{12} \cdot (2^N \cdot \Delta)}{2\sqrt{2} \cdot \Delta}\right) [\text{dB}] \approx (1.76 + N \cdot 6.02) [\text{dB}] \quad (3)$$

Videre kan vi beregne spenningen til hvert sample,  $V_{sample}$ , ved å multiplisere dens digitale verdi  $N_{dig}$  med  $V_{LSB}$ , som beskrevet i likning (4), basert på [2][s. 17, Eq. 4-2].

$$V_{sample} = N_{dig} \cdot V_{LSB} = N_{dig} \cdot \frac{V_{REF}}{2^N}, \quad N_{dig} \in \mathbb{Z}, 0 \leq N_{dig} \leq 2^N - 1 \quad (4)$$

Likning (4) kan også utvides til tilfeller der  $N_{dig}$  og  $V_{sample}$  representerer vektorer med digitale verdier for alle samplene og deres fysiske spenning.

Samplingsfrekvensen,  $f_s$ , til en ADC avhenger av klokkefrekvensen,  $f_{CLK}$ , konverteringsraten,  $t_{conv}$ , og den analoge input samplingsraten,  $t_{sample}$ , som beskrevet i likning (5), hvor konverterings- og samplingsraten er gitt i antall klokkesykluser, basert på [2][s. 5, Fig. 1-1].

$$f_s = \frac{f_{CLK}}{t_{conv} + t_{sample}} \quad (5)$$

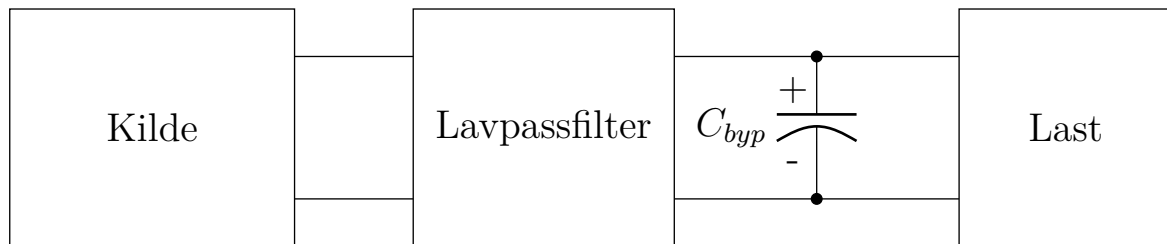
## 2.3 Støy damping

### Decoupling og Bypassing

For å isolere to sammenkoblede kretser og forhindre støyoverføring mellom dem kan *decoupling* implementeres. Dette realiseres hovedsaklig ved bruk av et retningsavhengig lavpassfilter [3].

*Bypassing* blir brukt for å redusere høyfrekvent støy i strømlinjene, dette oppnås ved å implementere en bypass-kondensator  $C_{byp}$ , som gir en alternativ vei til jord for dette høyfrekvente støyet [3].

Et støyreduserende system som implementerer decoupling og bypassing, er illustrert i Figur 4.

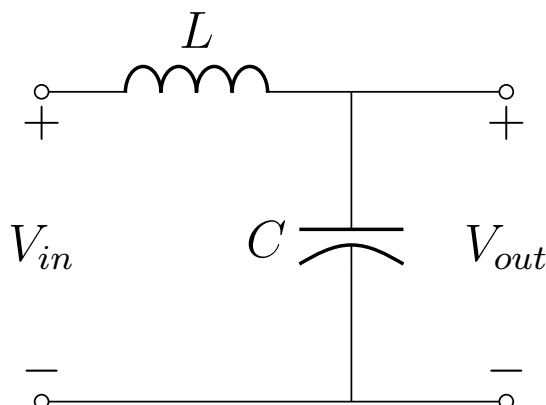


**Figur 4:** Støyreduert strøm- og spenningsforsyning, basert på lignende figur fra [3].

### Lavpassfilter

Det finnes flere mulige implementasjoner av lavpassfiltre, med ulike komponentkrav, og derav ulike resultater. F.eks. kan kaskadekobling av aktive filtre gi en brattere knekk i båndet. Dette målesystemet har ikke behov for et *ideelt* filter, altså er ikke brattest mulig knekk det viktigste, da filteret kun skal benyttes til decoupling.

En enkel potensiell implementasjon av et retningsavhengig lavpassfilter i et decoupling nettverk er en LC-krets. Et slik filter oppnås ved å benytte en spole med induktans  $L$  og en bipolar kondensator med kapasitans  $C$ , som illustrert i Figur 5, der  $V_{in}$  representerer inngangssignalet og  $V_{out}$  er utgangssignalet.



**Figur 5:** L-seksjon lavpassfilter

Lavpassfilteret i Figur 5 har en frekvensrespons  $H(\omega)$  og en amplituderrespons  $|H(\omega)|$ , begge avhengige av impedansen til spolen  $Z_L = j\omega L$  og impedansen til kondensatoren  $Z_C = \frac{1}{j\omega C}$ , der  $j = \sqrt{-1}$ , vinkelfrekvensen  $\omega = 2\pi f$  og  $f$  er frekvensen til inngangssignalet målt i hertz. Disse responsene er beskrevet i likning (6) og (7). Knekkfrekvensen,  $f_c$ , til systemet er gitt ved  $|H(f)| = \frac{1}{\sqrt{2}}$ , eller hvor amplituderresponsen er redusert med 3 dB fra sitt maksimum, noe som gir opphav til likning (8).

$$H(\omega) = \frac{V_{out}}{V_{in}} = \frac{Z_C}{Z_C + Z_L} = \frac{1}{1 - \omega^2 LC} \quad (6)$$

$$|H(\omega)| = \frac{1}{\sqrt{1 + (\omega^2 LC)^2}} \quad (7)$$

$$f_c = \frac{1}{2\pi\sqrt{LC}} \quad (8)$$

## 2.4 Direct Memory Access (DMA)

Det er ofte vanskelig å vite hvor ofte og når en sample/output er tatt/sendt når en bruker ADC-er, generelt vanskelig ved bruk av høynivå OS i prosesseringsenheten.

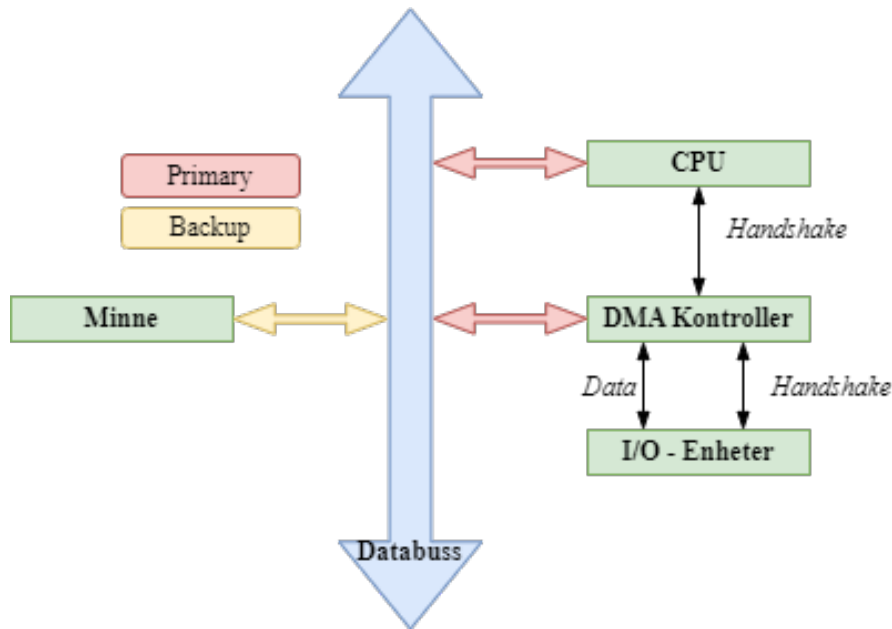
I denne laben, er det spesifisert bruk av Raspberry Pi til prosessering, som benytter Linux OS. Dette OS-et benytter *scheduler*, eller planlegger, som gir hver kjørende prosess på systemet et tidsvindu i køen. Dette vil gjøre at det kan oppstå forsinkelser på opptil på flere millisekunder når det er mye å gjøre [4][s.12]. Dette kan medføre problemer/feil når vi ønsker å sample live datastrømmer fra ADC-ene, fra f.eks. en sensor.

*Direct Memory Access* (heretter DMA) er en teknologi som tillater maskinvareenheten å hente data direkte fra systemets minne, uten å involvere CPU-en. Hardware modulene som ligger i ytterkanten av CPU-en, kan sample og kontrollere I/O enhetene uten avbrytelser fra CPU-en. Slik kan data fra pinnene bli kontinuerlig lagret direkte i minnet i en ringbuffer [4][s.12].



Kontrollprogrammet på CPU-en vil kun kjøre i korte tidssnutter om gangen for å hente ut data fra bufferen, før ny sample-data overskriver den eksisterende. Slik trenger vi kun CPU-en i korte tidssnutter om gangen med jevne mellomrom, uten å bli avbrutt [4][s.13]

Figur 6 viser hvordan DMA blir en slags “mellommann” mellom CPU-en og I/O-enhetene til datamaskinen



**Figur 6:** Signalebuss med DMA, basert på lignende figur fra [5].

## 2.5 Digital Signalbehandling

### Fast Fourier Transform (FFT)

Fourier-transform brukes for å omgjøre et signal fra tidsdomenet til frekvensdomenet, slik at man kan identifiseresignalets frekvenskomponenter og deres amplituder.

FFT kjennetegnes av sin raske prosesseringstid, effektivitet, og allsidighet, som gjør den gunstig for implementering i sanntids- og sensorsystemer, samtidig som den kan brukes til å analysere både periodiske og ikke-periodiske signaler. Antallet beregninger ved bruk av en Radix-2 FFT-algoritme sammenlignet med DTFT (Discrete-Time Fourier Transform) er som beskrevet i Tabell 1. Tilsvarende reduseres antallet beregninger drastisk for andre FFT-algoritmer sammenlignet med den direkte formen av diskret Fourier-transformasjon.

**Tabell 1:** Sammenligning av beregningskompleksitet mellom direkte og Radix-2 Fourier-transformasjoner.

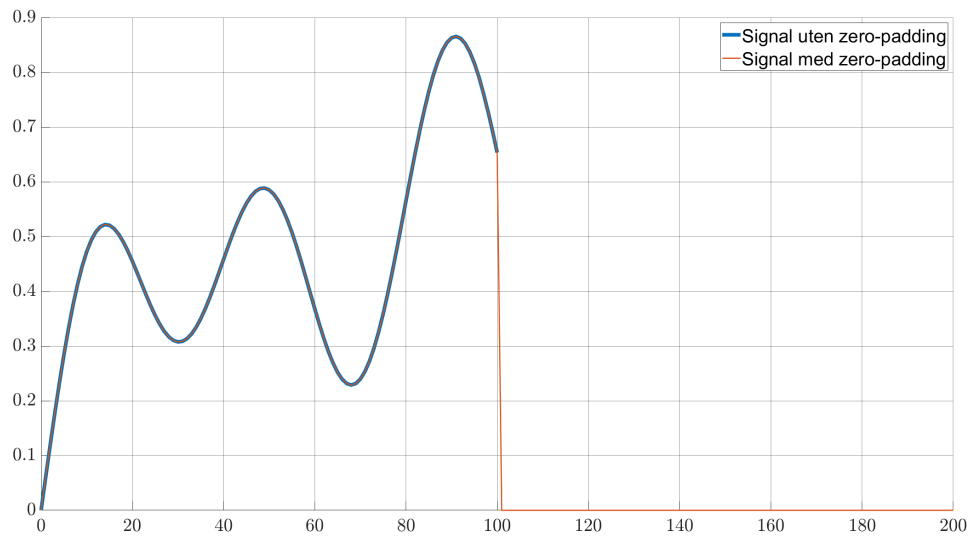
Antall punkter/sampler, $P$	Komplekse multiplikasjoner ved DTFT, $P^2$	Komplekse multiplikasjoner ved Radix-2 FFT, $\frac{P}{2}\log_2(P)$
$4=2^2$	16	4
$8=2^3$	64	12
$16=2^4$	256	32
$32=2^5$	1024	80
$64=2^6$	4096	192
$128=2^7$	16384	448
$256=2^8$	65536	1024
$512=2^9$	262144	2304
$32768=2^{15}$	1073741824	245760

Gitt et vilkårlig tidskontinuerlig eller diskret signal,  $x(t)$  eller  $x[n]$ , med Fourier-transform,  $X(f)$ , vil man ha en relativ effekttetthet,  $S(f)$ , som beskrevet i likning 9, hvor  $|X(f)|$  er amplituderesponsen til signalet og  $|X(f)|_{maks}$  er dens maksimum.

$$S(f) \text{ [dB]} = 20 \cdot \log \frac{|X(f)|}{|X(f)|_{maks}} \text{ [dB]} \quad (9)$$

### Zero-padding

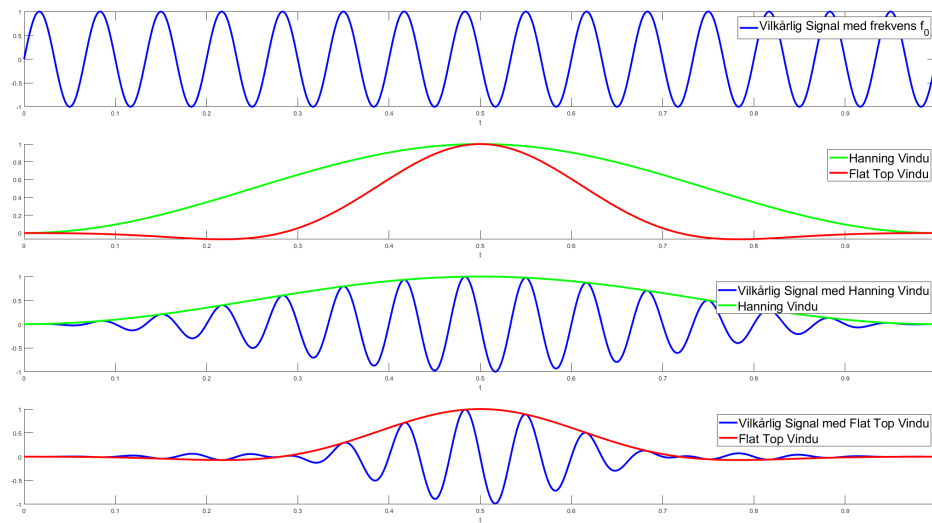
Zero-padding er en teknikk hvor man legger til nuller i slutten av en sekvens, for å øke den totale lengden på et digitalt signal, som illustrert i Figur 7. Zero-padding kan være gunstig fordi det øker antall punkter som analyseres av en Fourier-transformasjon, noe som resulterer en høyere oppløsning på frekvensaksen i en spektralanalyse. I tillegg vil Radix-2 FFT-algoritmen operere optimalt når antallet punkter i signalet er en potens av 2. Ved å legge til nuller i slutten av signalet for å oppnå dette, kan man dra nytte av effektiviteten til Radix-2 FFT-algoritmen uten behov for ytterligere prosessering.



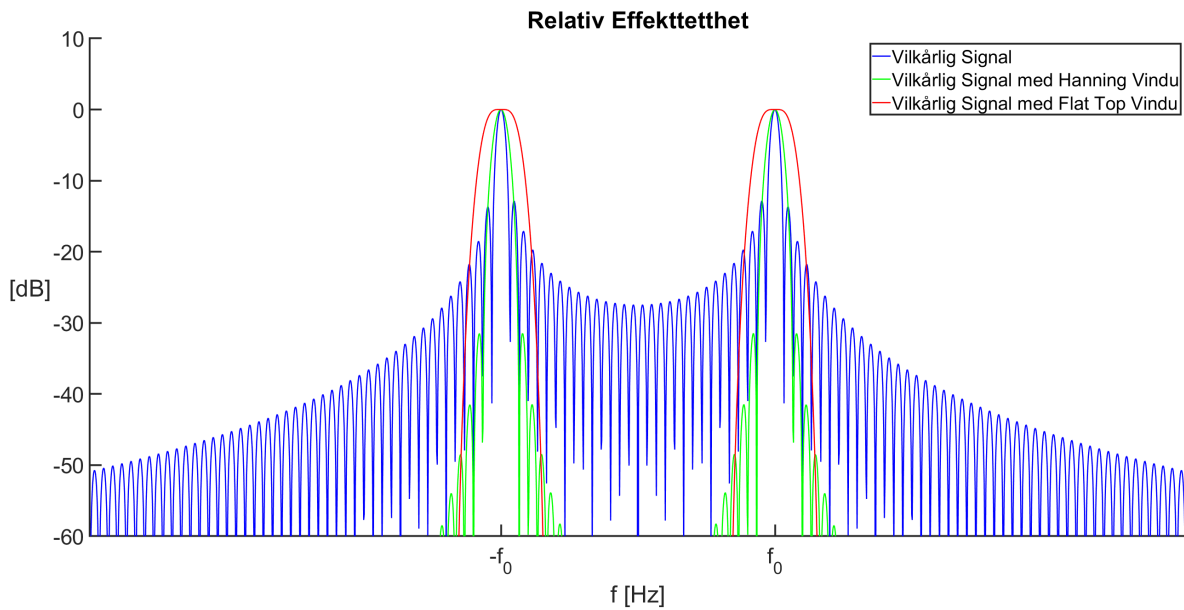
**Figur 7:** Vilkårlig signal, **med** og **uten** zero-padding. Matlab simulering *ZeroPadding.mlx* i Vedlegg A.1.

## Vindusfunksjon

Vindusfunksjoner brukes på signaler for å gi en glattere effekttetthets-kurve, samt gi tydeligere men bredere hovedlober, og lavere sidelober, noe som blir gjort ved å manipulere amplituden og fasen til signalet (illustrert i Figur 8), før den Fourier-transformeres (illustrert i Figur 9 som relativ effekttetthet).



**Figur 8:** Vilkårlig signal med Hanning- og Flat Top Vindu. Matlab simulering *Vindusfunksjoner.mlx* i Vedlegg A.1.

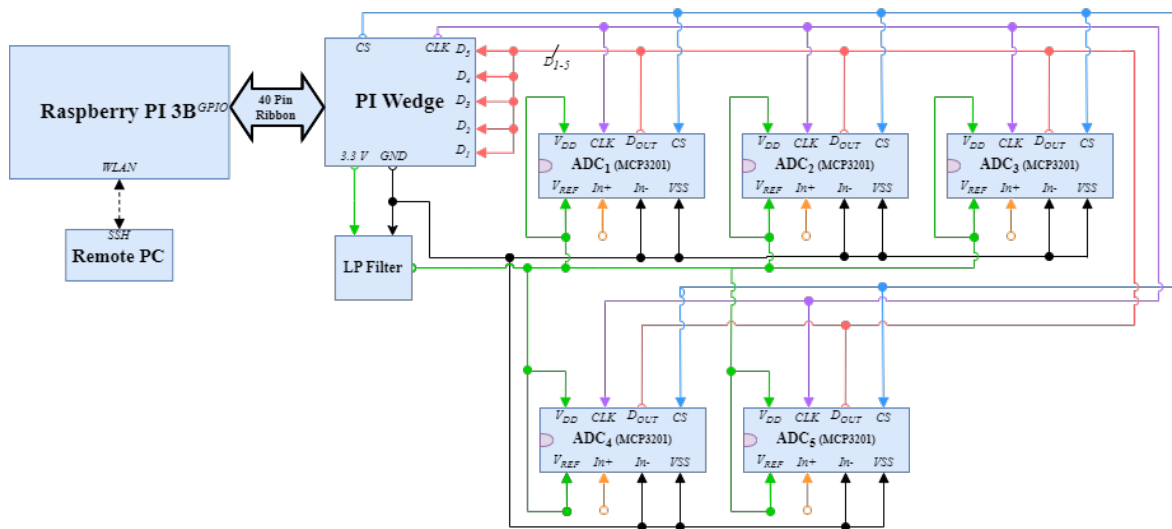


**Figur 9:** Relativ effekttetthet av et reellt vilkårlig signal med frekvens  $f_0$ , med Rektangulær vindusfunksjon (ingen vindusfunksjon), med Hanning vindusfunksjon og Flat Top vindusfunksjon. Matlab simulering *Vindusfunksjoner.mlx* i Vedlegg A.1.

### 3 Eksperimentelt Oppsett

#### 3.1 Blokk- og kretsskjema

En overordnet beskrivelse av funksjon og oppførsel til det totale systemet, er illustrert som et blokk-skjema i Figur 10. Blokk-skjema viser sammenhengen mellom de ulike delsystemene navngitt i Figur 2 med signalveier og kommunikasjonsporter. Fra Figur 10 kan en se at det ikke er inkludert noen sensorer, da det i dette prosjektet ikke skal måles noen fysisk data, men kun teste oppførselen til målesystemet.



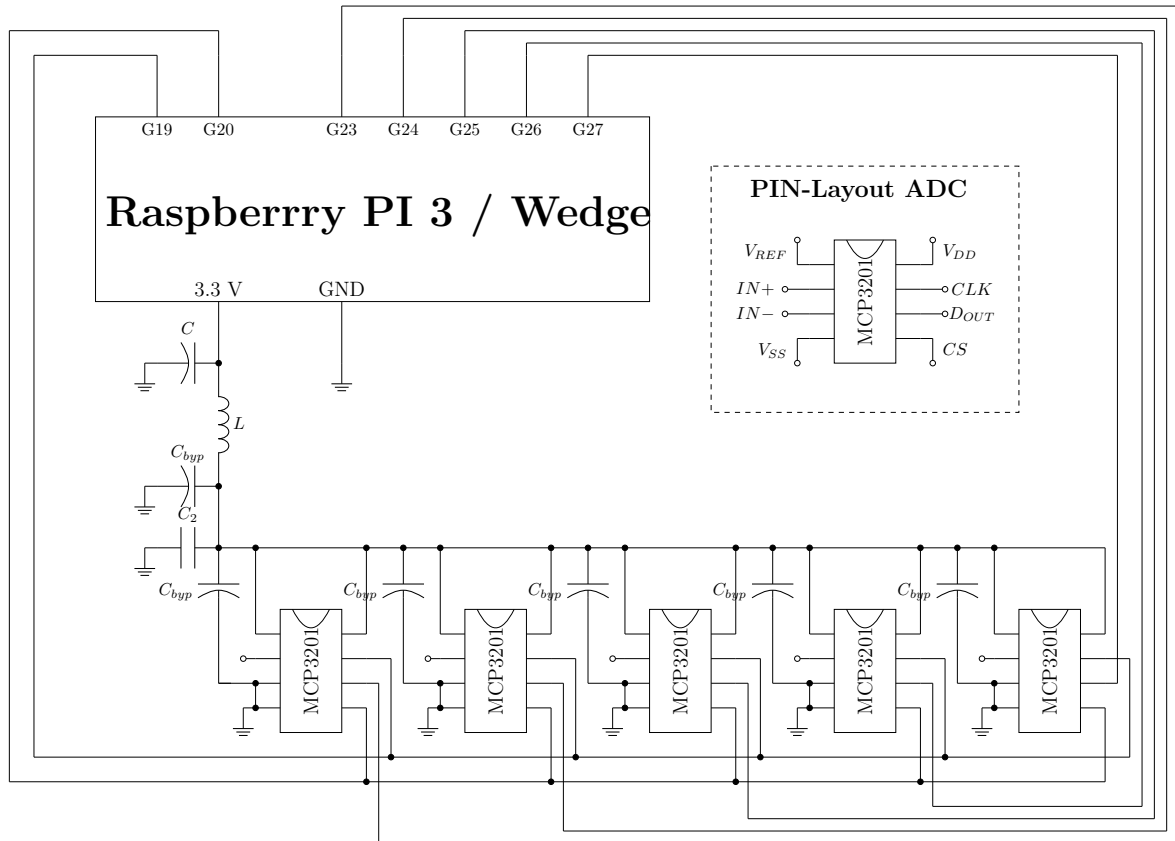
**Figur 10:** Blokk-skjema av målesystem.

I blokk-skjema representerer *Raspberry Pi 3B* (heretter RPi) og *Remote PC* prosesseringsenheten i Figur 2,  $ADC_{1-5}$  tilsvare AD-omformere, og *Pi Wedge*, *LP Filter* og alle ledninger tilsvare den fysiske front-end elektronikken.

Fra figur 10 kan en se at RPi-en leverer forsyningsspennig 3.3V til ADC-enes porter  $V_{DD}$  og  $V_{REF}$ , og jord  $GND$  til ADC-enes  $V_{SS}$ . Den sender også de digitale signalene  $CS$  (*chip-select/styringssignal*) og  $CLK$  (klokkesignalet) til de tilsvarende portene hos ADC-ene. Disse signalene sendes via en *Pi Wedge*, et breakout-board koblet til RPi-ens GPIO pins. ADC-enes pinner  $In+$  og  $In-$  er input pinnene for det analoge signalet som skal digitaliseres. I dette blokk-skjema er  $In-$  jordet, da vi ønsker nullnivå på 0V. Ouputet kommer fra  $D_{out}$  som går tilbake til RPi-ens/Wedgens GPIO-pinner.

Kretsskjemaet, som illustrert i Figur 11, vil dele noen likheter med blokk-skjemaet. Kretsskjemaet inkluderer ikke “Remote PC” (ingen fysisk kobling), men vil i motsetning til blokk-skjema inkludere bypass-kondensatorer  $C_{byp}$  og et decoupling-nettverk (LP-filteret) med komponentene sine  $C$  og  $L$ , som beskrevet i seksjon 2.3, samt en keramisk kondensator  $C_2$ , benyttet for ytterligere støydemping. I tillegg er koblingsnavnene på datalinjene endret for å samsvare med GPIO-pinnene som brukes i kildekode (*adc\_sampler.c*, se Vedlegg A.1):

- $CLK \rightarrow G19$
- $CS \rightarrow G20$
- $D_{1-5} \rightarrow G23-G27$



**Figur 11:** Kretsskjema av støyreduert målesystem med 5 analog-digital omformere.

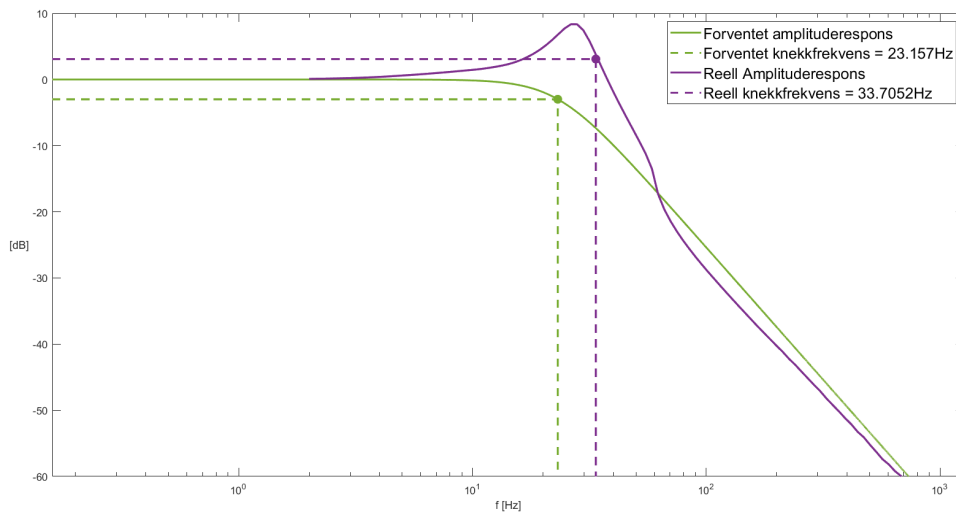
### 3.2 Digital styring, dataoverføring og databehandling

Ved digital styring av RPi-en benyttes SSH (Secure Shell). Til dataoverføring mellom RPi-en og “Remote PC” (se Figur 10) benyttes SFTP (SSH File Transfer Protocol) for sikker og pålitelig kommunikasjon.

Dataoverføring mellom ADC-ene og RPi-en brukes SPI-protokoll og DMA, som beskrevet i seksjon 2.4. DMA oppnås ved å laste ned og installere/unzip *master.zip* (C-biblioteket pi-gpio, se Vedlegg A.1) på RPi-en. Deretter, for å sample ADC-ene, kjøres C-koden *adc\_sampler.c* (se Vedlegg A.1), hvor de digitale verdiene lagres i en binær fil. Denne filen overføres deretter til en PC for videre digital signalbehandling.

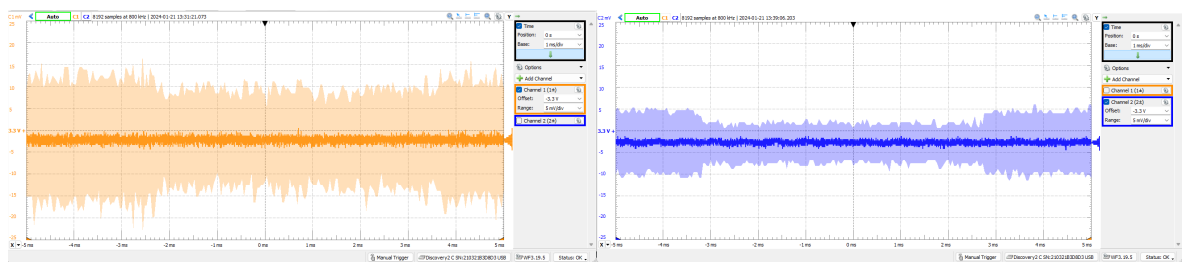
## 4 Realisering, Test og Resultater

Lavpassfilteret, illustrert i Figur 5, som er brukt som decoupling nettverk implementert i Figur 10 og 11, realiseres ved bruk av en kondensator  $C = 470\mu\text{F}$  og en spole  $L \approx 100\text{mH}$ . Fra likning (8), får man en forventet knekkfrekvens,  $f_c = 23.2\text{Hz}$ . Både forventet og reell amplituderrespons og knekkfrekvens er vist i Figur 12. Figuren viser at oppnådd knekkfrekvens er på  $33.7\text{Hz}$ , som gir et avvik på  $9.5\text{Hz}$ .



**Figur 12:** Forventet og reell amplituderrespons og knekkfrekvens, av decoupling nettverket (lavpass filteret). Matlab simulering *Lavpass.mlx* og data *Lavpass.csv* i Vedlegg A.1.

Likespenningen blir også målt før og etter decoupling nettverket implementeres. Resultater er vist i Figur 13. I Figur 13a er støynivået sentrert rundt  $3.3\text{ V}$  med et spenn på cirka  $\pm 15\text{mV}$ , og i Figur 13b er det et mindre spenn på cirka  $\pm 5\text{mV}$ .

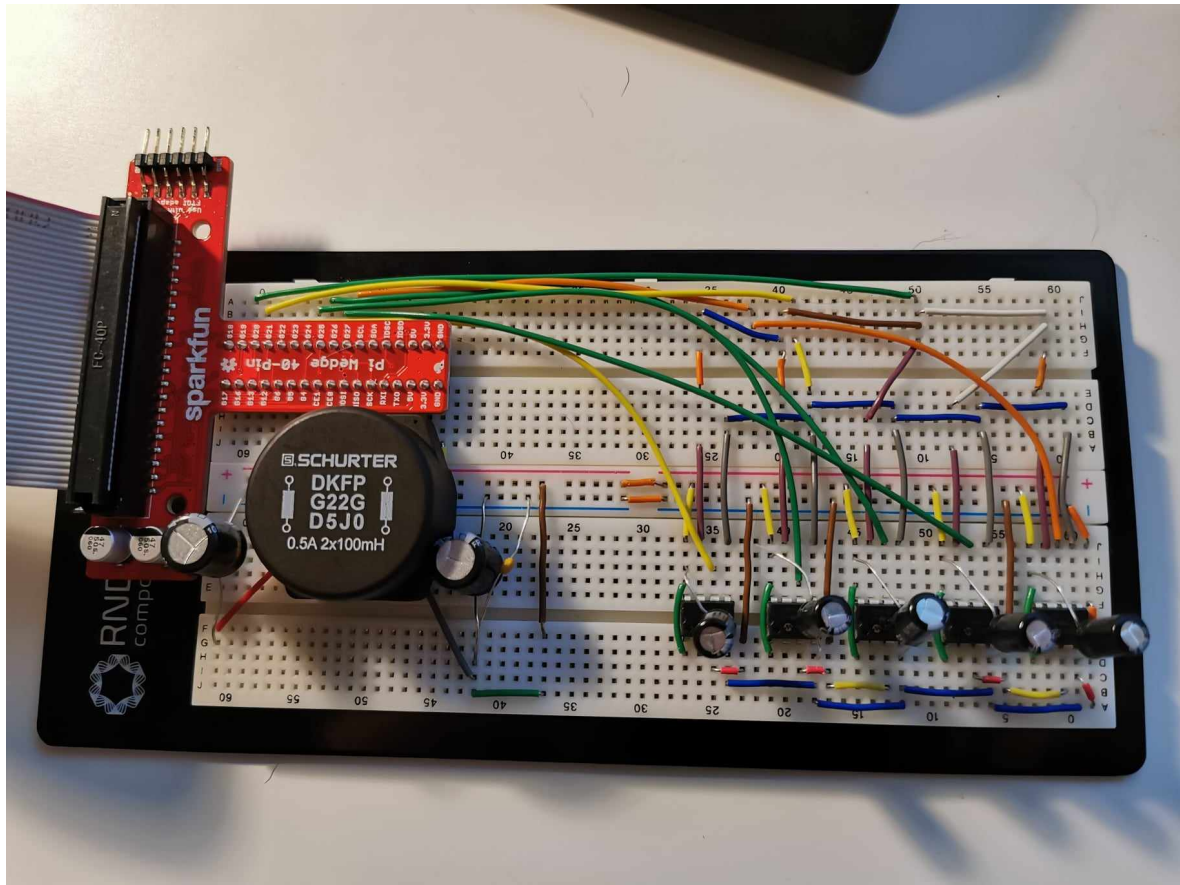


(a) Likespenning uten decoupling nettverk.

(b) Likespenning med decoupling nettverk.

**Figur 13:**  $3.3\text{ V}$  likespenning med og uten filtrering.

Videre implementeres resten av systemet i Figur 11 med decoupling nettverket, bypass kondensatorer  $C_{byp} = 100\mu\text{F}$  og den keramisk kondensatoren  $C_2 = 100\text{nF}$ . Realisert oppkobling vist i Figur 14.



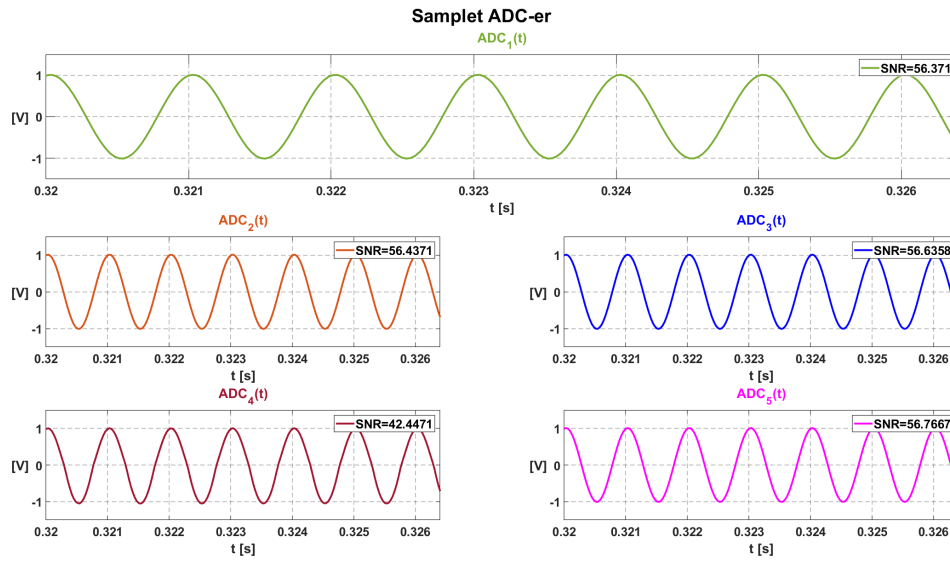
Figur 14: Oppkobling av målesystem.

Som diskutert i seksjon 3.2, blir RPi-en styrt fra en “Remote PC” vha. SSH. Denne SSH-tilkoblingen opprettes direkte fra Matlab for enkel digital styring og dataoverføring, (se de to øverste seksjonene i *readData.mlx* vedlagt i Vedlegg A.1). Her initialiseres sampling direkte fra Matlab og lagres i en binærfil, som deretter overføres til en brukerdefinert mappe på PC-en før den slettes fra Rpi-en. Denne binærfilen blir omgjort til matrise, hvor hver rad representeres de digitale samplingsverdiene gitt fra en av ADC-ene.

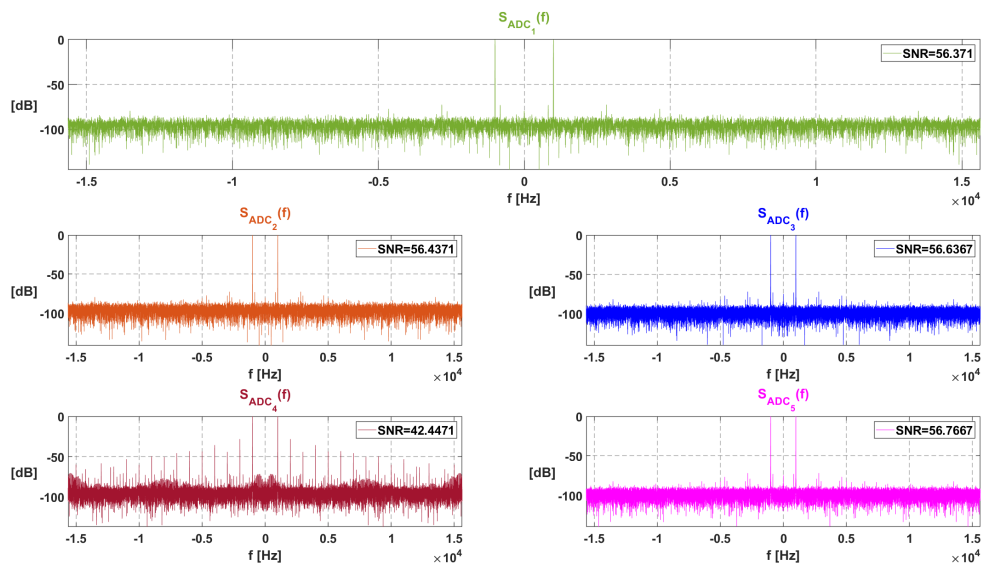
Målesystemet testes ved å sende et sinussignal med frekvens 1 kHz, amplitude 1V og off-sett 1.5V in på *In+* på alle ADC-ene, deretter samples ADC-ene og dataen lagres i filen *1kHzSine.bin* i Vedlegg A.1.

Videre prosesseres dataen i Matlab, (komplett kode *readData.mlx* i Vedlegg A.1), og plottes i sine respektive tidsserier,  $ADC_{1-5}(t)$ , illustrert i Figur 15, og medhørende frekvensspekter,  $S_{ADC_{1-5}}(f)$ , illustrert i Figur 16.





**Figur 15:** Tidsserie av samplet signaler.



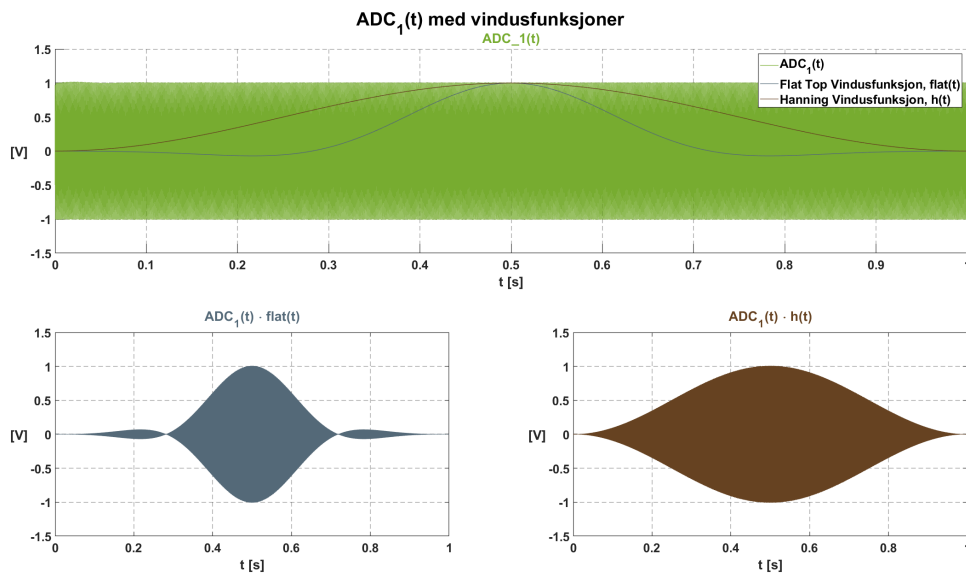
**Figur 16:** Frekvensspekter, i form av relativ effektetthet av samplet ADC-er,  $S_{ADC_{1-5}}(f)$ . Flere figurer av hvert frekvensspekter er vedlagt i Vedlegg A.4.

I Vedlegg A.4 ligger de samme frekvensspektrene som i Figur 16, bare med mindre frekvensbånd for å se signalets spekter nærmere, for enklere identifikasjon av støyfrekvenser.

I Matlab koden brukes blant annet likning (4) og (5) for å omgjøre samplene til sine fysiske spenningsnivåer ved sine respektive tidspunkt, samt at likespenningsnivået til signalene blir fjernet. Videre beregnes relativ effektetthet til tidsseriene, der Flat Top vindusfunksjonen

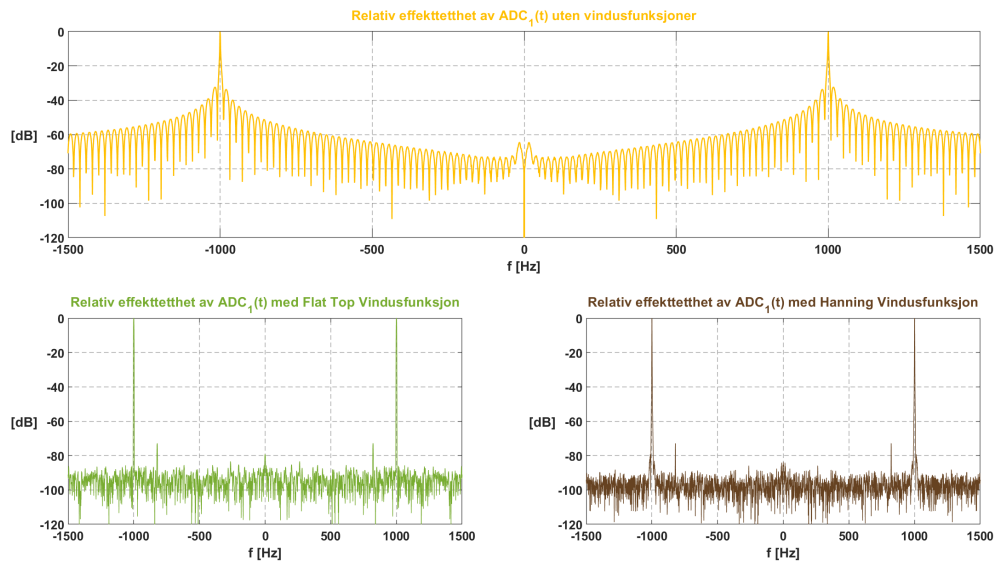
blir brukt på  $ADC_{1-2}(t)$  og Hanning vindusfunksjon blir brukt på  $ADC_{3-5}(t)$ , deretter blir alle zero-paddet til de får  $N_{fft} = 2^{15}$  sampler. Etterpå Fourier-transformeres signalene, og til slutt brukes likning (9), for å få den relative effekttettheten til signalene. Dette gir oss et mål på hvor mye effekt som er konsentrert innenfor et frekvensintervall, noe som gjør det lettere for oss å identifisere signalenes frekvenskomponenter samt eventuelle uønskede støykilder.

Figur 17 viser hvordan signalene i tidsdomene blir påvirket av vindusfunksjonene, der grønn er signalet  $ADC_1(t)$  uten vindusfunksjon på, grå med Flat Top vindusfunksjon på, og brun med Hanning vindusfunksjon.



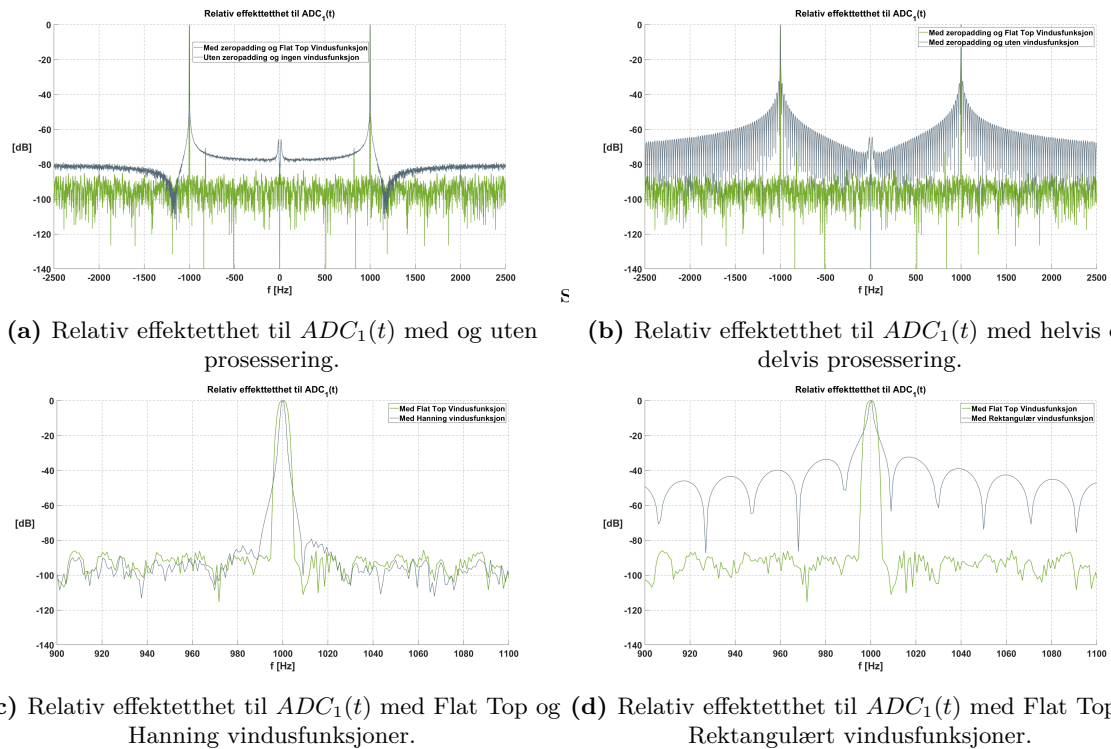
**Figur 17:** Helhetlig tidsserie av  $ADC_1(t)$  med ingen, Flat Top- og Hanning vindusfunksjon.  $flat(t)$  og  $h(t)$  er tidsseriene til de respektive vindusfunksjonene med samme lengde som  $ADC_1(t)$ .

Figur 18 viser hvordan den relative effekttettheten til signalet  $ADC_1(t)$  blir påvirket av vindusfunksjonene, der gul er uten vindusfunksjon, grønn med Flat Top vindusfunksjon, og brun med Hanning vindusfunksjon.



**Figur 18:** Sammenligning av effektetthetsspektrumet til  $ADC_1(t)$  ved bruk av forskjellige vindusfunksjoner.

For å tydeliggjøre funksjonaliteten til både vindusfunksjoner og zeropadding, er det i Figur 19 plottet den relative effektettheten til signalet  $ADC_1(t)$  med ulike vindusfunksjoner og zeropadding lagt på.



**Figur 19:** Funksjonalitet av vindusfunksjoner og zeropadding sett på den relative effektettheten til  $ADC_1(t)$

I Vedlegg A.4 ligger også relativ effektetthet til  $ADC_4(t)$  med og uten prosessering.

Ved bruk av likning (3) med  $N = 12$  det teoretiske maksimumet til hver av de 12-bits ADC-ene, og med den innebygde SNR funksjonen i Matlab får vi de målte SNR verdiene vist i Figur 15. Disse verdiene for de fem ADC-ene er vist i Tabell 2.

**Tabell 2:** Sammenligning av teoretisk maksimum og målte verdier av SNR, for hver ADC.

ADC #	Teoretisk $SNR_{maks}$ [dB]	Målt SNR [dB]
1	74.0	56.4
2	74.0	56.4
3	74.0	56.6
4	74.0	42.4
5	74.0	56.8

## 5 Diskusjon

Lavpassfilteret som er brukt som decoupling nettverk, med amplituderrespons vist i Figur 12, har en 9.5Hz høyere knekkfrekvens enn forventet. Dette skyldes mest sannsynlig den store toleransen til spolen som er brukt, som ifølge databladet er i området  $[-30\%, +50\%]$ . Ettersom at det kun skal forskyve en likespenning, vil ikke dette ha noen betydelig påvirkning på systemet, annet enn at litt flere støyfrekvenser vil unngå filtrering, og at støyet med frekvens litt mindre enn knekkfrekvensen vil få en liten forsterkning.

Fra Figur 13 kan vi tydelig observere en reduksjon av støynivået ved implementering av et decoupling nettverk. I figur 13a er støynivået sentrert rundt 3.3 V med et spenn på cirka  $\pm 15\text{mV}$ , mens i figur 13b observerer vi et mindre spenn på cirka  $\pm 5\text{mV}$ . Dette er som forventet og beskrevet i seksjon 2.3, hvor høyfrekvent støy blir filtrert effektivt. Med implementerte bypass kondensatorer kan vi forvente en ytterligere reduksjon av støynivået fra spenningskilden.

Fra Tabell 2 kan en se at det oppnådd SNR på ca.  $56.6 \pm 0.2$  dB for ADC  $\{1,2,3,5\}$ , mens ADC 4 hadde en SNR på 42.4 dB. Disse er henholdsvis  $17.4 \pm 0.2$  dB og 31.6 dB under maks oppnåelig SNR på 74.0 dB.

Fra både Figur 16 og Tabell 2, er det tydelig at det oppstår betydelig mer støy i den fjerde ADC-en sammenlignet med de andre. Ved nærmere inspeksjon av Figur 23 i Vedlegg A.4, kan vi tydelig identifisere interferenslinjer som oppstår ved de overharmoniske frekvensene til inngangssignalet. Dette indikerer en form for intern interferens, som mest sannsynlig skyldes kryssende eller lange ledninger i den fysiske oppkoblingen. En annen mulig årsak kan være elektromagnetisk interferens på grunn av dårlig skjermede komponenter, eller dårlig jording. En mulig løsning for å redusere disse effektene, kan være å bruke kortere og bedre skjermede kabler for å redusere den totale lengden på ledningene og samtidig minimere kryssende signaler. Videre kan god jordtilkobling og bruk av skjermede komponenter bidra til å minimere elektromagnetisk interferens.

Fra Figur 18 observerer vi at implementeringen av vindusfunksjoner resulterer i kraftig demping av sidelobene til signalet. Ved nærmere inspeksjon av Figurer 19a, 19d, og 19c, ser vi også at hovedloben blir bredere og glattere, i tråd med teoretiske forventninger. Sammenligningen av Flat Top- og Hanning vindusfunksjoner i Figur 19c viser at Flat Top har en bredere og jevnere hovedlobe. Dette antyder at bruken av Flat Top kan redusere spektral lekkasje og gi en mer nøyaktig amplitude representasjon. Imidlertid kommer dette også med en redusert frekvensoppløsning sammenlignet med Hanning vinduet. Rektangulært vindu gir ingen nytte.

Med utgangspunkt i den første ADC-en,  $ADC_1(t)$ , vil støygulvet på signalet ligge på omtrent -90 dB når det er prosessert (zero-padding og bruk av vindusfunksjon), og omtrent -80 dB når de ikke er prosessert, som illustrert i Figur 19a. Et støygulv på -90 dB vil vi anse som tilstrekkelig lavt nok for et sensorsystem som skal anvendes i våre fremtidige laboratorieprosjekter, bestående av sampling av mikrofoner og radar. Likevel har systemet vist seg å være sårbart for miljøer med høy elektromagnetisk interferens, og bør derfor forbedres ytterligere før det eventuelt tas i bruk i høy presisjons anvendelser som medisinsk teknologi eller romfart.

Systemet i seg selv er veldig enkelt å bruke, ettersom at all initialisering og behandling skjer direkte i Matlab. Dette gjør det godt egnet til bruk i laboratorieprosjekter, så lenge brukerne har en grunnleggende kjennskap til Matlab programmering og dataanalyse. Denne integrerte

tilnærmingen forenkler prosessen med datainnsamling og behandling, som igjen gjør det enkelt å tilpasse systemet til ulike eksperimentelle behov.

## 6 Konklusjon

Det er blitt realisert et målesystem basert på en grunnleggende sensormodell med ADC-er og RPi til bruk i senere prosjekter. Metoden brukt er beskrevet i seksjon 3, og målingene som ble gjort er vist i seksjon 4. Målingene viser at ADC-ene klarer å reprodusere et sinussignal med 1V amplitude til stor nøyaktighet. Videre viser de at ADC-ene {1,2,3,5} hadde en SNR på ca.  $56.6 \pm 0.2$  dB, mens ADC 4 hadde en SNR på 42.4 dB, som er henholdsvis  $17.4 \pm 0.2$  dB og 31.6 dB under maks oppnåelig SNR på 74.0 dB. All samhandling mellom RPi og ekstern PC, samt databehandling, er gjort i Matlab. Dette betyr målesystemet er stabilt og fleksibelt, og kan benyttes i senere prosjekter som krever et målesystem der en bør være oppmerksom på at ADC 4 er noe mer utsatt for støy.

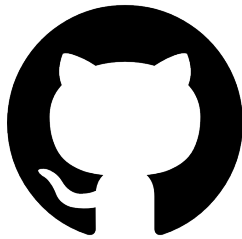
## Referanser

- [1] Svensson, U. P. & Eide, E. & Randeberg, L. L., *TTT4280 Sensorer og instrumentering - Kompendium*, Institutt for elektroniske systemer - NTNU, 2023.
- [2] Microchip Technology, *MCP3201 Datasheet*, 31.01.2024.
- [3] Kundert, K., *Power Supply Noise Reduction*, Designer's Guide Consulting, Inc., Versjon 4, Januar 2004.
- [4] Faculty of Information Technology and Electrical Engineering, *User Guide for the Laboratory TTT4280 - Sensors and Instrumentation Spring 2024*, Institutt for elektroniske systemer - NTNU, 08.01.2024.
- [5] Open4Tech, *Direct Memory Access (DMA) in Embedded Systems*, 03.02.2024.



## A Vedlegg

### A.1 Github



Matlab kode, kilde kode og data.

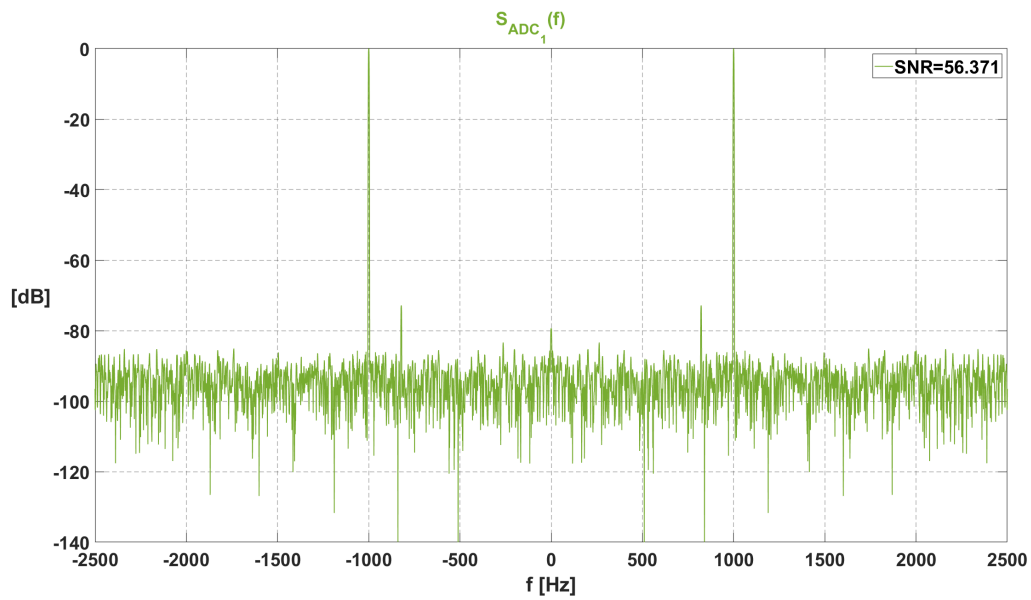
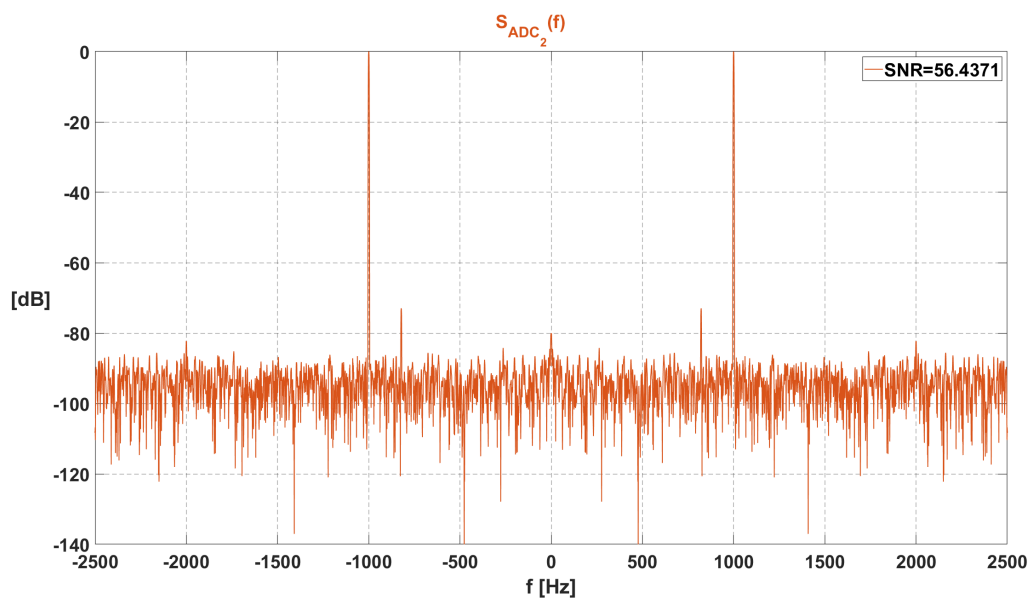
### A.2 Utstysrliste

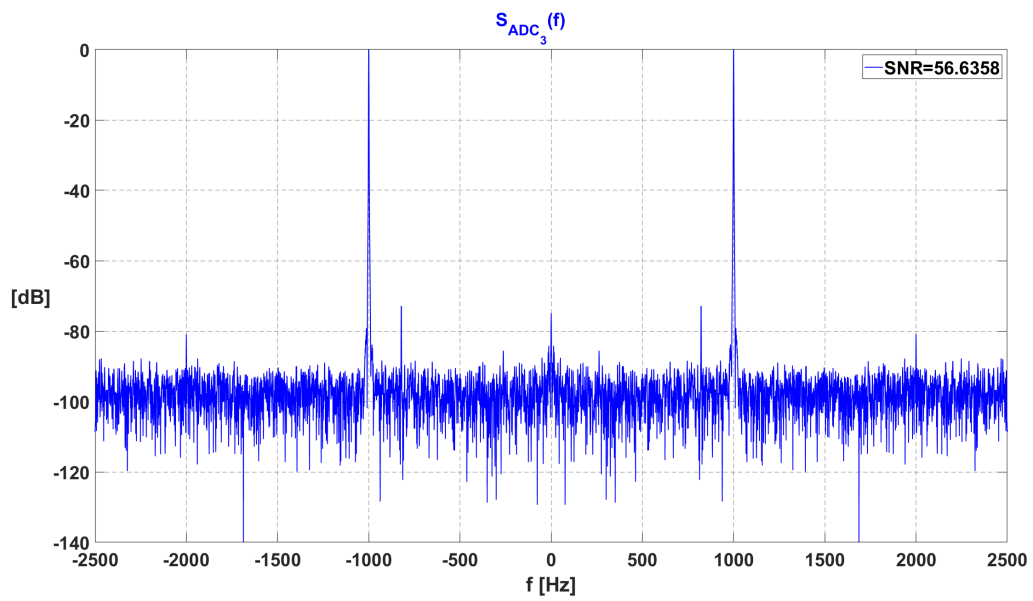
- PC
- Raspberry Pi 3B
- Raspberry Pi Wedge
- Analog Discovery
- Kondensatorer:  $470\mu\text{F}$ ,  $100\mu\text{F}$ ,  $100\text{nF}$
- Spole  $\approx 100\text{mH}$
- Analog-til-digital omformere (MCP3201)

### A.3 Programvare

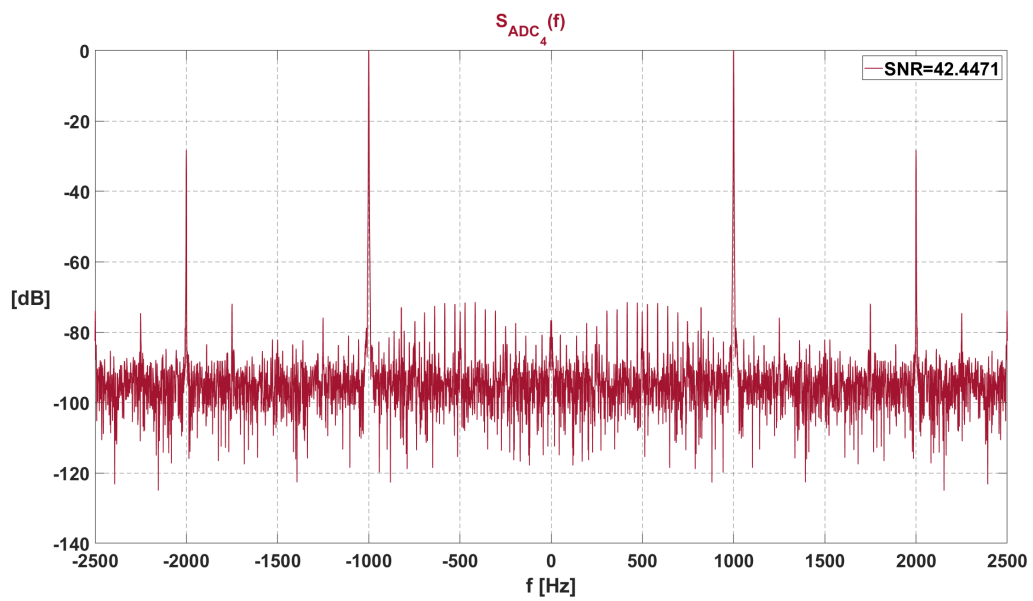
- Matlab
- Waveforms
- PuTTY

## A.4 Ekstra Figurer/Grafer

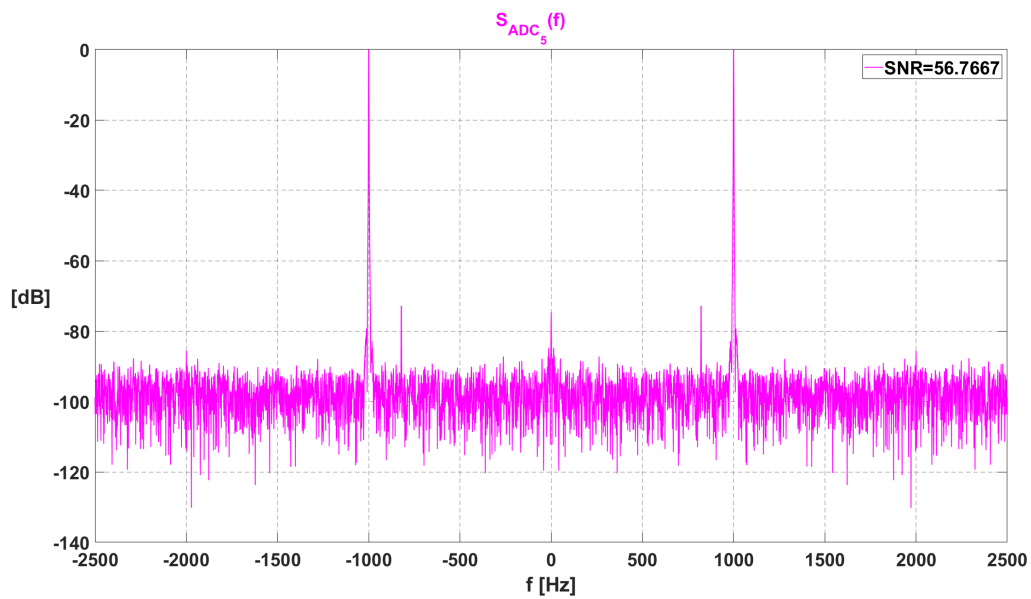
Figur 20: Relativ effektetthet,  $S_{ADC_1}(f)$ .Figur 21: Relativ effektetthet,  $S_{ADC_2}(f)$ .



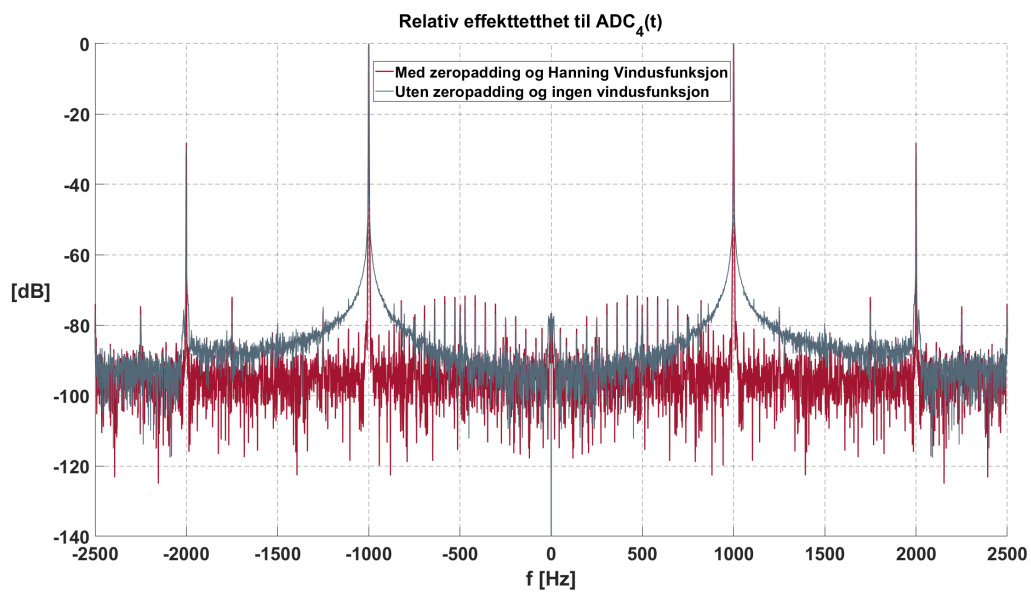
**Figur 22:** Relativ effektetthet,  $S_{ADC_3}(f)$ .



**Figur 23:** Relativ effektetthet,  $S_{ADC_4}(f)$ .



Figur 24: Relativ effekttetthet,  $S_{ADC_5}(f)$ .



Figur 25: Relativ effekttetthet til  $ADC_4(t)$  med og uten prosessering.