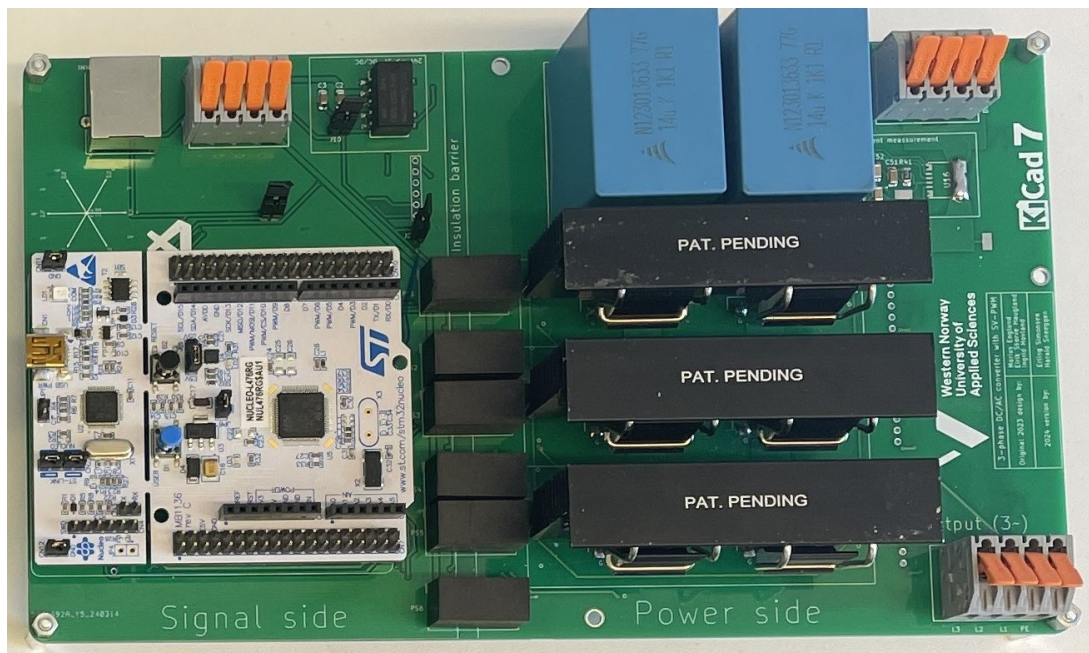




Western Norway  
University of  
Applied Sciences



# Videreutvikling av trefase DC/AC omformer med SV-PWM

Erling Simonsen  
Harald Rasmussen Sneeggen

Vår 2024

# Forord

Dette prosjektet med tittelen "Videreutvikling av trefase DC/AC omformer med SV-PWM", ble utført på Høgskulen på Vestlandet (HVL), fakultet for teknologi, miljø- og samfunnsvitenskap våren 2024.

Vi som forfattere av denne oppgave er glade for å kunne presentere arbeidet vi har gjort på veien mot en bachelorgrad i elkraftteknikk. Prosjektet ble valgt etter at vår faglærer i kraftelektronikk presenterte oppgaven, og at vi da fikk interessen for problemstillingen. For oss var dette en unik mulighet til å få se helheten i hvordan en omformer opererer.

Det siste semesteret har vi brukt på å lese oss opp på teori, utvikle en frekvensomformer og ta omformeren i bruk. Det har vært et veldig lærerikt semester, der vi har styrt vår egen hverdag. Vi er veldig glade for resultatet som er oppnådd, og for lærdommen vi tar med oss videre etter dette prosjektet.

Vi erklærer at denne oppgaven er et resultat av vår egen arbeidsinnsats, foruten å ha brukt den tidligere bachelorgruppen sin oppgave [2] som en kilde til inspirasjon, og et grunnlag for å videreutvikle omformeren.

## Anerkjennelse

Først og fremst ønsker vi å uttrykke vår takknemlighet til det tidligere bachelorgruppe-teamet, bestående av Marius Englund, Eirik Skorve Haugland og Ingrid Hovland. Deres enestående og grundige arbeid i løpet av våren 2023 har vært av uvurderlig betydning, og det har vært et fundament og grunnlag for den videre utviklingen av vårt prosjekt.

Vi vil også gi en like stor takk til vår veileder Eirik Haustveit for hans hjelpsomhet og kunnskap om faget. Vi takker for at han har veiledet oss over det som er forventet, og for at han har vært tilgjengelig og kunnet svare på store og små spørsmål under hele perioden.

For å effektivt realisere vårt prosjekt har det utvilsomt vært av stor betydning å kunne teste omformeren vår. Vi er takknemlige for den generøse bistanden fra Lars Manger Ekroll, som har stilt til rådighet måleutstyr, en likestrømsmotor og ulike komponenter som kreves for å kunne operere omformeren. Vi ønsker å uttrykke vår takknemlighet for hans tilgjengelighet, veiledning og råd som vi har mottatt underveis i prosessen. En stor takk går også til Farzan Jouleh, som har hjulpet oss med å bestille inn komponenter og kretskort.

Til slutt ønsker vi å takke Høgskulen på Vestlandet, og fakultet for teknologi, miljø- og samfunnsvitenskap for å ha gitt oss en god og variert utdanning innenfor elkraftteknikk. Det har vært noen lærerike år som har bydd på nok av utfordringer, og gjort oss klare for veien videre.

# Sammen drag

Frekvensomformere spiller en helt sentral rolle i dagens stadige mer elektriske samfunn, da de gjør det mulig å kontrollere overføring av elektrisk energi på en effektiv måte. Deres bruksområde er stort, fra styring av elektriske maskiner til kontroll av effektflyt i kraftnettet og mer. Det er derfor viktig for en Elkraftingeniør å ha god kunnskap om frekvensomformere, og det er med bakgrunn i dette at vi har valgt å fordype oss i hvordan de fungerer. Vår bacheloroppgave presenterer videre utvikling av en trefaset DC/AC omformer. Grunnet feil ved lodding av kretskortet som førte til at den ene fasen på kretskortet ble defekt, har vi i denne oppgaven testet omformeren som H-bro framfor trefaset vekselretter. Vi har benyttet kretskortet som fullbro DC/DC omformer for å drive en fremmedmagnetisert likestrøms maskin, og vi har utført tester på omformeren hvor den fungerer som enfaset vekselretter. Det er viktig for oss å presisere at vi har lagt ned mye tid til det praktiske arbeidet som går ut på å designe, bygge og teste omformeren vår.

Oppgaven vår går blant annet inn på teoretiske prinsipp for å forstå helheten i omformeren. I tillegg presenterer komponentvalg, kretskortdesign og softwareutvikling. En stor og viktig del av oppgaven er også den praktiske biten med å bygge omformeren. Vi har brukt mye tid laboratorieeksperimentering. Under testingen hadde vi resistiv og induktiv last samt drift av DC- maskin.

Vi er svært fornøyde med at vi etter en lang periode med testing, feilsøking og forbedringer har kommet der at vi har et velfungerende produkt som oppfyller de krav og forventninger som ble satt på forhånd. Det er gledelig å kunne overrekke produktet til oppdragsgiveren vår, og håper at den kommer til nytte i undervisning og laboratorieeksperimenter.

## Dokumentkontroll

<i>Rapportens tittel:</i> BO24EB-69 Videreutvikling SV-PWM Frekvensomformer	<i>Dato / Versjon:</i> 26.01.2024 / 1
	<i>Rapportnummer:</i> BO24EB-69
<i>Forfatter(e):</i> Harald Rasmussen Sneeggen Erling Simonsen	<i>Studieretning:</i> ELK
	<i>Antall sider m/vedlegg:</i> 124
<i>Høgskolens veileder:</i> Eirik Haustveit	<i>Gradering:</i> Åpen
<i>Eventuelle merknader:</i> xxxx	

<i>Oppdragsgiver:</i> Høgskolen på Vestlandet	<i>Oppdragsgivers referanse:</i>
<i>Oppdragsgivers kontaktperson(er) (inkludert kontaktinformasjon):</i> Eirik Haustveit (eirik.haustveit@hvl.no, +47 55 58 77 58) ()	

Rev	Dato	Status	Utført av
A.1.1	26.01.24	Forprosjekt mal. Levert på canvas som første utkast på oppgaven	HRS ES
A.2.1	21.05.24	Hovedoppgave siste utkast Levert.	HRS ES



## 0.1 Organisering av rapport

Vi har valgt å dele rapporten inn i tre hoveddeler. I begynnelsen av rapporten presenteres en oversikt over oppgaven. Denne delen introduserer hovedinnholdet i prosjektet samt forklaring av oppdragsgiveren, problemstillingen og til slutt et sammendrag av oppgaven.

**Del 1 inneholder teori** Her går vi først inn på elektriske maskiner, H-bro, modulasjonsteknikker og krafthalvledere. Dette kapitlet skal gi leseren en grunnleggende innføring i prosjektets teoretiske del.

**Del II inneholder utførelsen av oppgaven** Først kommer en analyse av oppgaven og videre kommer krav til kretskortet. Det er også en del om PCB designet, og utviklingen av det. Senere kommer det en del om det praktiske arbeidet med lodding og en stor del om software.

**Del III inneholder testing** Først kommer en del om laboratorie testing. Deretter kommer den del om testoppsettet vi benyttet på de ulike testene. Til slutt kommer det en analyse av test resultatene.

**Del IV inneholder vedlegg** Her kommer blant annet utgiftspost, fremdriftsplan, forkortelser, figurliste, kildeliste, kildekode for FFT-analyse og noen tabeller.

## 0.2 Bakgrunn

Våren 2023 ble det utviklet en trefase DC til AC omformer i forbindelse med et bachelorprosjekt av en gruppe Elkraft-studenter ved HVL. Denne omformerer skulle bli brukt som laboratoriemodell for studenter ved HVL, men grunnet mangel på tid for den forrige bachelorgruppen ble ikke omformerer ferdig utviklet. Vårt bachelorprosjekt våren 2024 vil basere seg på å videreutvikling av denne omformerer.

## 0.3 Oppdragsgiver

Oppdragsgiver for dette prosjektet er Høyskolen På Vestlandet (HVL). HVL er en statlig utdanningsinstitusjon for høyere utdanning, og er en av de største utdanningsinstitusjonene i Norge. HVL har totalt fem ulike studiesteder; Bergen, Sogndal, Førde, Stord og Haugesund. Antall studenter ved HVL ligger på rundt 16 500 med 1800 ansatte. Høyskolen tilbyr utdanningsprogrammer innenfor blant annet ingeniør- og maritime fag, lærerutdanninger, samfunnsvitenskap og mer. [4]

## 0.4 Problemstilling

Dette bachelorprosjektet baserer seg på å videreutvikle en trefase DC til AC omformer som ble utviklet våren 2023 av daværende HVL-studenter. Omformeren ble utviklet med et formål om å kunne styre en induksjonsmaskin ved romvektormodulering (SVPWM). Romvektormodulering er en modulasjonsteknikk som brukes til å sende styresignal til transistorene i en trefase vekselretter, og dette blir brukt til blant annet styring av elektriske motorer. Omformeren som ble utviklet er fungerende, men mangler en del funksjonaliteter som nå er ønskelig å implementere. Sammen med oppdragsgiver ble det enighet om at hovedfokusene for oppgaven skulle rettes mot følgende:

- **Reduksjon i fysisk arealbruk på kretskort.** Kretskortet som ble utviklet våren 2023 endte opp med å bli større enn nødvendig. Derfor vil et av hovedfokusene være å fysisk forminske dette kretskortet ved å designe et nytt.
- **Videreutvikle enkelt grensesnitt mot brukeren (HMI).** For overvåkning og styring av omformer er det ønskelig å utvikle et brukergrensesnitt som gjør dette enkelt og brukervennlig.
- **Utvikle og teste H-bro konfigurasjon av omformeren.** Dette innebærer å utvikle software for styring av H-bro samt utføre laboratorietester for å verifisere virkemåten ved H-bro drift. Kretskortet skal kunne fungere som fullbro DC/DC omformer med mulighet for å drifte en DC-maskin, i tillegg til at det skal kunne benyttes som enfase DC til AC omformer.

Dersom det er tid til overs kan det bli aktuelt med:

- Utvikling av 230V/400V en- og trefase likeretter med forladringskrets for filterkondensator.
- Utvikling av utgangsfiler for omformeren, enten av typen LC eller LCL.

Ved gjennomgang av komponentvalg som har blitt gjort tidligere vil vi ha som utgangspunkt å benytte samme komponenter, men med mulighet for å gjøre endringer dersom vi mener det er behov for det. Ytelsen og de generelle tekniske kravene som stilles til omformeren vil forbli uendret. Disse kravene blir nærmere beskrevet i delkapittel 5.1.

Faglige elementer i denne bacheloroppgaven er primært kraftelektronikk, kretskortdesign og microcontroller-programmering.

# Innhold

0.1	Organisering av rapport . . . . .	4
0.2	Bakgrunn . . . . .	4
0.3	Oppdragsgiver . . . . .	4
0.4	Problemstilling . . . . .	5
<b>I</b>	<b>Teori</b>	<b>10</b>
<b>1</b>	<b>Elektriske maskiner</b>	<b>11</b>
1.1	DC-Maskiner . . . . .	11
1.2	Induksjonsmotoren (asynkronmaskin) . . . . .	12
1.2.1	Ulike driftsmodus for asynkronmaskinen . . . . .	12
1.2.2	Konstant $\frac{U}{f}$ . . . . .	13
1.3	Synkronmaskinen . . . . .	14
<b>2</b>	<b>Krafthalvledere</b>	<b>15</b>
2.1	Effekttap i transistorer . . . . .	16
2.2	Ulike krafttransistorer . . . . .	16
2.3	Gate drivere . . . . .	18
<b>3</b>	<b>H-bro</b>	<b>19</b>
3.1	H-bro topologi . . . . .	19
3.2	Unipolar styring av H-bro . . . . .	21
3.3	Bipolar styring av H-bro . . . . .	24
<b>4</b>	<b>Modulasjonsteknikker for DC til AC omformere</b>	<b>26</b>
4.1	Puls-bredde modulasjon (PWM) . . . . .	27
4.2	Sinus PWM (SPWM) . . . . .	28
4.3	Space Vector PWM . . . . .	31
<b>II</b>	<b>Utførelse</b>	<b>36</b>
<b>5</b>	<b>Analyse av oppgaven</b>	<b>37</b>
5.1	Analyse av tidligere design . . . . .	37
5.1.1	Elektriske parameter for omformere . . . . .	37
5.1.2	Komponentvalg . . . . .	38

5.1.3	Kraftforsyninger . . . . .	40
5.1.4	Microcontroller . . . . .	40
5.1.5	Målesystemer . . . . .	41
5.2	Analyse av kretskortdesign . . . . .	42
5.2.1	Krav til kretskort . . . . .	42
5.2.2	Potensiale for arealeffektivisering av PCB . . . . .	43
5.3	Utvikling av eget HMI-kort . . . . .	47
5.3.1	Krav til HMI-kort . . . . .	47
5.3.2	Komponenter til HMI-kort . . . . .	47
5.4	Analyse av software . . . . .	48
<b>6</b>	<b>PCB design i KiCad</b>	<b>49</b>
6.1	Utvikling av PCB for HMI-kort . . . . .	49
6.2	Utvikling av PCB for hovedkort . . . . .	51
6.3	Lodding av omformer . . . . .	55
6.4	Sammenligning av ny og gammel versjon . . . . .	56
<b>7</b>	<b>Software</b>	<b>57</b>
7.1	Viktige konsepter . . . . .	57
7.1.1	Timere . . . . .	57
7.1.2	Output-compare (OC) . . . . .	58
7.1.3	Dødtid . . . . .	58
7.1.4	Ulike PWM-moduser ved output-compare . . . . .	58
7.2	Software for trefasedrift . . . . .	59
7.2.1	Konstant $\frac{U}{f}$ -kontroll for trefasedrift . . . . .	59
7.2.2	Regulering av modulasjonsfrekvens . . . . .	59
7.2.3	Modulasjonsalgoritmer for trefasedrift . . . . .	60
7.3	Software for drift av H-bro . . . . .	62
7.3.1	H-bro som DC/DC omformer . . . . .	62
7.3.2	H-bro som enfase vekselretter . . . . .	63
<b>III</b>	<b>Testing</b>	<b>64</b>
<b>8</b>	<b>Laboratorie testing</b>	<b>65</b>
8.1	Testresultater . . . . .	65
8.1.1	Verifikasjon av pulsbredde modulasjongenerering . . . . .	66
8.1.2	Test av gate-motstand . . . . .	71
8.1.3	Test av H-bro som DC/DC omformer . . . . .	75
8.1.4	Test av H-bro som enfase DC/AC omformer . . . . .	79
8.1.5	Test av DC-motor drift . . . . .	83
8.2	Testoppsett . . . . .	84
8.2.1	Testoppsett for H-bro som enfaset vekselretter . . . . .	84
8.2.2	Drift av fremmedmagnetisert DC-Maskin . . . . .	85
<b>9</b>	<b>Analyse av testresultater</b>	<b>86</b>
9.1	Verifikasjon av PWM . . . . .	86

9.1.1	Verifikasjon av dødtid . . . . .	86
9.1.2	Verifikasjon av styresignaler . . . . .	86
9.1.3	Måling av beregningshastighet . . . . .	87
9.2	Test av ulik gate-motstand . . . . .	88
9.3	Test av H-bro som DC/DC omformer . . . . .	89
9.4	Test av H-bro som DC/AC omformer . . . . .	91
9.4.1	Grunnleggende tester . . . . .	91
9.4.2	Test av overmodulasjon . . . . .	92
9.4.3	Test av DC-motor . . . . .	94
<b>Konklusjon</b>		<b>95</b>
<b>Forslag til videre arbeid</b>		<b>96</b>
9.4.4	Videreutvikling av HMI . . . . .	96
9.4.5	Testing og programmering av målesystemer . . . . .	97
9.4.6	Videre testing og utvikling av bremsechopper . . . . .	97
9.4.7	Utvikling av likeretter . . . . .	97
9.4.8	Utvikling av filter . . . . .	97
<b>IV Vedlegg</b>		<b>98</b>
<b>A Utgiftspost for prosjektet</b>		<b>99</b>
<b>B Fremdriftsplan</b>		<b>100</b>
B.1	Estimert fremdriftsplan . . . . .	100
B.2	Faktisk fremdrift . . . . .	100
B.3	Logg for laboratoriearbeid . . . . .	101
<b>C Forkortelser og porgrammer</b>		<b>103</b>
C.1	Forkortelser . . . . .	103
C.2	Liste med programvare brukt i oppgaven . . . . .	104
<b>D Liste med feil for PCB og software</b>		<b>105</b>
D.1	Liste med feil på PCB for hovedkort . . . . .	105
D.2	Liste med feil på PCB for HMI-kort . . . . .	105
D.3	Liste med feil i software . . . . .	105
<b>E Kode</b>		<b>107</b>
E.1	MatLab-kode for å utføre FFT-analyse på måledata . . . . .	107
E.2	MatLab-kode for å posserere samt plotte måledata fra picoscope . . . . .	109
E.3	Kode for enfase SPWM . . . . .	111
E.4	Kode for bipolar styring av fullbro DC/DC omformer . . . . .	113
E.5	Kode for unipolar styring av fullbro DC/DC omformer . . . . .	113
E.6	Kode for trefasedrift av omformer . . . . .	114
<b>F Kretskortutlegg</b>		<b>116</b>
F.1	HMI-kort . . . . .	116

F.2 Hovedkort . . . . .	117
<b>G Måledata fra diverse tester</b>	<b>118</b>
G.1 Måledata fra test av spenningrippel ved unipolar og bipolar fullbro DC/DC	119
G.2 Avgitt grunnharmonisk spenning ved overmodulasjon på enfase SPWM . .	120
<b>Kildeliste</b>	<b>123</b>

Del I

Teori

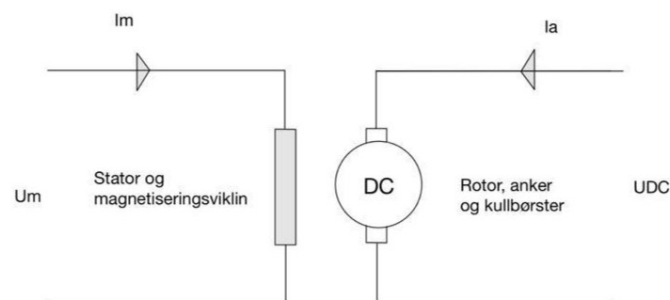
# Kapittel 1

## Elektriske maskiner

### 1.1 DC-Maskiner

Likestrøms maskinen har en viklet stator, som er maskinens magnetiseringssystem. Dreiemomentet til maskinen er produktet av magnetiseringsstrøm og ankerstrøm. Magnetiseringskretsen i likestrømsmaskinen kan kobles opp på to måter, enten parallell eller serie. Fordelen med en parallellkoblet motor er at den får en jevn momentkurve, mens den seriekoblede motoren får et høyere startmoment. Hos sistnevnte svekkes dreiemomentet ved økende hastighet. Parallell- og seirekobling kan kombineres til det som kalles en kompondmaskin, som innehar to magnetiseringsviklinger. Den ene er da koblet i parallell, mens den andre i serie med ankerviklingen. En annen variant er den fremmedmagnetiserte maskinen. Den har adskilt anker og magnetiseringskrets, som er den varianten vi brukte under testing 8.1.5. [9, s. 183]

#### Prinsippskisse for fremmedmagnetisert DC-maskin



Figur 1.1: Ekvivalentskjema fremmedmagnetisert DC-maskin

*Kilde: Tegnet i Goodnotes*



## 1.2 Induksjonsmotoren (asynkronmaskin)



Figur 1.2: Asynkronmaskin[7]

Asynkronmaskinen bruker induksjonsprinsippet og er en roterende elektromagnetisk maskin. Asynkronmaskinen benyttes bare for vekselstrøm da den er vekselstrømmagnetisert. Grunnen til at motoren kalles en asynkronmaskin er at den går med et asynkront turtall, ved en gitt belastning. Det asynkrone turtallet varierer fra vekselstrømsnettets dreiefeltfrekvens avhengig av den mekaniske lasten som er påført rotoren. Asynkronmaskinen brukes hovedsaklig til motordrift, men kan også i enkelte tilfeller brukes som generator, der energiflyten går tilbake på nettet. Asynkronmaskinen ansees som en enkel og robust konstruksjon, og har gode driftsegenskaper. Motoren er også relativt rimelig å anskaffe. [8, s. 87]

Det synkrone turtallet til dreiefeltet ( $n_1$ ) som produseres i statorviklingene til en asynkronmaskin er en funksjon av frekvensen til påtrykt spenning ( $f$ ) samt antall polpar i maskinen ( $P$ ) [8]. Ligning 1.1 beskriver denne sammenhengen.

$$n_1 = \frac{60 \cdot f}{P} \quad (1.1)$$

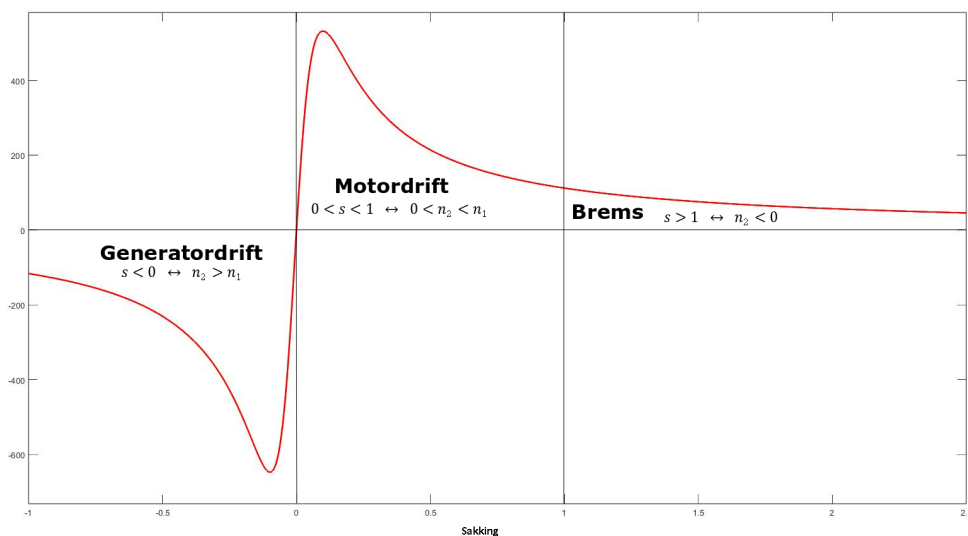
Sakking er definert som differansen mellom rotorens turtall og dreiefeltets turtall. Denne differansen kalles sakketurtalet eller slipturtalet. Matematisk kan sakingen beskrives som i ligning 1.2.

$$s = \frac{n_1 - n_2}{n_1} = \frac{n_{SL}}{n_1} \quad (1.2)$$

### 1.2.1 Ulike driftsmodus for asynkronmaskinen

En asynkronmaskin kan operere i tre ulike former: motordrift, generatordrift og i elektrisk bremsemodus. I motordrift har vi positiv sakking. Sakkingen i motordrift vil ligge mellom 1 og 0, avhengig av den mekaniske lasten på rotorakslingen.

Ettersom at asynkronmaskinen arbeider etter induksjonsprinsippet vil den induerte spenningen i rotorkretsen være beskrevet av Faradays lov. Faradays lov sier at induert



Figur 1.3: Avgitt dreiemoment fra asynkronmaskin som funksjon av sakking  
*Kilde: laget i MatLab og Inkscape*

spenning  $E$  er proporsjonal med endringen i magnetisk flux  $\Phi$ , som beskrevet i ligning 1.3.

$$E = -\frac{d\Phi}{dt} \quad (1.3)$$

Dersom rotoren roterer med samme rotasjonshastighet som dreiefeltet i stator, vil ikke rotorkretsen oppleve en endring i magnetisk flux. Dette vil si at  $\frac{d\Phi}{dt} = 0$ , og dermed følger det fra Faradays lov at den induerte spenningen i rotorstavene er 0V. Som følge av dette fenomenet vil asynkronmaskinen gå i tomgang når sakkingen er 0.

Dersom sakkingen er negativ har vi kommet over i generatordrift. Det som da skjer er at vi har et aktivt mekanisk påtrykk fra en turbin som gjør at turtallet presses forbi det synkrone turtallet. I dette tilfellet blir effektflyten ført tilbake på nettet. Denne løsningen blir oftest brukt i de minste vann eller vindkraftverk med en installert effekt på noen hundre kilowatt. Når sakkingen blir større enn 1 har vi kommet over i elektromagnetisk bremsemodus. Denne driftstilstand oppstår dersom to av fasene byttes om under drift. Her vil dreiefeltet i stator byttes om, samtidig som rotoren fortsatt fortsetter i samme retning på grunn av den mekanisk roterende lasten i omkoplings-øyeblikket. I dette øyeblikket blir sakkingen opp i mot 2, samtidig som motoren bremses opp og sakkingen går mot 1. Når motoren når stillstand må den koples av nettet for å unngå at den akselerer opp i motsatt dreieretning.[8, s. 119–120]

### 1.2.2 Konstant $\frac{U}{f}$

Konstant  $U/f$ , også kjent som skalar kontroll er en mye brukt metode for å kontrollere turtall og momentet for en asynkronmaskin. Metoden omhandler proporsjonal justering av amplituden og frekvensen av spenningen som er påtrykket stator, som gir maskinen mulighet til å levere maksimalt moment samtidig som strømmene og frekvensen holdes

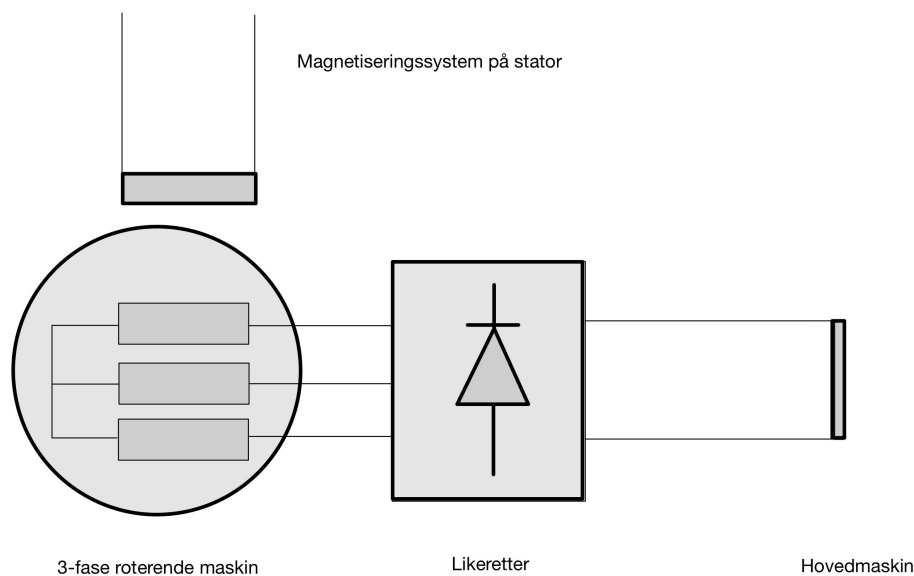
innenfor de gitte verdiene. Den mekaniske lasten på akslingen kan gjøre at nøyaktigheten til maskinen kan unngå noe på grunn av sankingen i maskinen. Når frekvensen i maskinen går under cirka 20% av merkefrekvens,  $fn$ , blir det nødvendig å gradvis øke volt-per-hertz forholdet for å kompensere for spenningsfallet som oppstår på grunn av motstand i statorviklingene.[2, s. 9].

### 1.3 Synkronmaskinen

Synkronmaskinen brukes i all hovedsak til generatordrift i kraftverk. I motsetning til asynkronmaskinen har synkronmaskinen et turtall som er fast med frekvensen. Synkronmaskinen har mulighet til å styre effektfaktoren ved å endre på magnetiseringsstrømmen i maskinen. Det som da skjer er at man endrer den magnetiske hovedfluxen. Den er også veldig robust mot spenningsvariasjoner. Synkronmaskinen kan også brukes som motor, men da i relativt store fabrikkutstyr eller pumpekraftverk.[8, s. 44]

**Magnetiseringssystemet til synkronmaskinen** Det har tradisjonelt sett vært vanlig å magnetisere synkronmaskiner via børster og sleperinger. Det kan gjøres enten ved å ha en DC-generator montert på akslingen, eller ved å ha et statisk magnetiseringssystem. I det sistnevnte systemet oversøres DC-strømmen direkte via børster og sleperinger til magnetiseringssystemet i rotoren. Denne metoden benyttes oftest i store vannkraftverk.

Det finnes også børsteløse synkronmaskiner. Disse har ikke sleperinger. Denne varianten er oftest brukt i mindre vannkraftverk, ombord i båter og i industrielle anlegg. Her er rotoren utstyrt med en AC-DC omformer (likeretter) som mottar vekselstrøm fra rotorviklingen i en omvendt synkronmaskin (ytterpolmaskin) som er montert på samme aksling. Denne omvendte synkronmaskinens poler danner da stator der den tilføres likestrøm fra et statisk magnetiseringssystem.[8, s. 47]



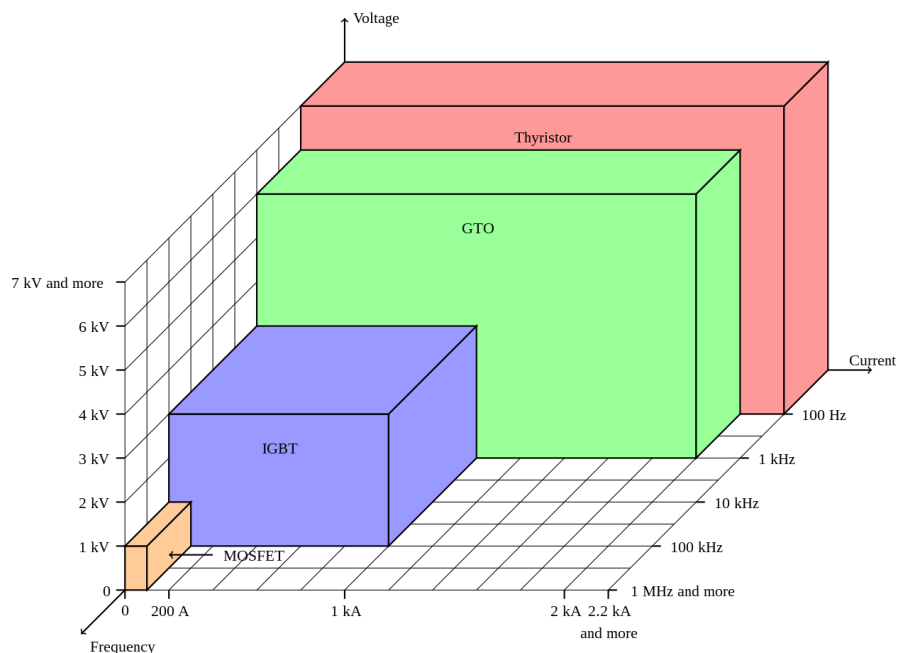
Figur 1.4: Børsteløst system  
Kilde: Tegnet i Goodnotes

# Kapittel 2

## Krafthalvledere

En vanlig klassifisering av krafthalvledere er å skille mellom ukontrollerte, halvkontrollerte og fullkontrollerte halvledere [10, s. 59]. Ukontrollerte halvledere er halvledere som mangler en kontrollelektrode. Mangel på kontrollelektrode gjør at halvlederen ikke kan styres av og på av et eksternt styresignal. Et eksempel på en ukontrollert halvleder er dioder. Halvkontrollerte halvledere er halvledere som kan kontrolleres til å bli skrudd på, men ikke av, når en ønsker. Et eksempel på en halvkontrollert halvleder er tyristorer, som krever nullgjennomgang for å slukkes. Sist men ikke minst har vi fullkontrollerte halvledere, som kan kontrolleres til å skrus både av og på av et eksternt styringssignal. Transistorer er en type fullkontrollert halvleder. I dette kapittelet vil vi ta for oss grunnleggende teori om fullkontrollerte krafthalvledere, hvor spenningsstyrte transistorer er i fokus.

Figur 2.1 gir en oversiktlig klassifisering av de mest brukte krafthalvlederne.



Figur 2.1: Transistorer. [6]

## 2.1 Effekttap i transistorer

Med effekttap i transistorer menes det uønsket effekt som omsettes i transistoren. Denne effekten blir omgjort til varme, som fører til varmgang i transistoren. Som følge av disse effekttapene er det kritisk at en kjøleribbe dimensjoneres slik at varmen fordeles ut på et større overflateareal. Effekttap i transistorer vil bidra til å redusere den totale virkningsgraden for en omformer. Vi skiller hovedsaklig mellom to ulike former for tap i en transistorer; svitsjetap og ledetap.

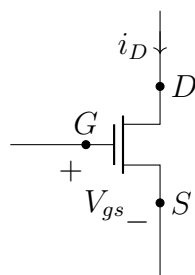
**Ledetap** oppstår i transistoren mens den leder strøm. Ledetap oppstår fordi det alltid vil være et lite spenningsfall over transistoren mens den leder. Og siden  $P = U \cdot I$  vil det dermed omsettes effekt i transistoren. Spenningsfallet ligger vanlig vis mellom én til et to volt [5]. Ledetapet vil være proporsjonalt med strømmen som ledes gjennom transistoren samt spenningsfallet over transistoren.

**Svitsjetap** oppstår i transistoren under overgangsintervallet fra ledende til ikke-ledende tilstand og motsatt. I dette overgangsintervallet vil både strømmen gjennom transistoren samt spenningen over den være ulik null, som resulterer i at det omsettes effekt i transistoren.

## 2.2 Ulike krafttransistorer

Vi vi her ta for oss de vanligste transistorene innenfor kraftelektronikk: MOSFET- og IGBT transistorer.

**MOSFET**, forkortelse for "Metal Oxide Field Effect Transistor", er en spenningsstyrt transistor. Den består av tre terminaler: drain, source og gate. Som vist i figur 2.1 er MOSFET den typen krafttransistor som kan operere med høyest svitsjefrekvens. Som følge av at MOSFET-transistorer kan skru seg av og på med høy hastighet vil svitsjetapene i en slik transistor være lave, sammenlignet med andre typer [10, s. 77].

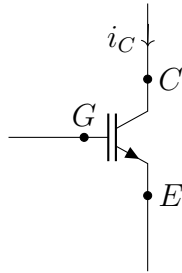


Figur 2.2: MOSFET symbol  
*Kilde: laget med Circuitikz*

En N-type MOSFET begynner å lede strøm fra drain til source når spenningen mellom gate og source  $V_{gs}$  er større eller lik en gitt terskelspenning  $V_{th}$ , typisk mellom 2 til 4 volt [3, s. 23]. For gate-source spenninger lavere enn dette er transistoren fullstendig av. Når en så øker  $V_{gs}$  over terskelspenningen vil drain-strømmen, som i praksis er strømmen gjennom transistoren, være proporsjonal med  $V_{gs}$  [3, s. 23]. Så ved å styre spenningen  $V_{gs}$

ved hjelp av for eksempel en mikrokontroller kan vi kontrollere når transistoren skal lede og når den ikke skal det.

**IGBT** er en forkortelse for "Insulated Gate Bipolar Transistor". IGBT-transistorer er også spenningsstyrte, som vil si at de i praksis styres på samme måte som en MOSFET. IGBT-transistorer kan håndtere høye spenninger, noe som gjør de svært aktuelle for høyeffekt-applikasjoner. En viktig egenskap hos IGBT-transistorer er at de har lave ledetap sammenlignet med andre typer [10, s. 78]. Men i motsetning til MOSFET kan IGBT-transistorer 'kun' arbeide med svitsjefrekvenser opp til 30kHz [3, s. 23]. Det er van-

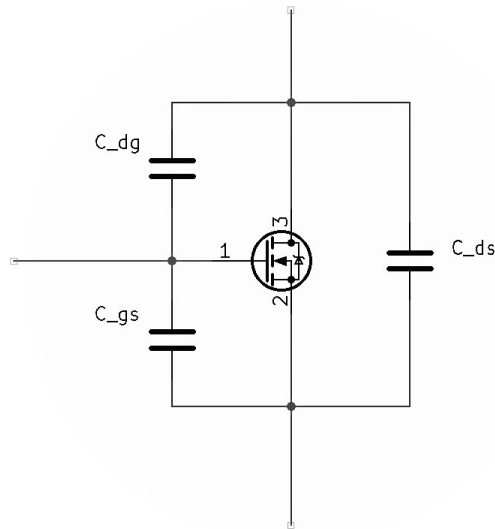


Figur 2.3: IGBT symbol  
*Kilde: laget med Circuitikz*

lig å skille mellom *symmetriske* og *asymmetriske* IGBT-transistorer [10, s. 79]. Det som skiller disse to er at en asymmetrisk IGBT ikke kan blokkere negative spenninger mellom kollektor og emitter, mens symmetriske IGBT kan. Den mest vanlige typen er den asymmetriske med innebygget friløps-diode koblet i anti-paralell med transistoren. Symmetriske IGBT egner seg derimot godt til omformere med AC-input, men denne typen IGBT har høyere ledetap.

## 2.3 Gate drivere

En typisk forenkling som gjøres ved modellering av transistorer i forbindelse med kraftelektronikk er å anse transistoren som en ideell bryter. Med ideell menes det at transistoren kan skru seg av og på momentant. Dette er ikke tilfellet i den virkelige verden. Det vil alltid ta en hvis tid før transistoren skrur seg av og på, som fører til svitsjetap. Jo lenger tid transistoren bruker på å skru seg på, jo høyere blir svitsjetapene. I tillegg begrenser dette tidsintervallet hvor fort transistoren kan skru seg av og på. For å redusere tidsintervallet benyttes en gate driver.



Figur 2.4: Ekvivalentskjema for MOSFET

*Kilde: laget i KiCad*

Enhver MOSFET vil ha en gitt kapasitans mellom gate og source. Denne kapasitansen kan modelleres som en kondensator  $C_{gs}$ , og fra grunnleggende elektroteknikk vet vi at spenningen over en kondensator ikke kan endre seg momentant, som følge av at kondensatorer må lades opp før spenningen over den når stasjonær verdi. Dette skaper trøbbel når vi skal skru av eller på en MOSFET med høy hastighet, da MOSFET styres av påtrykt spenning mellom gate og source. Fordi spenningen mellom gate og source i praksis ikke kan endre seg momentant som følge av  $C_{gs}$ , medfører dette at MOSFET ikke kan skrues av eller på momentant. Desto hurtigere vi klarer å lade opp og ut  $C_{gs}$ , desto hurtigere vil transistoren kunne svitsje.

Elektrisk strøm er definert som endring i ladning per tidsenhet:

$$I = \frac{dq}{dt} \quad (2.1)$$

En annen måte å tolke ligning 2.1 på er at endring i ladning er proporsjonal med strøm. Dette betyr at for å kunne lade opp  $C_{gs}$ , hurtig (høy  $\frac{dq}{dt}$ ) må vi øke strømmen inn på gate. For å få til dette benyttes en gate driver krets. Den sørger for å levere en høy strøm inn på gate i et kort tidsintervall slik at  $C_{gs}$  lades opp hurtig og deretter lade ut  $C_{gs}$  for ladning ved å koble  $C_{gs}$  til jord eller et negativt potensial når transistoren skrues av.

# Kapittel 3

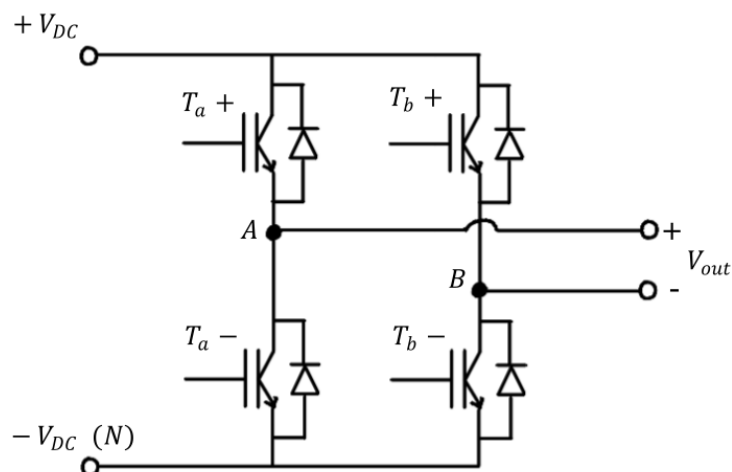
## H-bro

### 3.1 H-bro topologi

En H-bro er en krets som består av to bein med brytere, som oftest transistorer, med to brytere per bein. Den øverste bryteren på beinet refereres ofte til som high-side (HS) mens den nedre bryteren på beinet refereres til som low-side (LS). Det er vanlig å referere til de to beinene som bein A og bein B, og high-side transistoren på beinet som pluss og low-side som minus. En vanlig notasjon for bryterparene er:

$$\begin{aligned} &T_{a+} \text{ og } T_{a-} \\ &T_{b+} \text{ og } T_{b-} \end{aligned}$$

De to transistorene på samme bein kan aldri lede samtidig da dette medfører kortslutning av DC-forsyningen. Styresignalene til de to transistorene på samme bein er derfor alltid invertert i forhold til hverandre, så dersom øvre transistor leder tilser det implisitt at nedre transistor sperrer, og motsatt.



Figur 3.1: Standard H-bro topologi  
*Kilde: laget med ReMarkable*



En H-bro kan operere i alle fire kvadranter av  $V_d$ - $I_d$  planet [5]. Dette vil i praksis si at effektflyten kan gå begge veier gjennom omformerer, noe som gir H-broen en stor fordel ovenfor andre typer DC/DC omformere. En annen ting å merke seg er at en H-bro krets også kan brukes som énfase DC til AC omformer, alt ettersom hvilke PWM-algoritme som styrer bryterene.

Når polariteten på utgangsspenningen defineres som i figur 3.1 får en følgende uttrykk for utgangsspenningen:

$$V_{out} = V_{AN} - V_{BN} \quad (3.1)$$

hvor  $V_{AN}$  er spenningen mellom punkt A og N, og  $V_{BN}$  er spenningen mellom punkt B og N [5]. I praksis er N jord på forsyningen til H-broen. Dersom  $T_a+$  leder så blir  $V_{AN} = V_{DC}$ , og dersom  $T_b+$  leder så er  $V_{BN} = V_{DC}$ . Og dersom nedre transistoren i et av beinene leder, for eksempel  $T_a-$ , så kobles da punkt A til jord som medfører at  $V_{AN} = 0V$ . Tabell 3.1 oppsummerer hva utgangsspenningen blir i de ulike brytertilfellene.

Tabell 3.1: Utgangsspenning fra H-bro ved ulike brytertilstander

$T_a+$	$T_b+$	$V_{AN} - V_{BN}$	$V_{out}$
På	Av	$V_{DC} - 0$	$V_{DC}$
Av	På	$0 - V_{DC}$	$-V_{DC}$
På	På	$V_{DC} - V_{DC}$	0
Av	Av	$0 - 0$	0

Snittspenningen av  $V_{AN}$  avhenger av arbeidssyklusen til PWM-signalet som styrer  $T_a+$ :

$$V_{AN_{avg.}} = V_d \cdot \frac{t_{ON_A}}{T_{sw}} = V_d \cdot D_A \quad (3.2)$$

Og tilsvarende for  $V_{BN}$ :

$$V_{BN_{avg.}} = V_d \cdot \frac{t_{ON_B}}{T_{sw}} = V_d \cdot D_B \quad (3.3)$$

Kombinerer vi uttrykk 3.2 og 3.3 med 3.1 får vi følgende uttrykk for utgangsspenningen fra H-broen:

$$V_{out_{avg.}} = V_d \cdot D_A - V_d \cdot D_B = V_d \cdot (D_A - D_B) \quad (3.4)$$

Ligning 3.4 gjør det tydelig at ved å kontrollere arbeidssyklusen til styringssignalene til transistorene i H-broen,  $D_A$  og  $D_B$ , kan man styre avgitt snittspenning fra omformerer [5]. Det finnes flere måter å gjøre dette på, og videre i dette kapittelet skal vi ta for oss to av de vanligste metodene; unipolar og bipolar styring.

## 3.2 Unipolar styring av H-bro

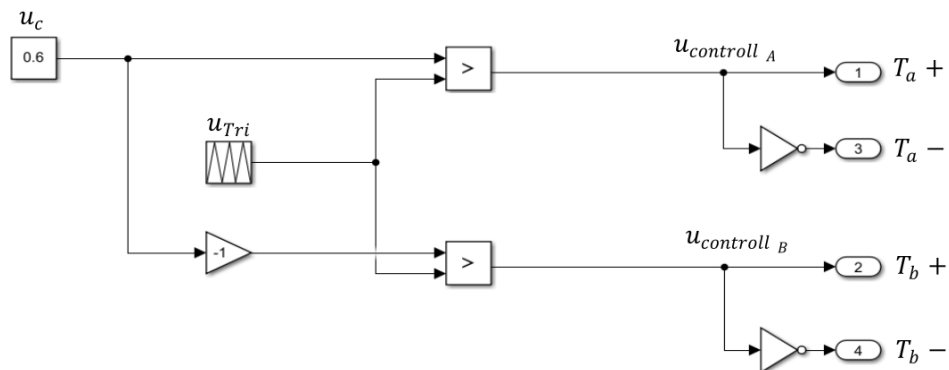
Unipolar styring er en metode som benyttes for å kunne kontrollere avgitt snittspenning, og dens polaritet, fra en H-bro. Dette gjøres via PWM-styringssignaler som da bestemmer  $D_A$  og  $D_B$ . Hvorfor metoden kalles for unipolar vil vi komme tilbake til.

Metoden fungerer ved at to kontrollsignaler,  $u_{control_A}$  og  $u_{control_B}$ , sammenlignes med en triangelkurve  $u_{Tri}$  med frekvens  $f_{sw}$ .  $u_{control_A}$  settes til å styre bryterparet i bein A, mens  $u_{control_B}$  styrer bein B. La  $n$  være et av de to beinene i H-broen. Når  $u_{control_n}$  er større enn momentanverdien til triangelkurven så leder øvre transistor i bein  $n$  (mens nedre transistor sperrer), mens når momentanverdien til triangelkurven blir større enn  $u_{control_n}$  så sperrer øvre transistor (mens nedre transistor leder) [5, s. 192]. Ofte vil triangelkurven ligge i intervallet  $[-1, 1]$ . Vi definerer et basis-kontrollsignal  $u_c$ . Ved unipolar styring settes de to kontrollsignalene til følgende:

$$u_{control_A} = +u_c$$

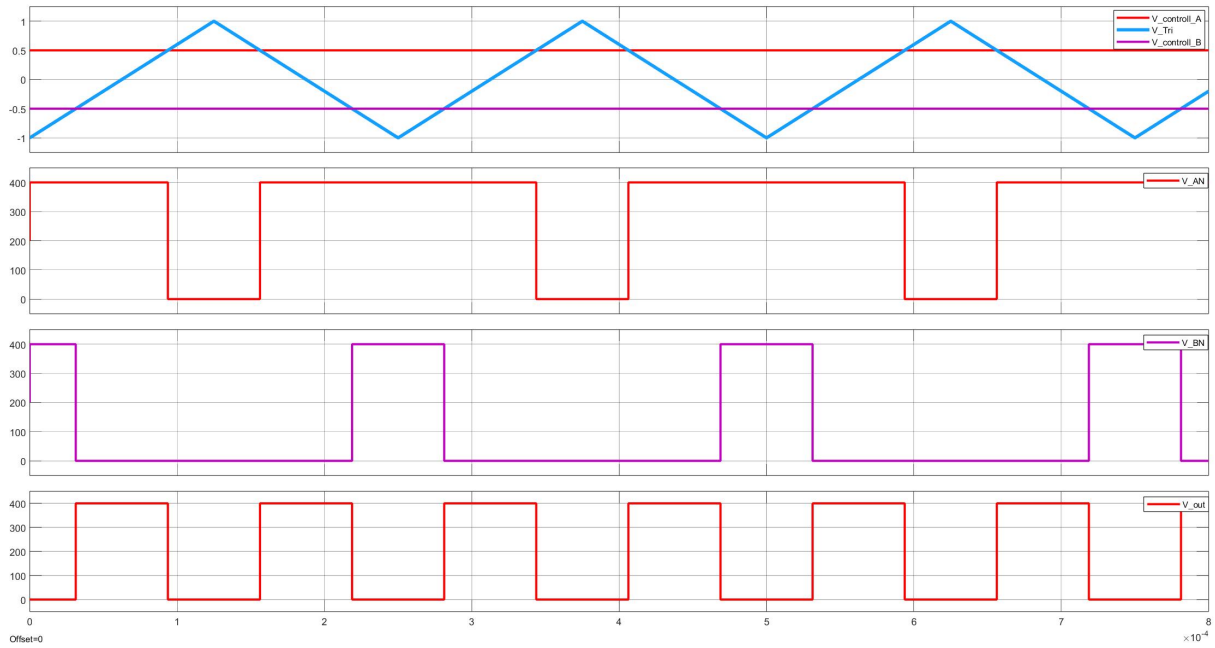
$$u_{control_B} = -u_c$$

Unipolar styring kan oppnås ved å benytte logikken i figur 3.2.

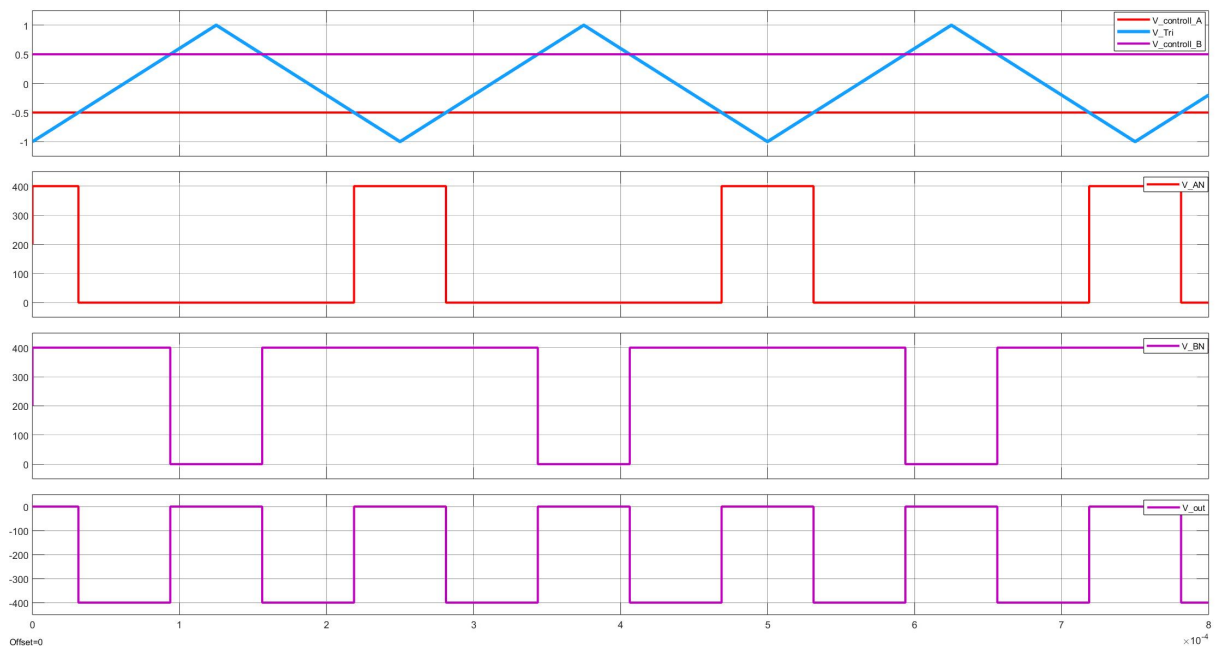


Figur 3.2: Kontrollsystem for unipolar styring med  $u_c = 0.6$

*Kilde: laget i SimuLink*



Figur 3.3: Simulering av unipolar styring med  $u_c = 0.5$  og positiv avgitt spenning  
*Kilde: simulering i SimuLink*



Figur 3.4: Simulering av unipolar styring med  $u_c = 0.5$  og **negativ** avgitt spenning  
*Kilde: simulering i SimuLink*

Figur 3.3 viser en simulering av avgitt spenning fra en H-bro ved unipolar styring. De to midterste grafene viser  $V_{AN}$  og  $V_{BN}$ , mens den nederste grafen viser den avgitte spenningen  $V_{out}$ . Vi kan se at jo større  $u_c$  blir, jo større blir  $D_A$  og jo mindre blir  $D_B$ , som gjør at snittet av den avgitte spenningen øker. En viktig observasjon fra figur 3.3 er at den avgitte spenningen alltid veksler mellom 0V og  $+V_{DC}$ ; ergo den har kun èn polaritet. Det er på grunn av dette at denne styringsformen kalles for *unipolar* kontroll.

Avgitt spenning ved unipolar styring kan uttrykkes:

$$V_{out_{avg.}} = V_d \cdot (2 \cdot D_A - 1) \quad (3.5)$$

Vi har følgende sammenheng mellom arbeidssyklus til PWM-signal for bein A og bein B:

$$D_B = 1 - D_A \quad (3.6)$$

Ved å kombinere ligning 3.1, 3.5 og 3.6 kan en utlede uttrykk 3.7 [5].

$$V_{out_{avg.}} = V_{DC} \cdot \frac{u_c}{u_{Tri}} \quad (3.7)$$

For å oppnå positiv polaritet på avgitt snittspenning (dersom polariteten er definert som i figur 3.1) så settes kontrollsignalene slik at  $u_{control_A}$  ligger øverst av de to kontrollsignalene ( $u_{control_A} > u_{control_B}$ ). Men dersom en ønsker negativ polaritet på avgitt snittspenning, for eksempel fordi man ønsker å endre dreieretning på en DC-motor, kan man oppnå dette ved å endre rekkefølgen på kontrollsignalene slik at  $u_{control_B}$  ligger øverst på triangelkurven ( $u_{control_A} < u_{control_B}$ ). Den avgitte spenningen vil da veksle mellom 0V og  $-V_{DC}$ . Et slikt driftstilfelle er illustrert i figur 3.4.

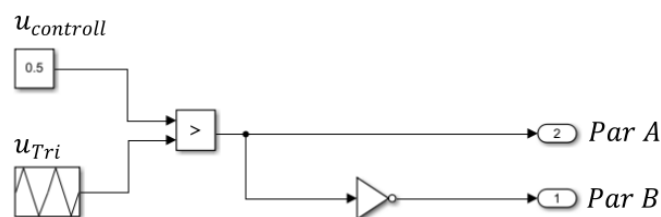
### 3.3 Bipolar styring av H-bro

Ved bipolar styring benyttes det, i motsetning til unipolar, kun ett kontrollsignal  $u_{control}$  som sammenlignes med en triangelkurve  $u_{Tri}$ . De fire transistorene i H-broen styres i par, hvor parene er som følge:

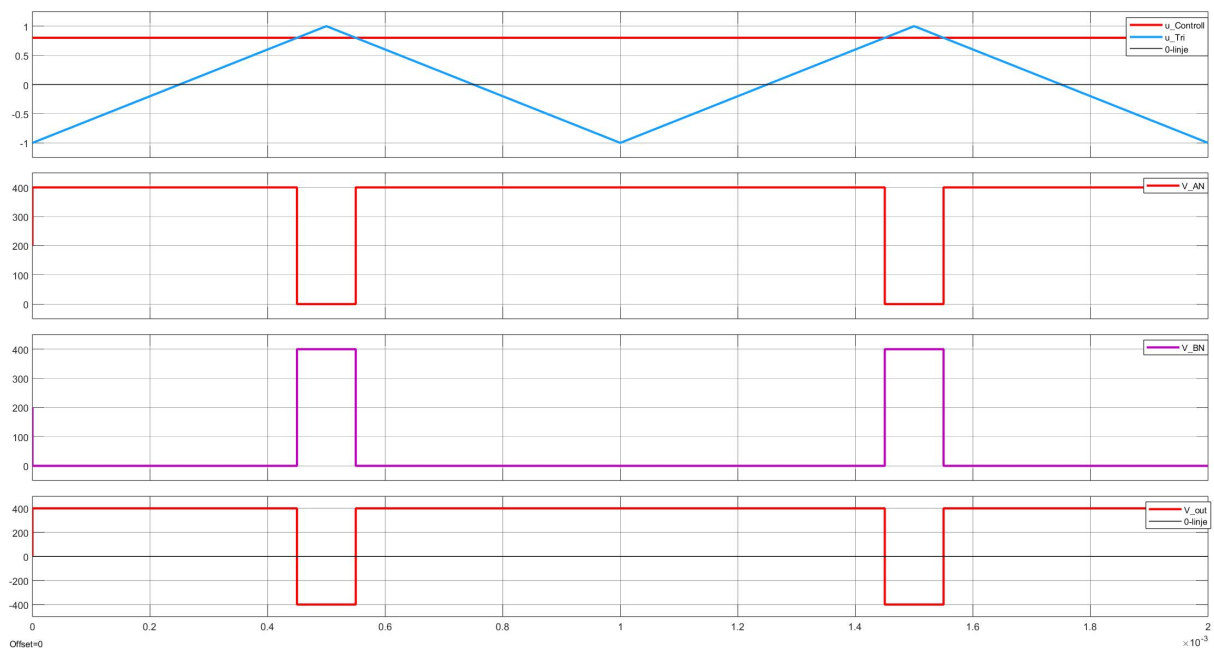
**Par A:**  $T_{a+}$  og  $T_{b-}$

**Par B:**  $T_{a-}$  og  $T_{b+}$

Når  $u_{control}$  er større enn  $u_{Tri}$  vil par A lede, mens når  $u_{control}$  er mindre enn triangelkurven vil par B lede [5, s. 190]. Den avgitte spenningen vil da bli positiv. Styringslogikken kan oppsummeres som vist i figur 3.5.

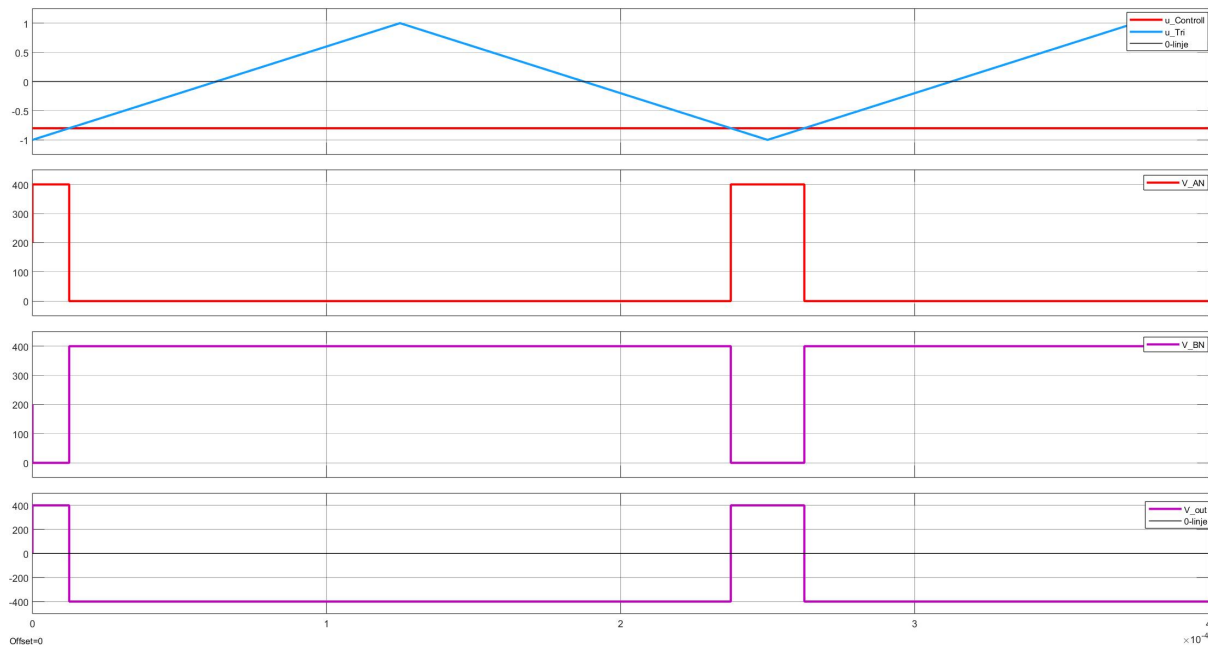


Figur 3.5: Styringslogikk for bipolar styring  
Kilde: laget i SimuLink



Figur 3.6: Simulering av bipolar styring med  $u_c = 0.8$  og positiv avgitt spenning  
Kilde: laget i SimuLink

En viktig observasjon fra figur 3.6 er at den avgitte spenningen veksler mellom  $+V_{DC}$  og  $-V_{DC}$ . Det er med bakgrunn i dette at algoritmen kalles *bipolar* kontroll. Dette er til stor forskjell fra unipolar kontroll som veksler mellom  $+V_{DC}$  og  $0V$ . På grunn av dette vil den



Figur 3.7: Simulering av bipolar styring med  $u_c = -0.8$  og **Negativ** avgitt snittspenning  
*Kilde: laget i SimuLink*

påtrykte DC-spenningen til omformerer bli bedre utnyttet ved bipolar styring, ettersom at arealet under kurven for én periode blir større enn arealet under kurven ved unipolar.

Den avgitte snittspenningen ved bipolar styring kan uttrykkes:

$$V_{out_{avg}} = V_{DC} \cdot \frac{u_{control}}{u_{Tri}} \quad (3.8)$$

Effektivverdien (RMS) av avgitt spenning, derimot, vil være lik den påtrykte DC-link spenningen.

Gitt at polariteten på utgangsspenningen er definert som i 3.1 kan en oppnå negativ polaritet ved å styre  $u_{control}$  til å være et negativt tall (gitt at  $V_{Tri} \in [-1, 1]$ ). Da vil  $V_{BN}$  være på lengre enn  $V_{AN}$  ( $D_B > D_A$ ), som fører til at

$$V_{out_{avg.}} = V_d \cdot D_A - V_d \cdot D_B < 0$$

Et slikt driftstilfelle er vist i figur 3.7.

## Kapittel 4

# Modulasjonsteknikker for DC til AC omformere

Med styring av DC til AC omformere menes det i praksis hvordan styringen av transistorene i vekselretteren foregår. I kapittel 2 blir det forklart at en spenningsstyrt transistor vil skru seg av og på i takt med spenningen som påtrykkes mellom gate og source. Som følge av dette kan en styre transistoren ved å påtrykke spenningspulser mellom gate og emitter. En modulasjonsteknikk er en bestemt måte å påtrykke styresignal på transistorene i en omformer for å oppnå ønsket styring. I dette kapittelet vil vi ta for oss to av de vanligste modulasjonsteknikkene for trefase vekselrettere; Space Vector Pulse Width Modulation (SVPWM) og Sinusoidal Pulse Width Modulation (SPWM).

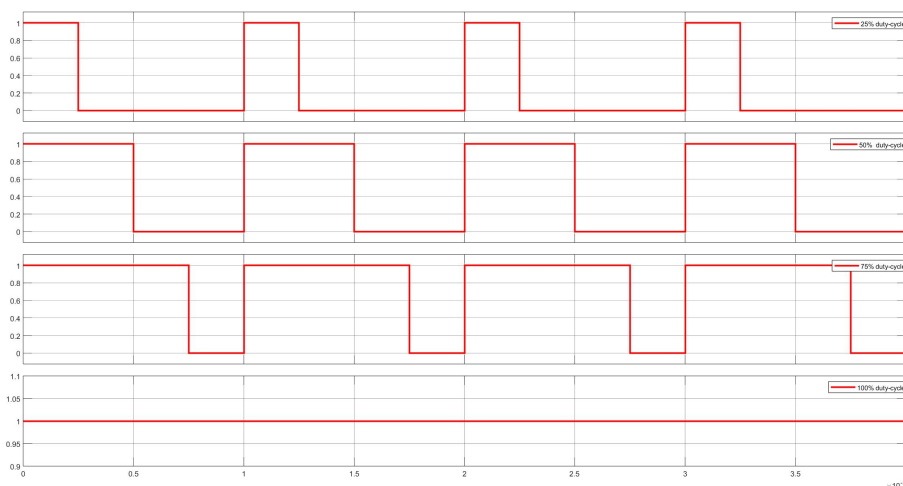
## 4.1 Puls-bredde modulasjon (PWM)

Puls-bredde modulasjon forkortes på engelsk til PWM og står for Pulse Width Modulation. Teknikken går i praksis ut på å styre hvor lenge et firkantpuls-signal er 'på' i løpet av periodetiden til signalet. Med firkantpuls menes et signal som har to tilstander; av (lav) eller på (høy).

Et sentralt begrep når det kommer til puls-bredde modulasjon er arbeidssyklus, eller på engelsk 'duty-cycle'. Arbeidssyklus er definert som på-tid delt på periodetiden til signalet:

$$D = \frac{t_{on}}{T_{sw}} \quad (4.1)$$

En fin måte å forstå arbeidssyklus på er å tenke på det som prosentdelen av periodetiden hvor signalet er på. Ergo; jo høyere arbeidssyklus, jo lengre er signalet på i løpet av en periode. Dersom arbeidssyklusen er 1 vil det si at signalet alltid er på, og dersom den er 0 vil det si at signalet alltid er av. I figur 4.1 er det illustrert fire firkantpulser med ulik arbeidssyklus men samme periodetid (20ms) og frekvens (50Hz).



Figur 4.1: Puls-bredde modulasjon (PWM)

*Kilde: laget i MatLab SimuLink*

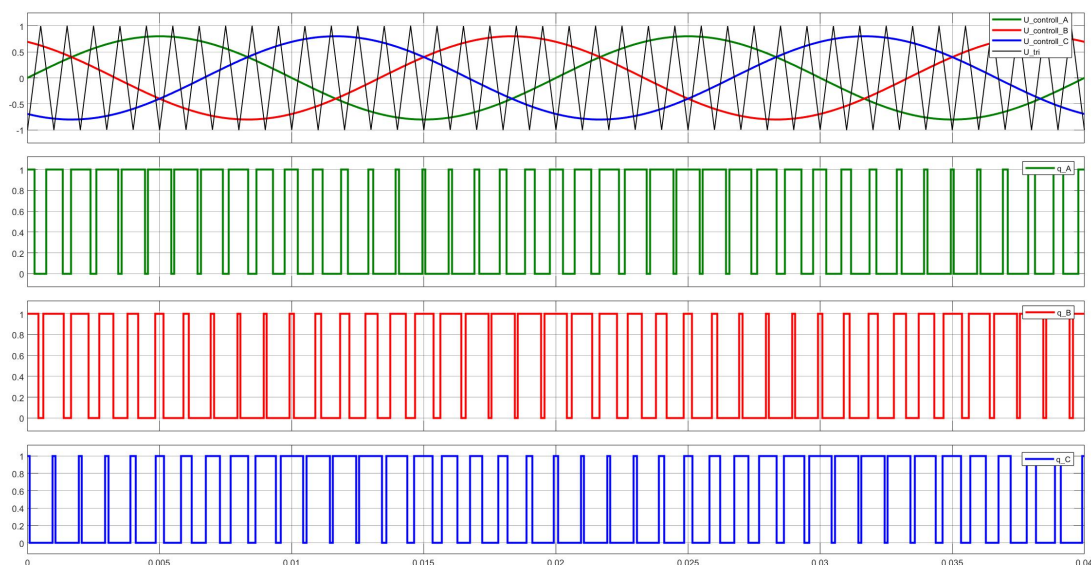
I forbindelse med kraftelektronikk er PWM mye brukt for å generere styresignal til krafttransistorer i omformere. Ved hjelp av en styringsenhet, for eksempel en microcontroller, kan en enkelt justere på arbeidssyklusen til styringssignalet som sendes inn på gate til en transistor. På denne måten styres transistoren, som igjen styrer selve omformeren.



## 4.2 Sinus PWM (SPWM)

Sinus PWM er en form for puls-bredde modulasjon som benyttes til å generere styresignal til bryterene i vekselrettere. Kort oppsummert fungerer modulasjonsteknikken ved å sammenligne én eller flere sinuskurver med en triangelkurve. I dette kapitlet vil vi ta for oss styring av en trefaset vekselretter med bipolar styring.

Trefase SPWM fungerer ved at tre sinuskurver sammenlignes med en og samme triangelkurve  $U_{tri}$  med frekvens  $f_{sw}$ . De tre sinuskurvene er 120 grader forskjøvet fra hverandre i tid for å oppnå riktig faseforskyvning på den avgitte spenningen. Hvert av de tre beinene i vekselretteren, bein A B og C, har hver sin tilhørende sinuskurve  $U_{control_A}$ ,  $U_{control_B}$  og  $U_{control_C}$ . Når momentanverdien til  $U_{control_x}$  er større enn momentanverdien til  $U_{tri}$  blir styresignalet til bein  $x$  satt logisk høy (øvre transistor i bein  $x$  leder), mens når momentanverdien til triangelkurven er større enn  $U_{control_x}$  blir styresignalet satt logisk lav (nedre transistor i bein  $x$  leder) [5]. I figur 4.2 er det forsøkt å illustrere dette.



Figur 4.2: Generering av styringssignaler for trefaset SPWM med bipolar kontroll

*Kilde: laget i MatLab SimuLink*

Frekvensen til de sinusformede signalene vil i praksis bestemme frekvensen til den *grunnharmoniske* avgitte spenningen fra vekselretteren. Denne frekvensen refereres ofte til som *modulasjonsfrekvens*. Ved å regulere modulasjonsfrekvensen kan en derfor oppnå ønsket frekvens på avgitt grunnharmonisk spenning. Frekvensen til triangelkurven  $U_{tri}$ ,  $f_{sw}$ , vil avgjøre svitsjefrekvensen til transistorene i vekselretteren. Ved svitsjefrekvens menes det enkelt og greit hvor mange ganger transistorene skrur seg av og på i løpet av et sekund [5].

En sentral parameter i forbindelse med sinus PWM er amplitude-modulasjonsforholdet  $m_a$ . Denne parameteren beskriver forholdet mellom amplituden til de sinusformede kontrollsignalene og amplituden til triangelkurven  $U_{tri}$ . Amplitude-modulasjonsforholdet er

definert som:

$$m_a = \frac{U_{controll}}{U_{tri}} \quad (4.2)$$

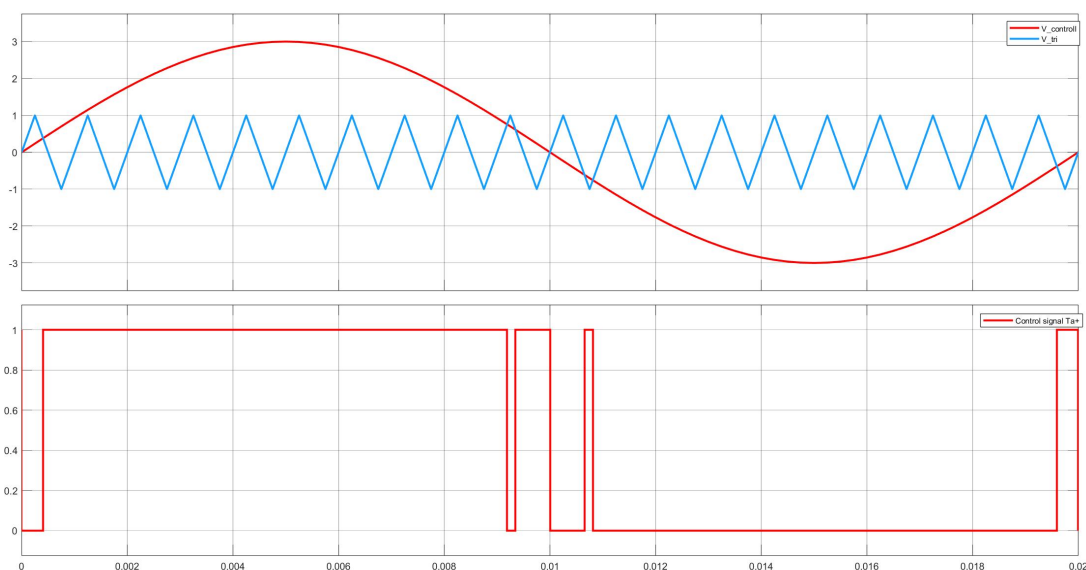
Amplitude-modulasjonsforholdet er avgjørende for hva den avgitte spenningen fra omformeren blir [5]. Avgitt grunnharmonisk fasespenning fra en trefaset omformer ved lineær SPWM ( $m_a \in [0, 1]$ ) kan beregnes ut fra følgende formel:

$$V_{AN1} = m_a \cdot \frac{V_d}{2\sqrt{2}} \quad (4.3)$$

Og avgitt grunnharmonisk linjespenning:

$$V_{LL1} = \sqrt{3} \cdot V_{AN1} = \sqrt{3} \cdot m_a \cdot \frac{V_d}{2\sqrt{2}} \quad (4.4)$$

Dersom vi øker amplitude-modulasjonsforholdet  $m_a$  til å være større enn 1, vil avgitt spenning begynne å ligne mer og mer en firkantpuls jo høyere  $m_a$  blir. Et slikt driftstilfelle kalles for **overmodulasjon**. Dette er forsøkt illustrert i figur 4.3 som viser et driftstilfelle hvor  $U_{controll} \gg U_{tri}$ . RMS-verdien til den grunnharmoniske komponenten avgitt spenning vil være høyere ved overmodulasjon enn ved lineær SPWM. Til tross for dette vil avgitt spenning ha et høyere harmonisk innhold ved overmodulasjon. Det er også vanskelig å regulere avgitt spenning ved overmodulasjon. Derfor er det ikke alltid forsvarlig med overmodulasjon til tross for at det gir høyere utgangsspenning [5].



Figur 4.3: Overmodulasjon på en fase vekselretter med bipolar styring

*Kilde: laget i MatLab SimuLink*

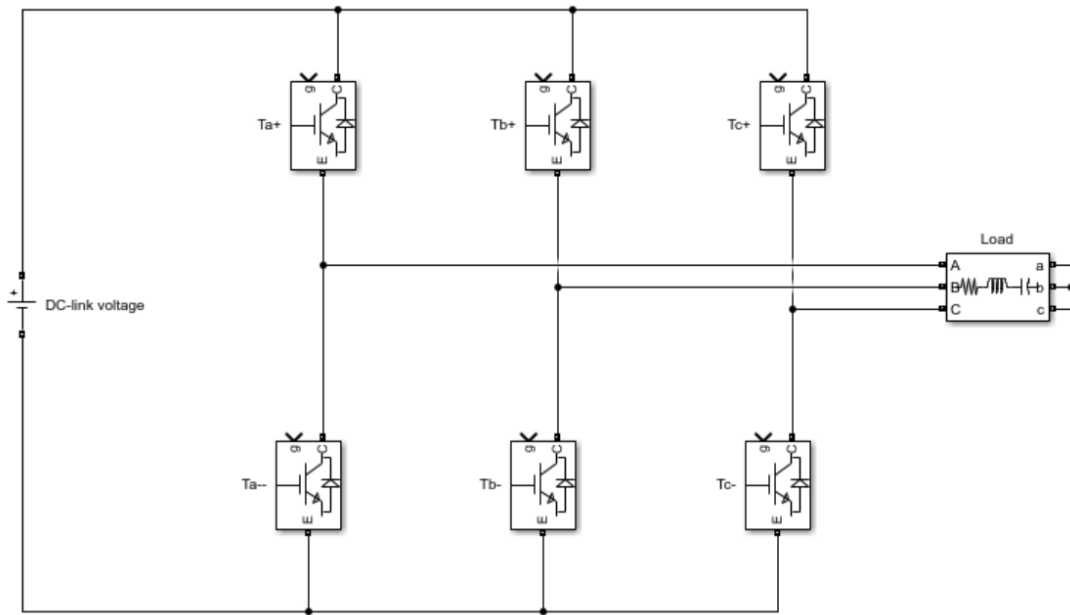
En annen viktig parameter ved SPWM er frekvens-modulasjonsforholdet  $m_f$ . Dette er også et forholdstall, og beskriver forholdet mellom svitsjefrekvens  $f_{sw}$  og modulasjonsfrekvens  $f_1$  [5].

$$m_f = \frac{f_{sw}}{f_1} \quad (4.5)$$

Når en skal dimensjonere en vekselretter er det anbefalt å velge  $m_f$  til å være et oddetall og et heltall, da dette resulterer i at partall-harmoniske komponenter i utgangsspenningen forsvinner som følge av at utgangsspenningen får odd og halvbølge symmetri [5, s. 206]. De harmoniske komponentene i utgangsspenningen vil ligge seg som sidebånd rundt svitsjefrekvensen og som sidebånd rundt multipler av svitsjefrekvensen. For å enkelt kunne filtrere vekk harmoniske komponenter i utgangsspenningen er det ønskelig å ha høyest mulig svitsjefrekvens, men høyere svitsjefrekvens betyr også høyere svitsjetap i transistorer [5, s. 207]. En er derfor nødt til å gjøre en vurdering ut fra hva en egentlig prøver å oppnå.

### 4.3 Space Vector PWM

Space vector PWM er en modulasjonsteknikk som benyttes i trefase vekselrettere. Med romvektor (space vector) menes i praksis vektorsummen av de tre spenningene i et trefase-system. Et symmetrisk og balansert trefasesystem vil produsere en roterende romvektor som følge av at de tre fasene er 120 grader faseforskjøvet fra hverandre. Figur 4.7 illustrerer avgitt romvektor i et øyeblikk. Det er dette fenomenet som skaper et roterende magnetfelt i statoren til en elektrisk motor. Ved Space Vector PWM kontrolleres den roterende romvektoren direkte ved å svitsje transistorene i vekselretteren på en bestemt måte.



Figur 4.4: Trefase vekselretter topologi

*Kilde: laget i MatLab SimuLink*

Styresignalet til de to transistorene på samme bein er alltid invertert i forhold til hverandre for å unngå kortslutning. Dette gjør at vi kan beskrive tilstanden til hvert bein som et binærtall hvor 1 tilsier at øvre bryter er på (og nedre bryter er av), og 0 tilsier at nedre bryter er på (og øvre bryter er av). Og siden vi har 3 bein i en trefase vekselretter, med 2 ulike tilstander per bein, får vi  $2^3 = 8$  ulike brytertilstander. La  $q_a$  være tilstanden til bein a,  $q_b$  være tilstanden til bein b og  $q_c$  være tilstanden til bein c. Avgitt romvektor fra vekselretter til lasten kan dermed beskrives som en funksjon av brytertilstandene [1, s. 120].

$$\vec{v}_s(q_c, q_b, q_a) = V_d [q_a e^{j0} + q_b e^{j\frac{2\pi}{3}} + q_c e^{j\frac{4\pi}{3}}] \quad (4.6)$$

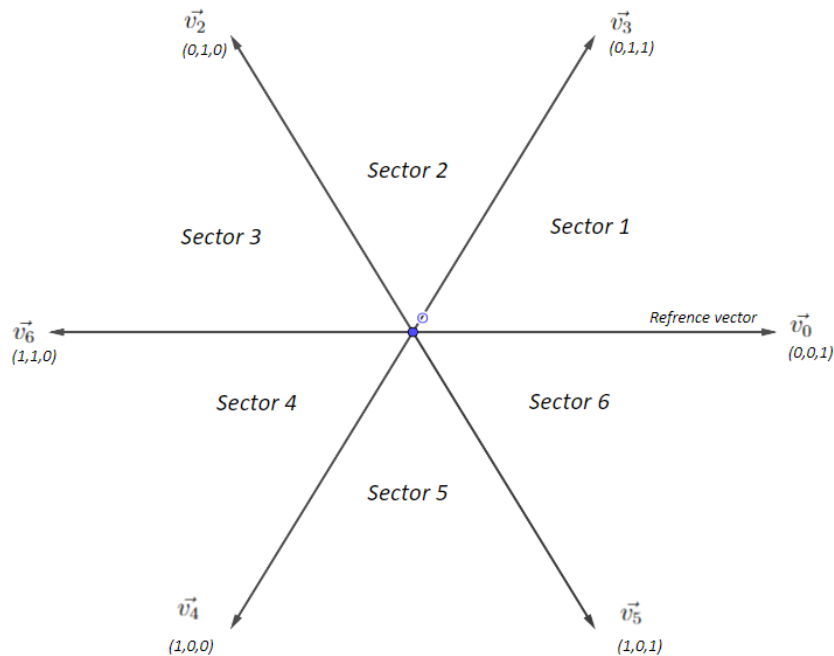
Ligning 4.6 kan skrives om til

$$\vec{v}_s(q_c, q_b, q_a) = V_d [q_a \underline{0} + q_b \underline{+120} + q_c \underline{-120}] \quad (4.7)$$

Setter vi inn de 8 ulike brytertilstandene i ligning 4.7 får vi:

$$\begin{aligned}\vec{v}_0 : v_s(0, 0, 0) &= V_d[0/0 + 0/+120 + 0/-120] = 0 \\ \vec{v}_1 : v_s(0, 0, 1) &= V_d[1/0 + 0/+120 + 0/-120] = V_d/0 \\ \vec{v}_2 : v_s(0, 1, 0) &= V_d[0/0 + 1/+120 + 0/-120] = V_d/+120 \\ \vec{v}_3 : v_s(0, 1, 1) &= V_d[1/0 + 1/+120 + 0/-120] = V_d/60 \\ \vec{v}_4 : v_s(1, 0, 0) &= V_d[0/0 + 0/+120 + 1/-120] = V_d/-120 \\ \vec{v}_5 : v_s(1, 0, 1) &= V_d[1/0 + 0/+120 + 1/-120] = V_d/-60 \\ \vec{v}_6 : v_s(1, 1, 0) &= V_d[0/0 + 1/+120 + 1/-120] = V_d/180 \\ \vec{v}_7 : v_s(1, 1, 1) &= V_d[1/0 + 1/+120 + 1/-120] = 0\end{aligned}$$

Figur 4.5 viser vektordiagrammet som de 8 romvektorene danner.



Figur 4.5: Vektordiagram for de 6 aktive vektorene ved SVPWM

*Kilde: laget i GeoGebra og Inkscape*

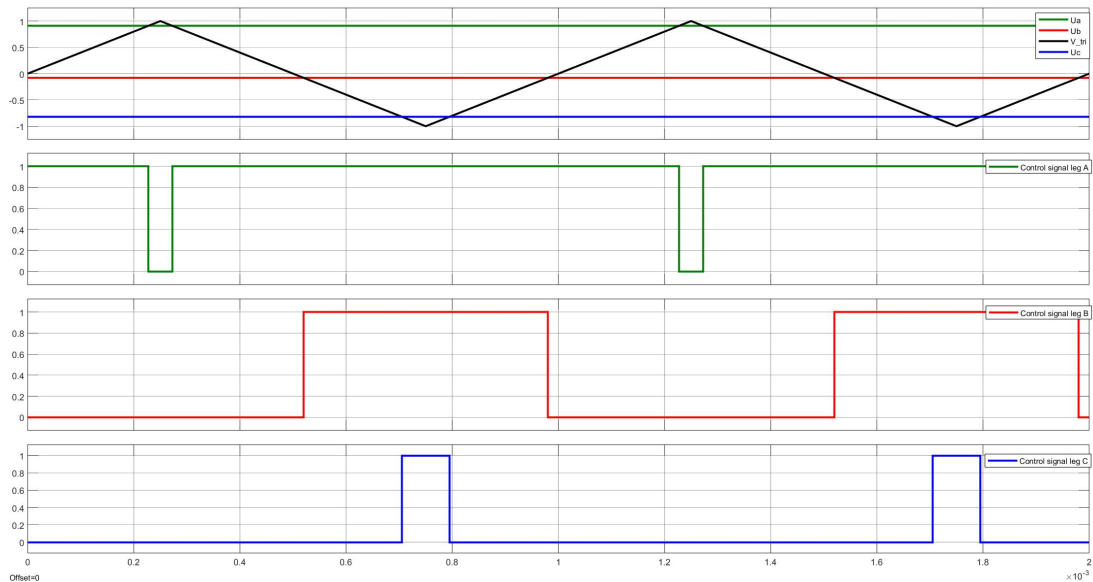
Viktige observasjoner her er først og fremst at lengden til hver av disse vektorene er DC-link spenningen  $V_d$ . Det er også viktig å merke seg at  $\vec{v}_1$  er referansevektoren som alle vinklene refereres i forhold til, og det er 60 grader mellom vektorene i samme sektor. I figur 4.5 er det kun de aktive vektorene, altså de romvektorene som har en lengde ulik null, som er tatt med. Vi har også to passive vektorer som gir ut 0V, nemlig vektorene  $\vec{v}_0$  og  $\vec{v}_7$ . Disse vektorene benyttes for å legge inn dødtid dersom en ikke ønsker å gi ut maksimal spenning fra omformereren [1].

Generelt sett fungerer modulasjonsteknikken ved at tre kontrollsignaler sammenlignes, et kontrollsignal per bein i vekselretteren, med en triangelkurve  $U_{tri}$ . La oss kalle kontrollsignalene  $U_a$ ,  $U_b$  og  $U_c$ . La oss for enkelhetens skyld si at allverdien til kontrollsignalene samt triangelkurven ligger mellom -1 og 1. For kontrollsignalene er det slik at -1 representerer at den gjeldende fasen tilknyttet det gjeldende beinet har sin negative maksverdi, 0 tilsier nullgjennomgang og 1 tilsier at den har sin toppverdi. Når kontrollsignalet er større enn momentanverdien til triangelkurven så settes øvre transistor til det tilhørende beinet på, og når momentanverdien til triangelkurven er større enn kontrollsignalet settes nedre transistor i beinet på. Generelt:

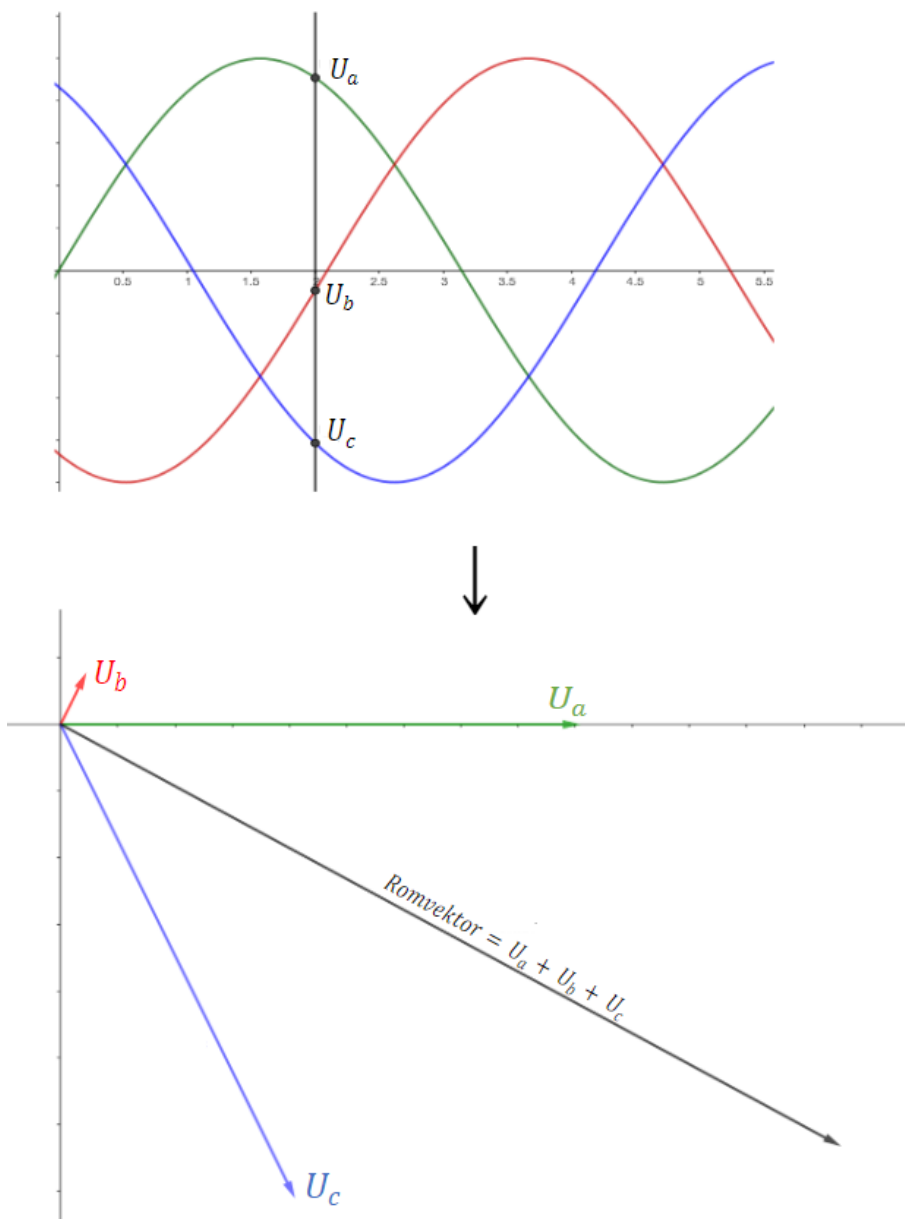
$$U_n > U_{tri} \rightarrow q_n = 1$$

$$U_n < U_{tri} \rightarrow q_n = 0$$

hvor  $U_n$  er kontrollsignalet til bein  $n$  og  $q_n$  er brytertilstanden til bein  $n$ . I figur 4.6 er det vist et eksempel hvor kontrollsignalene er som følge:  $U_a = 0.91$  ,  $U_b = -0.09$  og  $U_c = -0.82$ .



Figur 4.6: Kontrollsignal for SVPWM ved et gitt øyeblikk  
*Kilde: laget i MatLab SimuLink*



Figur 4.7: Romvektor i et gitt øyeblikk i et symmetrisk trefasessystem  
 Kilde: laget i GeoGebra og Inkscape

For å avgi en gjennomsnittlig roterende romvektor med fasekomponenter  $u_a$ ,  $u_b$  og  $u_c$  kan en uttrykke kontrollsignalene som beskrevet i 4.8, 4.9 og 4.10 [1, s. 124].

$$\frac{U_{control_A}}{U_{tri}} = \frac{u_a - u_k}{\frac{1}{2} \cdot U_d} \quad (4.8)$$

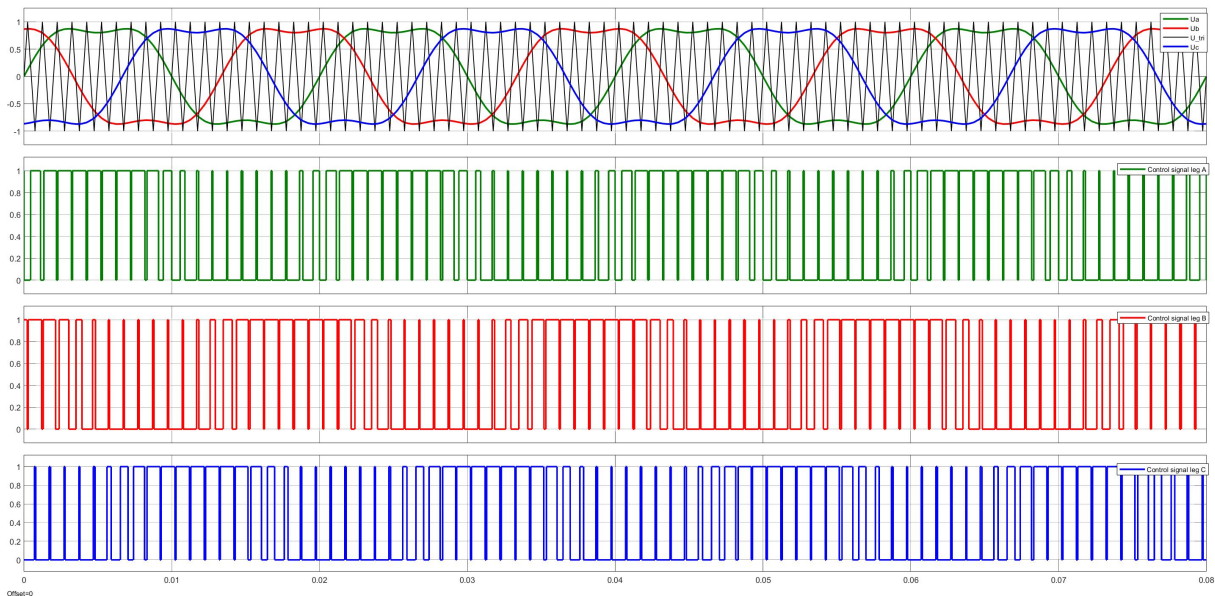
$$\frac{U_{control_B}}{U_{tri}} = \frac{u_b - u_k}{\frac{1}{2} \cdot U_d} \quad (4.9)$$

$$\frac{U_{control_C}}{U_{tri}} = \frac{u_c - u_k}{\frac{1}{2} \cdot U_d} \quad (4.10)$$

hvor

$$u_k = \frac{\max(u_a, u_b, u_c) + \min(u_a, u_b, u_c)}{2} \quad (4.11)$$

Figur 4.8 viser styresignalene ved et slikt tilfelle.



Figur 4.8: Simulerte styresignaler for SVPWM  
Kilde: laget i MatLab SimuLink



## Del II

# Utførelse

# Kapittel 5

## Analyse av oppgaven

Dette kapitlet tar for seg analysen som vi gjorde av problemstillingen på forhånd av utviklingsfasen i prosjektet. Kapitlet er delt opp i fire ulike deler. Den første delen gjennomgår en analyse av hardware som den forrige gruppen utviklet våren 2023. Den andre og tredje delen tar for seg vår analyse av hvordan vi skulle forbedre og arealeffektivisere kretskortet til den tidligere gruppen, og den siste delen tar for seg planleggingen av softwareutvikling.

### 5.1 Analyse av tidligere design

For å kunne videreutvikle kretskortet som ble utviklet våren 2023 er det helt essensielt med en grundig analyse av det tidligere designet. Dette innebærer analyse av komponentvalg som har blitt gjort av den tidligere gruppen. For å forstå komponentene har vi brukt rapporten som den tidligere gruppen skrev [2] i tillegg til databladene for de forskjellige komponentene. Som forklart i problemstillingen så vil ikke vårt fokus ligge på å forbedre hardware ved å gjøre endringer i komponentvalg, da den tidligere gruppen har bevist at komponentvalgene de gjorde har resultert i et robust og velfungerende design. Vårt fokus i dette bachelorprosjektet vil heller være å optimalisere designet til selve kretskortet.

#### 5.1.1 Elektriske parameter for omformer

Som følge av at vi vil benytte de samme komponentene som ble brukt i det originale designet, vil de elektriske parametrene til omformeren forbli uendret. De elektriske parametrene til omformeren er listet opp i tabell 5.1, og er hentet rett ut fra rapporten til den tidligere gruppen [2, s. 52].

Tabell 5.1: Elektriske parameter til omformer

Parameter	Nominell verdi	Maksimal verdi
DC-link spenning	$565V_{DC}$	$1000V_{DC}$
Avgitt linjespenning	$400V_{RMS}$	-
Avgitt linjestrøm	$5A_{RMS}$	-
Modulasjonsfrekvens	$50Hz$	$100Hz$
Svitsjefrekvens	$> 4kHz$	-

## 5.1.2 Komponentvalg

### Transistorer

Transistorene i vekselretteren ble valgt av den tidligere gruppen til å være IGBT-transistorer av typen *Infineon IKW40N120H3* [2, s. 63]. I følge databladet kan transistoren operere med kollektor-strømmer opp mot 40A, og jobbe med svitsjefrekvenser mellom 20-100kHz. Maksimal påtrykt spenning mellom kollektor og emitter er  $V_{CEmax} = 1200V$ . Den har innebygget friløps-diode, noe som er en stor fordel. Sammenligner vi disse verdiene opp mot de elektriske parametrene til omformerer i tabell 5.1 ser vi at denne transistoren oppfyller disse kravene med god margin.

### Kjøleribbe til IGBT

Som forklart i kapittel 2 blir det omsatt effekt i en transistor mens den leder strøm samt under svitsjing. Denne effekten blir i praksis omgjort til varme som fører til opphetning av transistoren. Det er derfor essensielt med tilstrekkelig kjøling av transistoren for å unngå overopphetning som kan føre til redusert levetid og i verste fall havari. Dette gjøres vanligvis ved å koble transistoren mekanisk til en kjøleribbe som øker det totale overflatearealet til komponenten slik at varme avgis mer effektivt til omgivelsene. Det ble utført en grundig analyse av kjølebehovet til IGBT transistorene av den tidligere bachelorgruppen våren 2023 [2]. I analysen ble det tatt hensyn til ledetap og svitsjetap i selve transistoren samt den antiparallele dioden, og det ble lagt til grunn en kollektor-strøm på 5A. Det ble beregnet at kjølebehovet var maksimalt 9.13K/W [2, s. 73]. På basis av dette valgte den tidligere gruppen å benytte kjøleribber av typen *Ohmite C247-075-3AE*, som har en minimal termisk motstand på 6K/W [2, s. 73]. Dette er en dobbel kjøleribbe, og det vil dermed kunne benyttes én slik kjøleribbe per bein i omformerer (altså to transistorer per kjøleribbe).

### DC-link kondensatorer

For å minimere rippel DC-link spenningen er det vanlig å benytte filterkondensatorer. Størrelsen til denne kapasitansen dimensjoneres ut fra hvilke rippelspenning en tolererer på DC-linken, og hvilke svitsjefrekvens omformerer opererer med [5]. Under dimensjonering av DC-link kondensatorer i det originale designet ble det lagt til grunn en svitsjefrekvens på 5kHz samt en maksimal rippelspenning på 1 prosent av nominell DC-link spenning (565V, ref 5.1) [2, s. 73]. Den tidligere gruppen valgte å benytte to separate kondensatorer koblet i serie, for å på denne måten skape et nullpunkt mellom kondensa-



### 5.1.3 Kraftforsyninger

Hovedforsyningen til elektronikken ble valgt av den tidligere gruppen til å være 24V DC [2]. Samtlige komponenter krever et spenningsnivå som er ulikt fra dette, og det er derfor nødvendig med diverse DC/DC omformere som forsørger riktig spenningsnivå til riktig komponent. Vi vil i dette delkapittelet gå gjennom hvilke omformere som er valgt, og til hvilket formål de er brukt.

#### Forsyning til gate-drivere

Gate-driveren krever en forsyning på +15V og -8V på kraftsiden for å drive IGBT-transistoren. For å forsørge dette trenger vi derfor DC/DC omformere. Siden 24V forsyning (som forsyner denne omformeren) ligger på signalsiden av kortet men denne omformeren skal forsyne på kraftsiden, er omformeren nødt til å ha galvanisk skille mellom input- og output. Det er her valgt en DC/DC omformer av typen *MGJ2D241509SC* [2]. Denne omformeren tar inn +24V på input siden og gir ut +15V og -8,7V i tillegg til at den har innebygget galvanisk skille. Hver gate-driver vil trenge 2 stk *MGJ2D241509SC*, da high-side og low-side av Infineon 2ED020I12-F2-chippen krever hver sin forsyning. Dette er illustrert i figur 5.1.

#### Uisolert 5V forsyning signalside

Som forsyning til alle komponentene som krever 5V på signalsiden av kretskortet er det valgt en uisolert 24V til 5V DC/DC omformer av typen *Multicomp Pro MPK7805T-500R3* [2, s. 62]. Det er ikke behov for galvanisk skille i denne omformeren da den er lokalisert på signalsiden av kretskortet, som er den samme siden som 24V forsyningen. Multicomp Pro MPK7805T-500R3 har innebygget kortslutnings-beskyttelse, høy virkningsgrad og er overflatemontert. Dette gjør den til et solid valg.

#### Isolert 5V forsyning kraftside

Som 5V forsyning til spenningsmålings-systemet på kraftsiden er det valgt en isolert 5V til 5V DC/DC omformer av typen *Murata NTE0505MC* [2, s. 63]. Denne omformeren vil ligge på isolasjonsbarrieren av kortet med input-siden av chippen på signalsiden og output-siden på kraftsiden.

### 5.1.4 Microcontroller

Det ble spesifisert i bachelorproppgaven [2, s. 39] at det skulle bli brukt en 32-bit microcontroller. Den aktuelle microcontrolleren som ble valgt av den forrige gruppen ble nøye utvalgt basert på flere faktorer, og vi vil benytte den samme microcontrolleren i det nye designet. Det er viktig å ha med seg at valget falt basert på disse punktene.

- Minst seks PWM utganger
- Det må være en analog inngang per strøm- og spenningsensor
- Innebygde timere som støtter avansert motor kontroll
- Det må være nok antall I/O porter i tilfelle utvidelse

### 5.1.5 Målesystemer

Det komplette produktet har spesifikke krav når det kommer til målinger av diverse elektriske parametre på kretskortet [2]. Parametrene som er ønskelig å ha måling på er:

- Strømmåling av DC-LINK
- Spenningsmåling over DC-link
- Måling av middelverdien og RMS strøm igjennom alle tre fasene på utgangen

#### Strømmåling

Som den tidligere bachelorgruppen [2, s. 49] har skrevet skal det være strømmålinger på både DC-linken sin hovedstrøm, og på alle de 3 utgående fasene. For å få til dette valgte den tidligere gruppen å bruke strøm-målingssensoren **TLI4971A050T5UE0001XUMA1** fra *Infineon Technologies*. Denne komponenten muliggjør målinger i området +/- 50A ved en driftsspenning på 3.3V. Den sender et analogt signal til ADC (Analog to Digital Converter) i mikrokontrolleren. I tillegg har den to overstrøms-målingsfunksjoner, OCD1 og OCD2, som oppdager overstrømmer. Disse utgangene gir funksjoner som mikrokontrolleren kan benytte for å opprette en sikkerhetsfunksjon som deaktiverer omformerens i tilfelle av overdreven strømtrekk.

Oppkoblingen av sensoren er gjort i henhold til anbefaling i datablad, der motstander og kondensatorer er valgt i forhold spesifikasjonene. Det er også designet en dedikert krets for å kunne måle en referansespenning til sensoren. Denne spenningen er viktig å kunne måle for å både positive og negative strømmer, som strømmer i begge retninger. ADC'en i mikrokontrolleren kan behandle spenninger fra 0V-3.3V. Derfor er det satt opp en spenningsdeler som gir 1.65V til  $V_{REF}$  pin [2, s. 65]. Dersom det ikke går strøm igjennom strømsensoren vil  $A_{OUT}$  pin ha en spenning på  $V_{REF} = 1.65V$ . Den analoge utgangen vil skaleres i henhold til produsentens datablad i enhver økning eller reduksjon i spenningsnivå. Denne økningen eller reduksjonen måles i området millivolt per ampere.

#### Spenningsmåling for måling av DC-link spenning

Det skal i nominell drift måles en DC-link spenning på 565V DC. For å måle Spenningen, må den bli gjort om til et nivå som fungerer med mikrokontrollerens ADC. For å få til denne løsningen ble det funnet ut av den tidligere bachelorgruppen at det måtte settes opp en spenningsdeler her også, og at det måtte være et galvanisk skille mellom kraftsiden og signalsiden på kretskortet [2, s. 65–66]. Det ble da bestemt at **ACPL-C87B-000E** fra *Broadcom* skulle brukes til å oppnå det galvaniske skillet. Denne chippen trenger 5V på begge sider, og det ble derfor bestemt at det skulle brukes en isolert DC-DC omformer til dette. Videre viste det seg at det er viktig å inkludere en operasjonsforsterker for å oppnå riktig inngangsspenning for ADC'en i mikrokontrollern.

## 5.2 Analyse av kretskortdesign

Som problemstillingen i 0.4 tilsier er et av hovedelementene i denne bacheloroppgaven å redusere fysisk arealbruket på kretskortet til omformerer ved å designe et nytt og mer arealeffektivt kretskort. Programvaren som er valgt til design av kretskort er KiCad.

### 5.2.1 Krav til kretskort

Ved utledning av kravene til kretskortet er sikkerhet og funksjonalitet hovedfokus. Sammen med veileder ble det enighet om at vi skulle forholde oss til følgende overordnede krav til kretskortet:

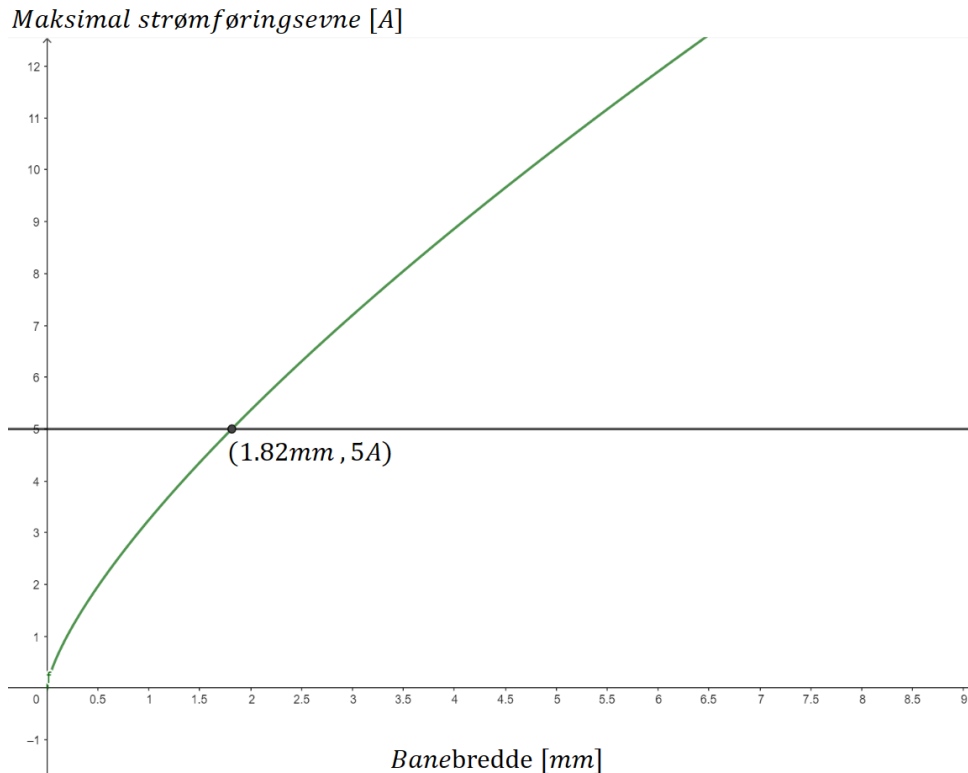
- Elektrisk isolasjonsbarriere mellom signalsiden og kraftsiden av kretskortet
- Mest mulig identisk lengde på kobberbanene til fasene og DC-link
- Tykkest mulig baner for fasene og DC-link
- Kobberlag tilkoblet jord over hele signalsiden for EMI-skjold
- Styresignal til transistorene prioriteres over alle andre signaler på kretskortet
- Skru-hull i hjørnene for enkelt feste

Det første kravet om elektrisk isolasjonsbarriere på kretskortet er kanskje det viktigste. Med signalside menes den siden av kretskortet hvor styringsenheten er plassert, i dette tilfellet microcontrolleren, hvor det primært sett kun går signaler og ingen sterke strømmer. Med kraftsiden menes siden hvor selve vekselretteren og sterkstrømsbanene er plassert. For å kunne opprettholde dette kravet er det essensielt at komponentene som skal ligge på isolasjonsbarrieren, som primært sett er gatedriverne som styrer transistorene, har innebygget galvanisk skille slik at den siden av chippen som ligger på signalsiden er elektrisk isolert fra siden som ligger på kraftsiden. Gatedriverne og resten av komponentene som ligger på isolasjonsbarrieren har denne egenskapen, dette er beskrevet i 5.1.

To av kravene gir spesifikasjoner for hvordan banene til sterkstrømskoblingene (DC-link tilkobling og fasene) skal være. Det første sier at det bør være mest mulig identisk lengde på banene til fasene, og dette er enkelt å greie for at impedansen i de tre fasene skal bli mest mulig identisk for å unngå ubalanse. Det andre kravet sier at tykkelsen på sterkstrømsbanene skal være tykkest mulig, og dette er for å minimere varmeutvikling i kretskortet under drift av omformerer. KiCad tilbyr en kalkulator for å beregne banetykkelse, og for å få en indikasjon på hva benetykkelsene *minst* måtte være så benyttet vi oss av denne. Kalkulatoren i KiCad beregner strømføringssevnen til PCB-banene ved hjelp av formel 5.1.

$$I = K \cdot \Delta T^{0.44} \cdot (W \cdot H)^{0.725} \quad (5.1)$$

I figur 5.1  $K = 0.048$  for eksterne baner,  $W$  er bredden på banen i mils,  $H$  er tykkelsen (høyden) til banen i mils, og  $\Delta T$  er temperaturøkningen i grader Celsius. Etter samtale med veileder ble det besluttet å dimensjonere ut fra  $\Delta T = 20C$  og  $H = 0.035$  mm. Plotter vi strømføringssevnen som funksjon av banebredde med disse parametrene blir resultatet som vist i figur 5.2. Figur 5.2 forteller oss at den minste banebredden som sterkstrømsbanene kan ha er 1.82mm dersom omformerer skal kunne føre strøm på 5A.



Figur 5.2: Maksimal strømføringssevne som funksjon av banebredde  
*Kilde: laget i GeoGebra*

Dette er et absolutt minimumsverdi, i praksis er det ønskelig med betydelig bredere baner enn dette for å øke sikkerhetsmarginene.

## 5.2.2 Potensiale for arealeffektivisering av PCB

For å arealeffektivisere kretskortet er det mange ulike tiltak en kan gjøre. Før vi tok for oss design av kretskortet i KiCad gjorde vi en grundig analyse om hvordan vi kunne arealeffektivisere samtidig som kravene nevnt i 5.2.1 ble ivaretatt. Vi vil her gjennomgå tankeprosessen bak planleggingen av kretskortet.

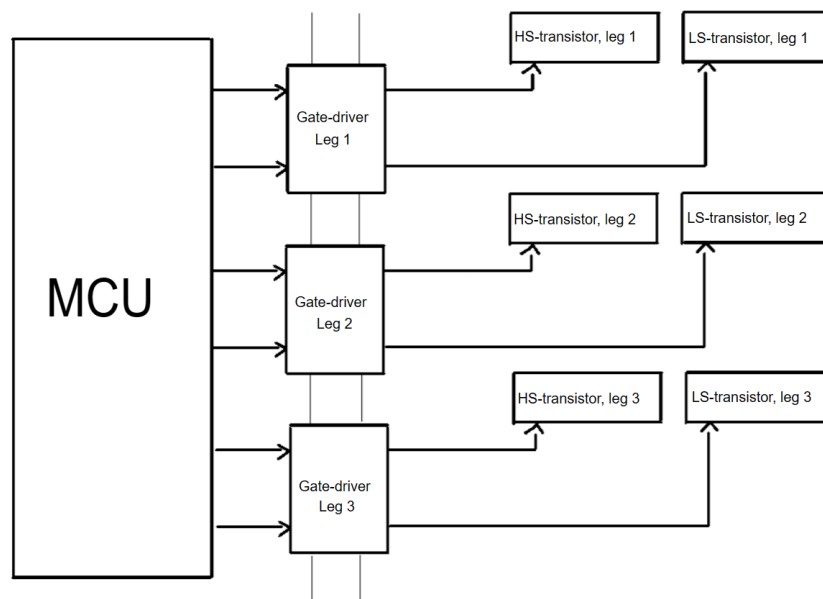
### Komponentplasseringer

En av hovedgrunnene til at kretskortet som ble utviklet våren 2023 er mindre arealeffektivt enn hva det kunne vært er at de aller fleste komponentene på kretskortet ble plassert på oppsiden av kortet. Dette førte til at undersiden av kortet ble så og si ubrukt, noe som utelater et stort område. Denne observasjonen gjorde vi oss tidlig under planleggingsfasen. Etter å ha studert flere kommersielle kretskort samt drøftet med veileder innså vi at en god løsning for å utnytte plass på kortet er ved å plassere de minste SMD-komponentene (motstander, kondensatorer, chipper og så videre) på undersiden av kretskortet, mens de større komponentene (microcontroller, kjøleribber med IGBT osv.) plasseres på oppsiden.

Det siste kravet i 5.2.1 som omhandler banene til styringssignalene medførte at vi måtte planlegge nøye hvordan vi skulle plassere transistorene i forhold til gatedriver-chippene.



Som kravet tilsier skal disse signalene prioriteres ovenfor alle andre signal på kortet. Derfor tenkte vi at det var lurt å starte planleggingen med å finne den mest ideelle plasseringen av gate-driverene og transistorene for å tilfredsstille dette kravet. Deretter kunne vi tilpasse resten av kortet ut i fra dette. Figur 5.3 viser et blokkdiagram av styringskretsen til transistorene. Ved å plassere transistorene i samme bein ved siden av hverandre på denne måten vil en oppnå korte baner mellom gatedriver og gate på transistor. Som figur 5.3 prøver å illustrere så vil gatedriver-chippene, som nevnt tidligere, ligge midt på isolasjonsbarrieren av kortet. Det er derfor ønskelig å plassere microcontrolleren nærmest mulig isolasjonsbarrieren for å få kortest mulig bane mellom MCU og gate-driver. Det samme gjelder kjøleribbene med transistorer; det vil være mest gunstig om de ligger nærmest mulig isolasjonsbarrieren.



Figur 5.3: Blokkdiagram av styringskrets for transistorer  
*Kilde: laget med ReMarkable*

### Fjerning av HMI og bremsekrets fra hovedkort

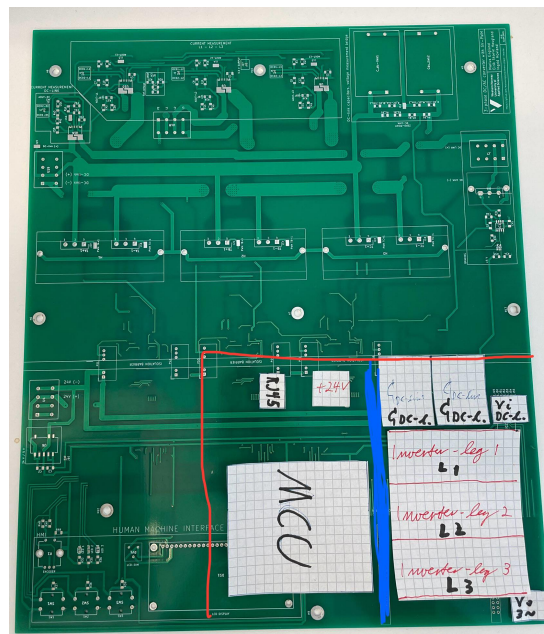
Sammen med veileder besluttet vi tidlig at det var gunstig å ha et eget kretskort for HMI som en kan tilkoble hovedkortet via kabel. Dette ble ikke gjort på kretskortet som ble utviklet våren 2023, her var HMI på samme kretskort som vekselretteren. Ved å ha HMI på et eget kretskort blir naturligvis kretskortet til omformeren fysisk mindre. I tillegg bidrar det til økt personsikkerhet da dette gjør at en unngår å ta på kretskortet til omformeren mens omformeren er spenningsatt. Vi bestemte oss for å benytte en RJ45-kontakt på signalsiden av kretskortet som microcontrolleren kobles opp mot. På denne måten kan en enkelt koble til et eksternt HMI-kort via en nettverkskabel som styres av den samme microcontrolleren som resten av hovedkortet.

På kretskortet som ble utviklet våren 2023 var bremsekretsen for håndtering av overflødig energi fra re-generativ drift av motor plassert på kraftsiden hovedkortet sammen med omformeren. Vi besluttet sammen med veileder at denne kretsen skulle være på et eget

kretskort. Kort fortalt fungerer denne kretsen ved at en kontinuerlig måler DC-link spenningen, og dersom den overstiger en gitt verdi så kobles det inn en motstand i parallell med DC-link kondensatoren som omgjør energien som motoren leverer til omformeren til varme [2]. For å kunne ha denne kretsen på et eget kretskort ble det derfor nødvendig med en ekstra tilkoblings-terminal til DC-linken. Dette problemet kunne vi løse ved å bruke en 4-terminals Wago tilkoblings-klemme til tilkobling av DC-link som hadde to separate pluss- og minuspoler, slik at både DC-link forsyning og bremsekrets kunne tilkobles samme punkt.

### Utvikling av prototype for kretskort

Vi fikk tidlig tilgang til et eksemplar av kretskortet som ble utviklet våren 2023. For å teste ut ulike kombinasjoner av komponentplasseringer klippet vi ut papir-biter av de største komponentene i samme mål som den fysiske footprinten til komponentene. Disse papirbitene brukte vi så til å teste ulike utlegg av kretskortet, og vi brukte et eksemplar av kretskortet utviklet av den tidligere gruppen til å visualisere potensialet for arealeffektivisering. Etter litt kom vi frem til det som vi mente var den mest arealeffektive løsningen, resultatet er vist i figur 5.4. Den blå streken i figur 5.4 viser hvor vi planlagte å ha isolasjonsbarrieren på kortet. Denne prototypen er vell å merke kun av fremsiden til kretskortet. Samtlige chipper, blant annet gate-driverene, ble som nevnt tidligere planlagt å plassere på baksiden av kortet, og er derfor ikke med i denne prototypen. Dersom en studerer blokkdiagrammet i figur 5.3 kan man se at prototypen i figur 5.4 er mye inspirert av filosofien til blokkdiagrammet.



Figur 5.4: Prototype av kretskort til omformer  
*Kilde: bilde tatt med iPhone*

Reint designmessig bestemte vi oss for at det ville være logisk at alle input-tilkoblingene (DC-link tilkobling, RJ45-kontakt og 24V-forsyning) skulle være på oppsiden av kretskortet, mens output (3 faser + PE) skulle være på nedsiden av kretskortet. Denne filosofien holdt vi oss til gjennom hele designprosessen. Planen var at vi på et senere tidspunkt skulle designe en boks som kretskortet kunne plasseres i, og det ville derfor være naturlig at input og output er på hver sin side.

## 5.3 Utvikling av eget HMI-kort

Som forklart i 5.2.2 besluttet vi å fjerne HMI fra hovedkortet for å utvikle et eget kretskort for dette formålet. HMI er forkortelse for Human Machine Interface. HMI-kortet har som formål å styre hovedkortet, og virker i praksis som en kontroller som kan opereres manuelt av brukeren.

### 5.3.1 Krav til HMI-kort

Etter samtale med veileder ble det fastslått at HMI-kortet skulle inneholde følgende komponenter:

- 1x LCD-display
- 3x trykknapper
- 1x encoder
- I/O extender med 16 I/O
- 3x lysdioder
- RJ45-kontakt

For avlesning av målinger var det nødvendig med LCD-display, og for å ta inn inputs fra brukeren skulle vi bruke trykknapper og en encoder. Spesifikk funksjonalitet til trykknappene og encoderen ble ikke bestemt under planleggingsfasen av HMI-kortet, dette var noe vi skulle komme tilbake til når kortet skulle programmeres. Det ble enighet om at tre trykknapper og én encoder var nok for denne applikasjonen. Lysdioder skulle benyttes for å gi indikasjon om status til omformereren.

### 5.3.2 Komponenter til HMI-kort

For å kunne sammenkoble HMI-kortet og hovedkortet bestemte vi oss for å bruke RJ45-kontakt; én på HMI-kortet og én på hovedkortet. Vi valgte her å benytte RJ45-kontakt av typen *Hearting 09455511102*. Denne typen har totalt 8 pins, noe som er mer enn nok for denne applikasjonen. I/O ekspander var nødvendig for å gi flere tilgjengelige pins til microcontrolleren, og vi valgte her å benytte *MCP23016-I/SO*. Denne expanderen benytter I2C-kommunikasjon, noe som vil gå via RJ45. I2C-kommunikasjon krever kun 2 pins for signaloverføring, noe som gjorde at vi kun behøvde totalt 6 ledninger mellom HMI-kort og hovedkort.

Komponentene som er i fysisk kontakt med brukeren ble vi enig om at skulle være hullmonterte, da hullmonterte komponenter tåler fysiske påkjenninger bedre enn hva overflatemonterte komponenter gjør. Det ble derfor bestemt at trykknappene, encoder, LCD-display og RJ45 skulle være hullmonterte komponenter. Encoderen som ble valgt til denne applikasjonen var en helt standard encoder av typen *PEC11R-4220F-S0024*.

## 5.4 Analyse av software

En stor del av prosjektet omhandler videreutvikling av software. Våren 2023 ble det utviklet kode for trefase SPWM og SVPWM [2]. Målet vårt er å bygge videre på denne koden ved å muliggjøre styring og overvåking med HMI. I tillegg ønsker vi å legge til mulighet for drifte kretskortet i H-bro konfigurasjon som både enfaset vekselretter og fullbro DC/DC omformer. Følgende styringsalgoritmer er ønskelig å implementere i software:

- Trefse SPWM med bipolar styring
- Trefase SVPWM med bipolar styring
- Enfase SPWM med unipolar og bipolar styring
- Fullbro DC/DC omforming med bipolar og unipolar styring

For trefase SPWM og SVPWM vil vi ta utgangspunkt i den tidligere gruppen sitt arbeid. En mangel i koden til den tidligere gruppen var mulighet for enkel justering av avgitt frekvens. Koden som de utviklet våren 2023 fungerer ved at bruker manuelt skriver inn i koden hvilke frekvens og avgitt spenning som ønskes. Det vil derfor være et mål for oss å videreutvikle denne koden til at en kan justere avgitt frekvens og spenning fra HMI-kortet ved hjelp av encoder, og det er ønskelig å kunne lese av frekvens, DC-link spenning og strømmålinger på LCD-displayet på HMI-kortet. Siden det også er ønskelig å kunne drifte en DC-motor med omformerens konfigurert som H-bro, vil vi også ha som mål å kunne styre avgitt DC-spenning og dens polaritet via HMI.

# Kapittel 6

## PCB design i KiCad

Programvaren som er valgt til design av kretskort er KiCad. I kapittel 5.2 er planleggingsfasen av kretskortet til både HMI og omformerer beskrevet. I dette kapitlet vil vi ta for oss hvordan vi utviklet kretskortene i praksis, både ved design i KiCad og ved lodding av fysisk kretskort. Kapitlet er delt i tre deler. Den første delen tar for seg utviklingen av PCB for HMI-kortet. Den andre delen av kapitlet tar for seg PCB design av hovedkortet. Med hovedkortet mener vi kretskortet til selve omformerer. Den tredje og siste delen tar for seg lodding av de fysiske kretskortene og viser hvordan resultatet av kretskortutviklingen endte opp med å bli.

### 6.1 Utvikling av PCB for HMI-kort

Før vi satt i gang med design i KiCad startet vi først med å koble opp kretsen til HMI-kortet fysisk på et breadboard. Deretter testet vi kretsen med en enkel kode for å sjekke at alt fungerte som det skulle, slik at vi kunne forsikre oss om at alle koblingene var gjort riktig. Når vi hadde bekreftet at kretsen fungerte som forventet, var neste steg å tegne koblingsskjema i KiCad. Dette var ganske 'rett fram'. Når kretsskjema var ferdig tegnet var neste steg å designe selve kretskortet i PCB Editor. Første steget var å legge til riktige fysiske komponenter til de komponentene vi hadde i koblingsskjema. Veileder tipset oss om at det var lurt å bruke samme dimensjon for alle motstandene og kondensatorene, så vi ble enige om at vi skulle gjøre det på denne måten. I/O expanderen var en standard chip av typen SOP, så for denne chippen brukte vi et standard SOP-footprint som lå inne i KiCad. Trykknappene, encoderen og RJ45-kontakten var alle komponenter som lå inne i databasen til KiCad med tilhørende footprint, noe som gjorde det enkelt for oss.

Etter at alle komponentene i kretsskjema hadde blitt tildelt et footprint, manglet vi bare å trekke kobberbaner mellom komponentene. Vi innså raskt at dette kunne bli krevende dersom en ikke planlagte komponentplasseringene på forhånd. Vi skjønnte fort at det var lurt planlegge hvor komponentene skulle plasseres på kortet i forhold til hverandre for å få kortest mulig kobberbaner mellom komponentene samt for å minimere antall kryssningspunkter for kobberbanene. Det var et ønske fra veilerer at vi skulle forsøke å bruke minst mulig via, noe som gjorde det mer utfordrende å legge banene. Etter å ha prøvd seg litt fram begynte utlegget etter hvert å ta form, og utlegget endte med å bli som



## 6.2 Utvikling av PCB for hovedkort

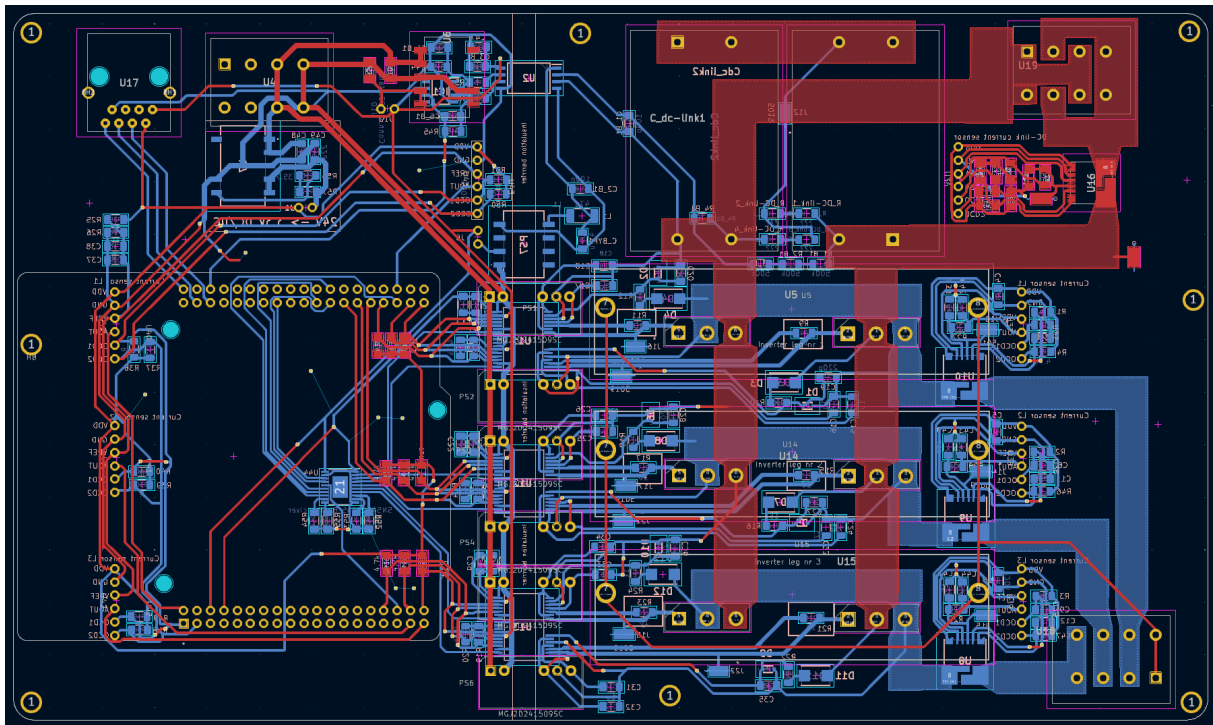
Som nevnt tidligere hadde vi som utgangspunkt å benytte samme komponenter, og dermed samme koblinger, som den tidligere gruppen. Som følge av dette kunne vi benytte koblingsskjemaene som ble utviklet våren 2023 av den tidligere gruppen. Dette ville spare oss for mye tid og gjøre at vi kunne rette all tid og fokus mot å optimalisere PCB-layouten.

Når vi skulle sette i gang med å designe kretskortet i PCB Editor i KiCad, var første steget å plassere komponentene. Vi tok da utgangspunkt i hvordan komponentene var plassert på prototypen vist i figur 5.4. Når komponentene var riktig plassert var neste steg å begynne å rute baner, det vil si trekke kobberbaner mellom komponentene. Vi startet med å ta for oss kraftsiden av kretskortet, og de første banene som ble rutet var sterkstrømsbanene (DC-link og fasene). For å unngå å måtte bruke via på sterkstrømsbanene innså vi fort at den enkleste løsningen var å ha DC-link banene på det øverste kobberlaget og fasene på det nederste. Denne løsningen gjorde det også mulig å trekke baner mellom gate-driverene og gate på transistorene uten å bruke via. Når sterkstrømsbanene samt banene mellom gatedriver og gate på transistorene var ferdig, tok vi for oss den resterende rutingen som manglet på kraftsiden.

En problemstilling som vi fort støtte på var hvordan en skulle forveksle signaler mellom MCU på signalsiden av kretskortet og strømsensorene på kraftsiden. Strømsensorene er utrustet med internt galvanisk skille, så problemet kunne i teorien løses ved å plassere strømsensorene på isolasjonsbarrieren, på samme måte som gate-driverene. Dette viste seg å være veldig vanskelig å få til i praksis, og derfor besluttet vi sammen med veileder at strømsensorene skulle plasseres på kraftsiden og kobles opp mot MCU via ekstern kabelforbindelse. Dette gjorde vi ved å koble alle de 6 signalpinnene på hver av chippene til hver sin pinheader, for å så lage en pinheader på signalsiden som tilhørte enhver av pin headerene på kraftsiden. Denne løsningen fikset også et annet problem; nemlig kraftforsyning til strømsensorene. Sensorene er avhengig av en spenningsforsyning på 3.3V, og via den eksterne kabelforbindelsen kunne vi forsyne sensorene fra 3.3V forsyningen på signalsiden istedenfor å implementere en ny 3.3V forsyning på kraftsiden.

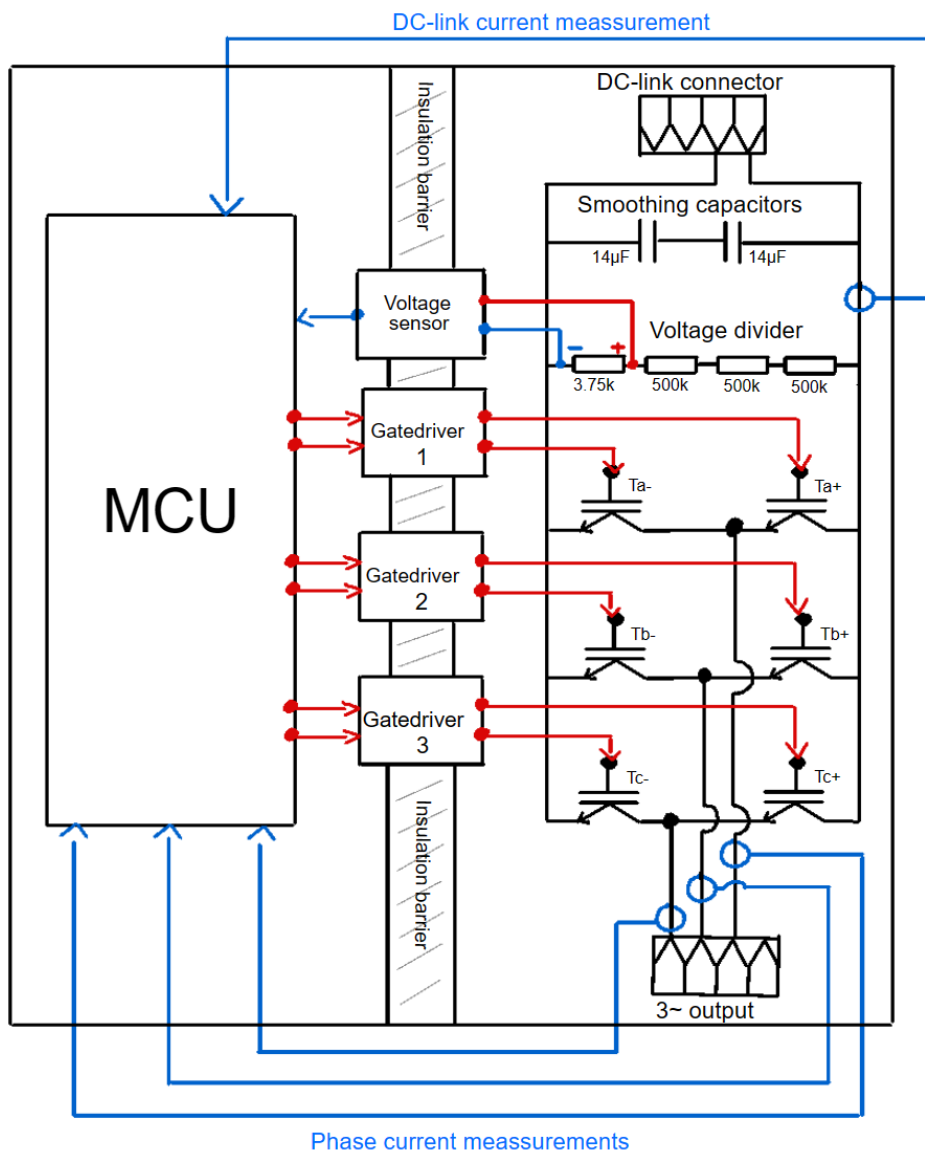
Når alle banene på kraftsiden var lagt, tok vi for oss signalsiden av kortet. Denne siden består stort sett av microcontrolleren og dens tilkoblinger. Som forklart i kapittel 5.2 er styresignalene til transistorene høyest prioritert av alle banene på kretskortet. Med dette som utgangspunkt begynte vi med å plassere MCU på en slik måte at styresignalene kunne nå fram til gatedriverne uten at disse bannene krysset hverandre slik at en kunne risikere å måtte bruke via. Det var også viktig at disse banene var mest mulig identisk i lengden, så dette ble også tatt hensyn til under plassering av MCU. En helt essensiell ting når det kommer til styresignalene er at de er tilkoblet pins på MCU som støtter en avansert timer som kan benyttes til å generere SPWM og SVPWM (mer om dette senere). Det er et fåtall av de totalt 78 pinnene på MCU som støtter dette, noe som gjør at vi ikke har mulighet til å velge fritt hvilke pin på MCU som styresignalene skal tilkobles basert på hva som er den mest lukrative plasseringen på kortet. Etter å ha testet litt forskjellige løsninger kom vi fram til en god løsning hvor banene mellom MCU og gate-driverer ble tilnærmet like lang, uten noen som helst via. Når disse banene var lagt begynte vi å rute resten av signalsiden. Figur 6.3 viser hvordan PCB-layouten endte opp med å bli til slutt.





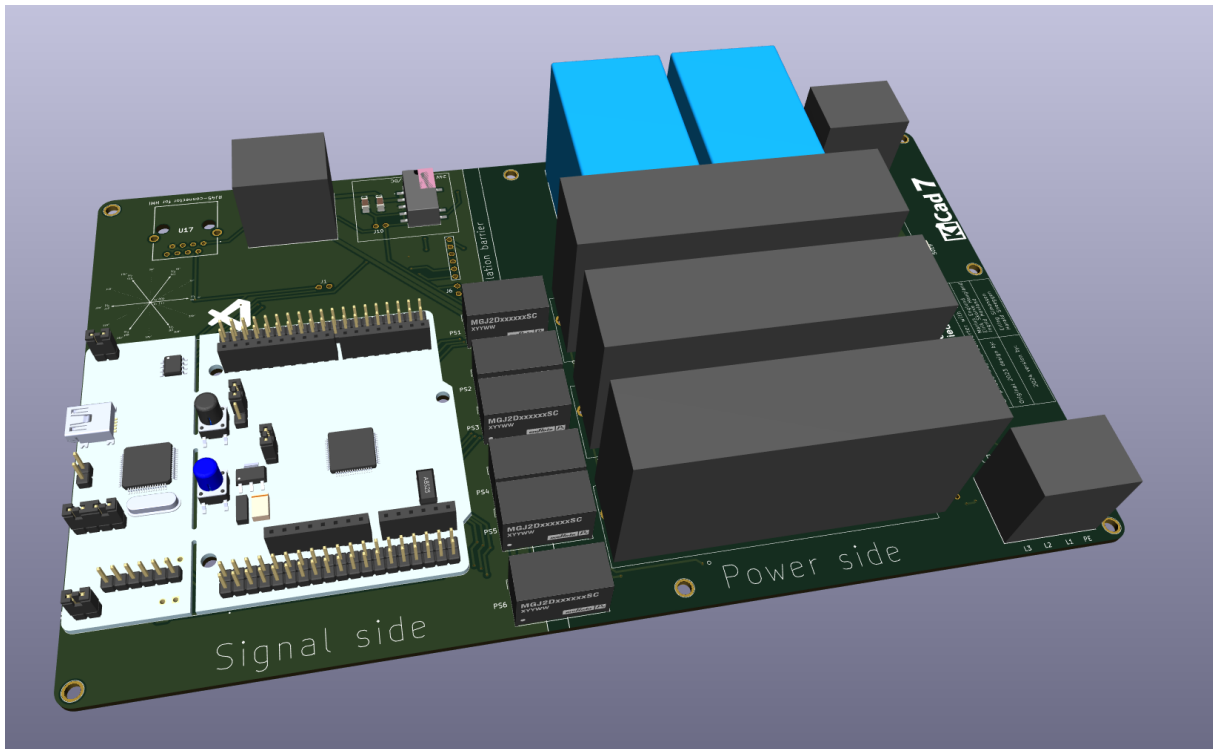
Figur 6.3: Siste utkast av PCB-layout for hovedkort  
*Kilde: skjermbilde fra KiCad*

Figur 6.4 viser et forenklet blokkdiagram som forsøker å illustrere topologien til kretskortet. Venste siden av isolasjonsbarrieren med MCU er signalsiden av kortet, mens høyresiden av isolasjonsbarrieren er kraftsiden med selve vekselretteren på. I figur 6.4 er komponentene plassert på samme lokasjon på kretskortet som de er i realiteten. I praksis er det flere I/O til MCU enn hva som er vist i figur 6.4. For strømsensorene er det to digitale statussignaler som sjekker for overstrøm. Det vil si at, siden vi har 4 stk strømsensorer, vil det totalt være 8 digitale overstrømsignaler som må inn på MCU. Dette er veldig mange innganger. For å unngå å ha så mange innganger fra strømsensorene besluttet vi å sammenkoble alle pinnene med samme overstrømsignal (ODC1 og OCD2) på de 4 sensorene, slik at vi totalt fikk 2 innganger til MCU i stedet for 8 om statusovervåking av overstrøm. Fordelen med dette er at vi reduserer antall I/O på MCU, men ulempen er at vi ikke kan vite akkurat hvilken av strømsensorene som detekterer overstrøm (DC-link eller hvilke av fasene).

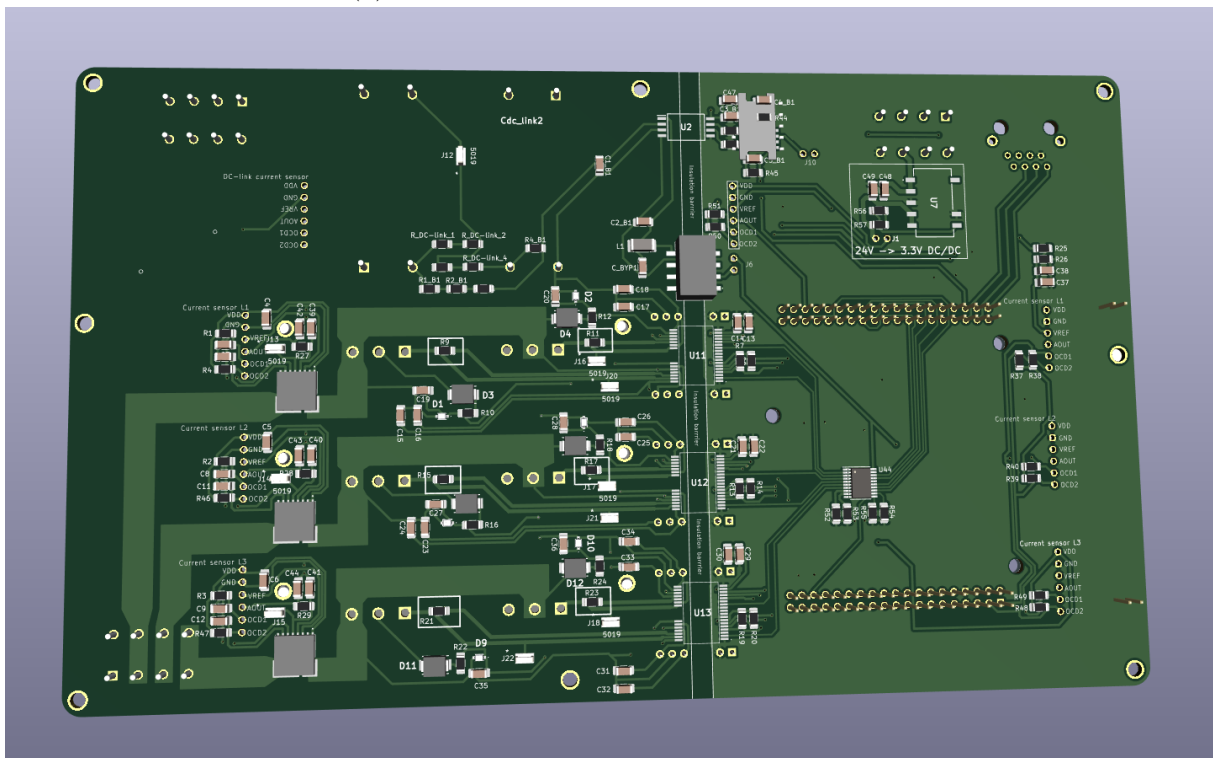


Figur 6.4: Prinsippskisse for PCB layout  
 Kilde: tegnet med ReMarkable

Når rutingen av kretskortet var ferdig var siste steg i prosessen å lage en 3D-modell av kretskortet. Dette er viktig for å dobbeltsjekke at ingen av komponentene overlapper hverandre, i tillegg til at det er kjekt å få et bilde av hvordan sluttproduktet vil se ut. 3D-modell av de forskjellige komponentene lastet vi ned fra diverse nettsider, men det var ikke alle komponentene som vi klarte å finne tilhørende 3D-modell for. Disse komponentene er vist som svarte blokker i modellen, hvor målene på blokken tilsvarer de fysiske målene på komponenten. Figur 6.5 viser hvordan 3D-modellen endte med å se ut.



(a) 3D-modell av oppsiden til kretskortet



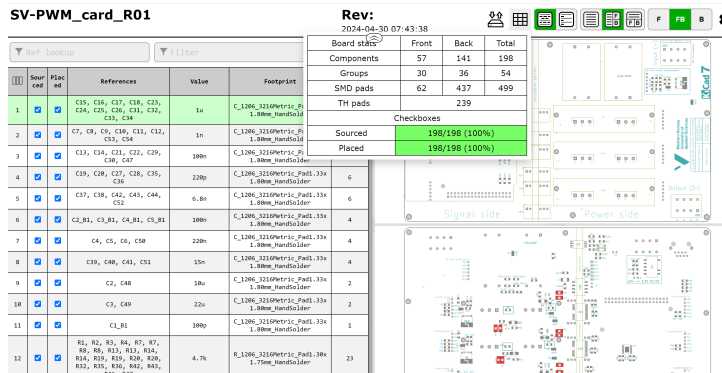
(b) 3D-modell av baksiden til kretskortet

Figur 6.5: 3D-modell av hovedkort

*Kilde: skjermbilde fra KiCad*

## 6.3 Lodding av omformer

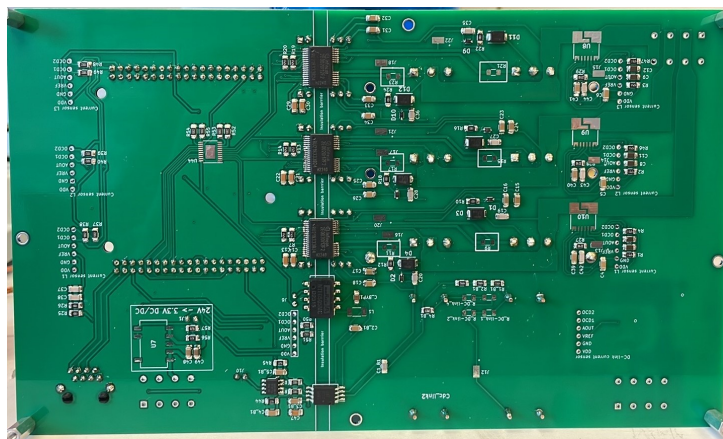
Før vi startet med å lodde omformeren, brukte vi en del tid på å lære oss hvordan man skal lodde på chippene. Dette var greit å øve på da man fort kan ødelegge tilkoblingene på kortet. For å holde oversikt over hvilke komponenter som skulle hvor, brukte vi KiCad sin BOM generator (Bill of Materials) som vist i 6.6 BOM viser en liste over alle komponentene som skal loddess på kortet, i tillegg til at det viser plasseringen til den enkelte komponent med verdi. Den holder også oversikt over hvor langt man har kommet i prosessen.



Serial	Placed	References	Value	Footprint
1	<input checked="" type="checkbox"/>	C39, C36, C17, C18, C23, C24, C25, C26, C31, C32, C33, C34	1u	C_1206_3216Metric_P1.150mm_Handholder
2	<input checked="" type="checkbox"/>	C7, C8, C9, C10, C11, C12, C13, C14	1n	C_1206_3216Metric_P1.150mm_Handholder
3	<input checked="" type="checkbox"/>	C13, C14, C11, C22, C25, C26, C27	100n	C_1206_3216Metric_P1.150mm_Handholder
4	<input checked="" type="checkbox"/>	C39, C28, C27, C18, C15, C16	220p	C_1206_3216Metric_Pad1.33x1.50mm_Handholder
5	<input checked="" type="checkbox"/>	C37, C38, C42, C43, C44, C45	6.8n	C_1206_3216Metric_Pad1.33x1.50mm_Handholder
6	<input checked="" type="checkbox"/>	C2, R1, C1, R1, C4, R1, C5, R1	100n	C_1206_3216Metric_Pad1.33x1.50mm_Handholder
7	<input checked="" type="checkbox"/>	C4, C5, C6, C8	220n	C_1206_3216Metric_Pad1.33x1.50mm_Handholder
8	<input checked="" type="checkbox"/>	C39, C40, C41, C51	15n	C_1206_3216Metric_Pad1.33x1.50mm_Handholder
9	<input checked="" type="checkbox"/>	C2, C48	10u	C_1206_3216Metric_Pad1.33x1.50mm_Handholder
10	<input checked="" type="checkbox"/>	C3, C49	22u	C_1206_3216Metric_Pad1.33x1.50mm_Handholder
11	<input checked="" type="checkbox"/>	C1, R1	100p	C_1206_3216Metric_Pad1.33x1.50mm_Handholder
12	<input checked="" type="checkbox"/>	R1, R2, R3, R4, R7, R7, R8, R9, R10, R11, R44, R14, R19, R19, R20, R20, R21, R20, R20, R42, R42, R45, R45, R47	4.7k	R_1206_3216Metric_Pad1.33x1.75mm_Handholder

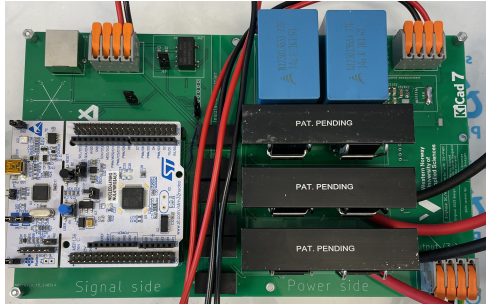
Figur 6.6: Generert BOM via KiCad  
Kilde: skjermbilde fra KiCad BOM generator

De største utfordringene med loddingen var å lodde på gate-driverne. Dette er en overflatemontert komponent der pinnene er utrolig tett plassert i forhold til hverandre. Derfor er det veldig lett å kortslutte flere pins hvis man ikke er forsiktig. Metoden vi brukte her var at vi tok på et flussmiddel på overflatene som skulle loddess. Flussmiddel rengjør og gjør at metallene som skal loddess enklere binder seg til overflatene. Under ser man hvordan undersiden på omformeren så ut etter lodding 6.7.

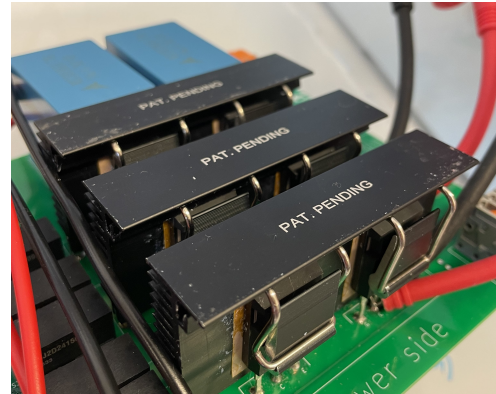


Figur 6.7: Bilde av omformerens bakside etter lodding  
Kilde: bilde tatt med iPhone

Figur 6.8 illustrerer hvordan oppsiden av omformereren så ut etter lodding.



(a) Bilde av omformer sett ovenfra



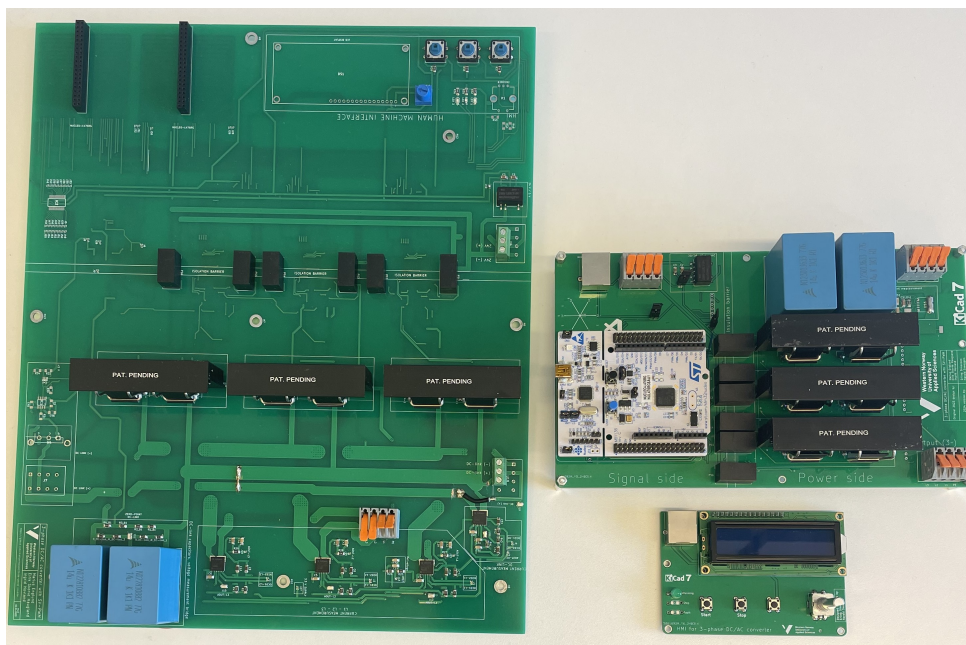
(b) Bilde av transistorer og kjøleribber

Figur 6.8: Bilder av omformer sett ovenfra

*Kilde: bilde tatt med iPhone*

## 6.4 Sammenligning av ny og gammel versjon

Figur 6.9 viser en sammenlikning av omformereren som ble utviklet våren 2023 [2] (til venstre i figur 6.9) og omformereren som vi utviklet i våren 2024 (til høyre i figur 6.9). Figur 6.9 illustrerer en betydelig arealeffektivisering i 2024-designet, takket være nøye planlegging. Vi er veldig fornøyde med resultatet av utseende og designet på omformereren.



Figur 6.9: 2023 utgave til venstre og 2024 utgave til høyre.

*Kilde: bilde tatt med iPhone*



# Kapittel 7

## Software

Dette kapittelet tar for seg filosofien bak software som ble utviklet for styring av omformer-  
ren. Dette innebærer å implementere både SVPWM og SPWM algoritmene for trefasedrift  
av omformereren, i tillegg til utvikling av styringsalgoritmer for unipolar og bipolar styring  
av H-bro både fullbro DC/DC omformer samt enfase vekselretter. Kapittelet er delt i tre  
seksjoner. Den første seksjonen, seksjon 7.1, tar for seg sentrale konsepter ved STM32-  
programmering. Den andre seksjonen, 7.2, tar for seg virkemåten av software som ble  
utviklet for trefasedrift av omformereren, mens den tredje seksjon 7.3 tar for seg software  
som ble utviklet for styring av H-bro.

### 7.1 Viktige konsepter

#### 7.1.1 Timere

For å få til å sammenligne en triangelkurve med et eller flere kontrollsignaler på en effektiv  
måte benyttes interne timere i microcontrolleren. *Nucleo L47RG* har samtlige timere  
innebygget, men det er kun et fåtall av disse som har funksjonalitetene som kreves for  
PWM-algoritmene som vi ønsker å implementere. Det kreves at timeren kan telle oppover,  
nå sin maksimale telleverdi satt av auto-reload parameteren  $k_{reload}$ , og så telle nedover  
til 0. Timeren vil da i praksis imitere triangelkurven som brukes til å sammenligne med  
kontrollsignalene. Telleverdien til timeren er en 16-bit heltallsverdi som ligger i intervallet  
[0, 65536], det samme gjelder naturligvis for auto-reload verdien  $k_{reload}$ .

Hastigheten som timeren teller med er gitt av tick-frekvensen. Denne frekvensen bestem-  
mer i praksis hvor lang tid timeren bruker på å inkrementere. Maksimal tick-frekvens  
for *Nucleo L47RG* er 80MHz. Timerens tick-frekvens kan skaleres ut fra den maksimale  
tick-frekvensen ved hjelp av en *prescaler*:

$$f_{tick} = \frac{80MHz}{Prescaler} \quad (7.1)$$

### 7.1.2 Output-compare (OC)

For å kunne sammenligne den momentane telleverdien til timeren med et (eller flere) kontrollsignal, er timeren nødt til å støtte output-compare funksjonalitet. Output compare gjør det mulig å sammenligne den momentane telleverdien til timeren med én eller flere eksterne verdier i intervallet  $[0, 65536]$  som vi her kaller  $k_{compare}$ . Det vil være mulig å initialisere timeren slik at når telleverdien er mindre enn kontrollsignalet den sammenlignes med, så settes utgangene på en bestemt kanal høy. Dette er essensielt for å kunne kjøre PWM-algoritmene som vi har som mål å implementere i software, da alle disse algoritmene bygger på prinsippet om å sammenligne en triangelkurve med et eller flere kontrollsignal.

### 7.1.3 Dødtid

Kanalene CH1, CH2 og CH3 på Nucleo L47RG har en tilhørende kanal CH1N, CH2N og CH3N som er invers av den tilhørende kanalen. Dette er veldig nyttig når en skal benytte microcontrolleren til styring av sivtsjede omformere som for eksempel en H-bro. Ved hjelp av denne funksjonaliteten kan en sørge for at begge transistorene på samme bein i H-broen aldri er på samtidig. Dette oppnås ved å sette CH1 til å styre HS transistoren og CH2N til å styre LS transistoren.

STM32 sitt LL-bibliotek gjør det mulig å legge inn en dødtid mellom når CHx skal endre tilstand. Under dødtiden er begge signalene, både CHx og CHxN lav, for en av de går høy. Dette skaper en ekstra sikringsmargin ved at de to kanalene aldri er på samtidig.

### 7.1.4 Ulike PWM-moduser ved output-compare

Ved innstilling av output-compare på en timer i STM32 har en valget om å velge mellom to forskjellige PWM-moduser; PWM1 og PWM2. De to modusene er invertert i forhold til hverandre.

**PWM1** gjør at utgangene på den tilhørende kanalen settes høy så lenge compare-verdien er større enn telleverdien til timeren. Når telleverdien til timeren er større en compare-verdien settes kanalen lav.

**PWM2** er motsatt av PWM1; når compare-verdien er større enn telleverdien til timeren settes kanalen lav, mens når compare-verdien er mindre enn telleverdien til timeren så settes kanalen høy.

## 7.2 Software for trefasedrift

### 7.2.1 Konstant $\frac{U}{f}$ -kontroll for trefasedrift

Som overordnet styringsprinsipp for trefasedrift, uavhengig av modulasjonsalgoritme, har vi valgt å ta utgangspunkt i prinsippet om å holde forholdet mellom påtrykt spenning og frekvens konstant. Dette prinsippet er beskrevet i 1.2. Gjennom HMI mottar MCU ønsket modulasjonsfrekvens fra encoderen. Basert på modulasjonsfrekvens beregnes nødvendig amplitudemodulasjonsforhold  $m_a$  for at avgitt spenning fra omformer skal justeres etter samme forhold som frekvensen.

I kapittel 4 forklarte vi at grunnharmonisk linjespenning ved SPWM/SVPWM er gitt ved:

$$V_{LL1} = \frac{\sqrt{3}}{2\sqrt{2}} \cdot m_a \cdot V_d \quad (7.2)$$

Dersom vi skal holde  $\frac{U}{f} = konst.$  så får vi:

$$\frac{V_{LLnominell}}{f_{nominell}} = \frac{V_{LLny}}{f_{ny}} \rightarrow V_{LLny} = V_{LLnominell} \cdot \frac{f_{ny}}{f_{nominell}} \quad (7.3)$$

Setter vi inn uttrykket for  $V_{LL1}$  fra ligning 7.2 inn for  $V_{LLny}$  i ligning 7.3 får vi:

$$\frac{\sqrt{3}}{2\sqrt{2}} \cdot m_a \cdot V_d = V_{LLnominell} \cdot \frac{f_{ny}}{f_{nominell}} \quad (7.4)$$

Og hvis vi løser ligning 7.4 for  $m_a$  får vi:

$$m_a = f_{ny} \cdot \frac{V_{LLnominell}}{f_{nominell}} \cdot \frac{2\sqrt{2}}{\sqrt{3} \cdot V_d} \quad (7.5)$$

Vi ser da at uttrykk 7.5 gir oss mulighet til å beregne hva amplitudemodulasjonsforholdet må være gitt ønsket frekvens. Uttrykket viser også at påtrykt DC-link spenning har stor betydning for amplitudemodulasjonsforholdet. Ved å overvåke DC-link spenningen via en spenningssensor kan en oppnå en lukket kontrollsløyfe.

### 7.2.2 Regulering av modulasjonsfrekvens

Både SPWM og SVPWM innebærer bruk av sinusfunksjoner. For å imitere sinuskurvene benyttes en array som inneholder sinus-verdier for én full periode. Totalt inneholder tabellen som ble brukt av den tidligere bachelorgruppen 101 tallverdier [2]. For å dynamisk oppdatere sinusverdiene når de sammenlignes med TIM1 på output-compare modus, kan compare-funksjonene programmeres inn i en ISR-funksjon for et interrupt som trigges på bestemt tid. Inne i ISR-funksjonen initialiseres tre tellere som er 120 grader tidsforskjøvet fra hverandre. Ved hjelp av de tre tellerene kan en generere tre sinuskurver som er 120 grader tidsforskjøvet fra hverandre. Telleren må da ligge i intervallet [1,101] for å kunne gjennomgå hele sinustabellen. Når telleren har fullført én komplett gjennomgang av sinustabellen har én sinus-periode gått. Det er dermed tydelig at ved å styre hastigheten til telleren så kan en i praksis styre frekvensen til sinuskurvene. Den interne timeren som



ble valgt til å kjøre output-compare på var TIM1, og timer 2 (TIM2) ble benyttet for timer-interrupt [2].

En kan instille et interrupt slik at interruptet trigges hver gang timeren når sin maksimale telleverdi. Da kjøres koden som ligger i ISR-funksjonen én gang, før timeren resetter telleverdien sin til 0 og begynner å telle oppover på nytt. Som forklart i 7.1.1 er det mulig å endre hastigheten som timeren teller med,  $f_{tick}$ . For enkelhetens skyld valgte vi å sette  $f_{tick}$  til å konstant være lik 10MHz, som i praksis tilsier at timeren inkrementerer hvert 100ns. Autoreload-parameteren til timeren,  $k_{reload}$ , er dermed avgjørende for hvor ofte interruptet trigges. Trigger-frekvensen til interruptet kan dermed beregnes ut fra formelen:

$$T_{interrupt} = T_{tick} \cdot k_{reload} \quad (7.6)$$

Vi har valgt å sette interrupt-frekvensen til 10µs, som medfører fra ligning 7.6 at autoreload parameteren til timeren (TIM2),  $k_{reload}$ , må settes til 100. Videre definerer vi en interrupt-teller som teller hvor ofte interruptet trigges. Gitt at vi ønsker en modulasjonsfrekvens  $f_{modulation}$  så vet vi at de tre tellerene som gjennomgår sinustabellen må inkrementeres med følgende inkrementeringsfrekvens:

$$f_{increment} = 101 \cdot f_{modulation} \quad (7.7)$$

Vi kan oppnå denne inkrementeringsfrekvensen med en enkel if-setning som sier at dersom interrupt-telleren er større enn en gitt limit gitt av:

$$lim := \frac{f_{increment}}{f_{interrupt}} \quad (7.8)$$

så øker de tre tellerene med én, og interrupt-telleren resettes til 0. På denne måten kan en styre modulasjonsfrekvensen ved å beregne  $lim$  som funksjon av ønsket modulasjonsfrekvens  $f_{modulation}$ .

### 7.2.3 Modulasjonsalgoritmer for trefasedrift

Vi vil her legge fram algoritmene som har blitt benyttet for å generere trefase SPWM og SVPWM. Her har vi hentet inspirasjon fra den tidligere gruppens arbeid og basert oss på gjenbruk av deres kode.

#### Trefase SPWM

Løsningen som ble implementert våren 2023 var å oppdatere compare-verdiene til timer 1 med uttrykk som vist under [2, s. 96].

$$\begin{aligned} U_{CH1_{compare}} &:= \frac{1}{2} \cdot (m_a \cdot k_{reload} \cdot sineTable[i1] + k_{reload}) \\ U_{CH2_{compare}} &:= \frac{1}{2} \cdot (m_a \cdot k_{reload} \cdot sineTable[i2] + k_{reload}) \\ U_{CH3_{compare}} &:= \frac{1}{2} \cdot (m_a \cdot k_{reload} \cdot sineTable[i3] + k_{reload}) \end{aligned}$$

Her er  $i1$ ,  $i2$  og  $i3$  de tre tellerene som er 120 grader tidsforskjøvet og som inkrementeres med inkrementeringsfrekvensen gitt av ligning 7.7.

## SVPWM

Til motsetning fra SPWM, legges det til en tredjeharmonisk referansekomponent til hver av de tre sinuskurvene som sammenlignes med triangelkurven ved SVPWM [2]. Ved implementering av SVPWM i software har vi gjenbrukt kode som ble utviklet våren 2023 [2, s. 99]. I koden oppdateres OC-verdiene til TIM1 etter følgende uttrykk:

$$U_{CH1_{compare}} := k_{reload} \cdot \left( \frac{1}{\sqrt{3}} \cdot m_a \cdot sineTable[i1] - \frac{U_k}{U_d} + \frac{1}{2} \right) \quad (7.9)$$

$$U_{CH2_{compare}} := k_{reload} \cdot \left( \frac{1}{\sqrt{3}} \cdot m_a \cdot sineTable[i2] - \frac{U_k}{U_d} + \frac{1}{2} \right) \quad (7.10)$$

$$U_{CH3_{compare}} := k_{reload} \cdot \left( \frac{1}{\sqrt{3}} \cdot m_a \cdot sineTable[i3] - \frac{U_k}{U_d} + \frac{1}{2} \right) \quad (7.11)$$

hvor:

$$U_k = \frac{1}{2} \cdot (max(U_{Ao1}, U_{Bo1}, U_{Co1}) + min(U_{Ao1}, U_{Bo1}, U_{Co1})) \quad (7.12)$$

og

$$U_{Xo1} = m_a \cdot \frac{U_d}{\sqrt{3}} \cdot sin(2\pi \cdot f_{modulation} \cdot t + \theta_s) \quad (7.13)$$

## 7.3 Software for drift av H-bro

Dette delkapittelet tar for seg software som ble utviklet for å kunne drive omformereren som H-bro. De to transistorparene i vekselretteren som benyttes til å konfigurere omformereren som H-bro er par B og par C (L2 og L3), som styres av CH2 og CH3.

### 7.3.1 H-bro som DC/DC omformer

Denne delen av software har til hensikt å implementere styringsmetodene som ble beskrevet i kapittel 3. Det er ønskelig å kunne bruke H-broen til å drive en DC-maskin, og for dette formålet kan en bruke de to styringsmetodene som nå vil bli beskrevet.

#### Unipolar styring ( $DC \rightarrow DC$ )

For å implementere unipolar styring i software har vi tatt utgangspunkt i teorien som er gjennomgått i 3.2. I denne algoritmen styres transistorene individuelt. Vi kan dermed benytte to ulike kontrollsignaler; et til CH2 (bein 1 i H-bro) og et til CH3 (bein 2 i H-bro). Både CH2 og CH3 settes til å drifte i PWM1 moduset (ref 7.1.4), men de to kanalene får en egen compare-verdi. Koden er lagt opp slik at brukeren legger inn ønsket arbeidssyklus på *avgitt* spenning  $D_{out}$ . Basert på  $D_{out}$  beregner koden hva  $D_A$  og  $D_B$  må være etter følgende uttrykk:

$$D_A = 0.5 \cdot (D_{out} + 1) \quad (7.14)$$

$$D_B = 1 - D_A \quad (7.15)$$

Compare-verdiene for CH2 og CH3 kan dermed oppdateres etter følgende uttrykk:

$$U_{CH2_{compare}} := D_A \cdot k_{reload_{TIM1}} \quad (7.16)$$

$$U_{CH3_{compare}} := D_B \cdot k_{reload_{TIM1}} \quad (7.17)$$

#### Bipolar styring ( $DC \rightarrow DC$ )

For bipolar styring benyttes konseptet fra kapittel 3.3. Ved denne styringsformen styres transistorene i par, som forklart i 3.3, og en opererer kun med et kontrollsignal som sammenlignes med triangelkurve. For å få dette til i software kan en sette samme compare-verdi til CH2 og CH3, men siden LS transistor på bein B skal lede i par med HS transistor på bein A (og motsatt), er en nødt til å invertere CH3 i forhold til CH2 slik at CH3 er høy når kontrollsignalet er *mindre* enn triangelkurven mens CH2 er høy når kontrollsignalet er *større* enn triangelkurven. Dette kan oppnås ved å sette CH2 i PWM-modus 1 (PWM1) og CH3 i PWM-modus 2 (PWM2).

Ellers har vi her også gjort det slik at bruker oppgir ønsket verdi på arbeidssyklus til avgitt spenning, og koden beregner så hva kontrollsignalet (i praksis compare-verdien til TIM1) må være for å oppnå den ønskede arbeidssyklusen. Kontrollsignalet oppdateres etter følgende formel:

$$U_{CH2_{compare}} = U_{CH3_{compare}} := D_{out} \cdot k_{reload_{TIM1}} \quad (7.18)$$

Selve koden som er utviklet i forbindelse med bipolar styring av fullbro DC/DC omformer er lagt fram i vedlegg E.4.

### 7.3.2 H-bro som enfase vekselretter

Denne delen av software har til mål å implementere enfaset SPWM i software for å gjøre omformereren i stand til å omforme likestrøm til enfaset vekselstrøm. Koden som er utviklet i forbindelse med enfase SPWM for H-bro er å finne i vedlegg E.3.

#### Bipolar styring ( $DC \rightarrow AC$ )

For bipolar SPWM styres transistorene i par på samme måte som med unipolar styring med DC/DC drift som er beskrevet i 7.3.1. Forskjellen her er at compare-verdien her er nødt til å være en sinuskurve med ønsket modulasjonsfrekvens. Vi bruker her, av samme grunn som beskrevet i kapittel 7.2.2, en interrupt-funksjon for å oppdatere compare-verdien til TIM1. Sinus-verdien settes som compare-verdi både på CH2 og CH3, men CH2 er nå i PWM1-modus mens CH3 er i PWM2-modus (på akkurat samme måte som beskrevet i 7.3.1). For å imitere sinusfunksjonen benyttes her også sinustabellen som er beskrevet tidligere. Compare-verdiene oppdateres etter følgende formel:

$$U_{CH2compare} = U_{CH3compare} := \frac{1}{2} \cdot k_{reload_{TIM1}} \cdot (m_a \cdot sineTable[i] + 1) \quad (7.19)$$

#### Unipolar styring ( $DC \rightarrow AC$ )

De to beinere i H-broen styres nå individuelt, derfor settes både CH2 og CH3 i PWM1-modus. Ved unipolar SPWM for enfasedrift må en generere 2 sinuskurver som er 180 grader tidsforskjøvet i forhold til hverandre. Den ene sinuskurven settes som compare-verdi for CH2 mens den andre for CH3. På samme måte som beskrevet tidligere blir en her nødt til å benytte et interrupt for å oppnå dynamisk oppdatering av de to compare-verdiene. Prinsippet er likt som for trefase SPWM og SVPWM, men for å oppnå 180 grader faseforskyvning mellom de to sinuskurvene må en opprette to tellere; en som starter på 0 og en som starter på 50 (midt-verdien av sinustabellen som totalt inneholder 101 verdier). La  $i_1$  være teller 1 som starter på 0 og  $i_2$  være teller 2 som starter på 50. De to compare-verdiene oppdateres dermed etter følgende uttrykk:

$$U_{CH2compare} = \frac{1}{2} \cdot k_{reload_{TIM1}} \cdot (m_a \cdot sineTable[i_1] + 1)$$
$$U_{CH3compare} = \frac{1}{2} \cdot k_{reload_{TIM1}} \cdot (m_a \cdot sineTable[i_2] + 1)$$

Del III

Testing

# Kapittel 8

## Laboratorie testing

For å validere designet av kretskortet som vi har utviklet i forbindelse med dette prosjektet, er det aktuelt å gjennomføre laboratorieeksperimenter. Testprosedyren vi nå presenterer har til hensikt å teste de ulike modulasjonsteknikkene som er implementert i software. Under lodding av kretskortet oppsto det et uhell som førte til at den ene fasen ble defekt. Dette gjorde at vi ikke fikk testet kretskortet som en trefaset vekselretter, men heller som en H-bro. Som følge av dette fikk vi derfor ikke mulighet til å teste koden som vi utviklet for trefasedrift av omformereren i praksis.

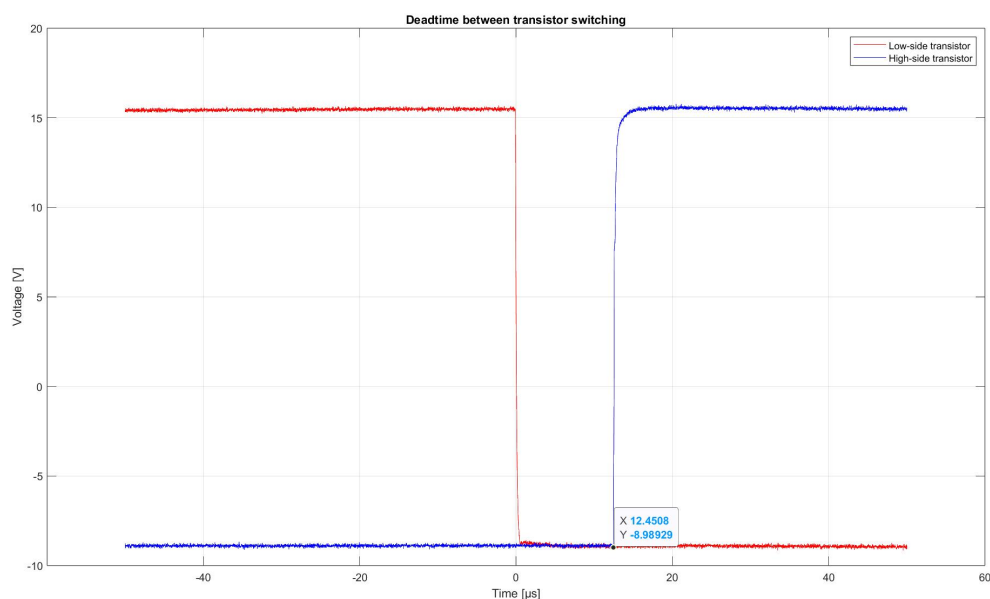
### 8.1 Testresultater

Dette delkapittelet har som hensikt å legge fram resultater fra målingene som vi gjorde ved de ulike testene vi utførte på omformereren.

## 8.1.1 Verifikasjon av pulsbredde modulasjongsenerering

### Verifikasjon av dødtid

For å forsikre at high side og low side transistorene på samme bein aldri er på samtidig, er det vanlig å legge inn en dødtid mellom hver gang transistorene endrer tilstand. Dette er beskrevet i kapittel 7. Før vi skulle teste omformerer med belastning var det viktig å teste at dødtiden ble satt slik som koden vi hadde utviklet skulle tilsi. Testen ble utført med en dødtid som vi satt til  $12,3\mu\text{s}$  i koden. Figur 8.1 viser spenningssignalene mellom gate og emitter på high side og low side transistorene.



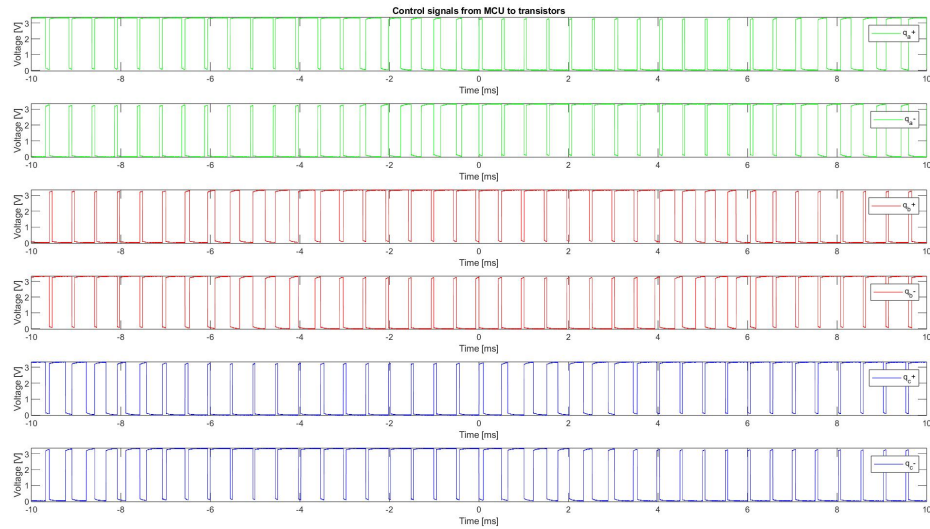
Figur 8.1: Spenning mellom gate og emitter for low side og high side transistorer

*Kilde: Plottet i MATLAB med data fra oscilloskop*

### Verifikasjon av styresignaler

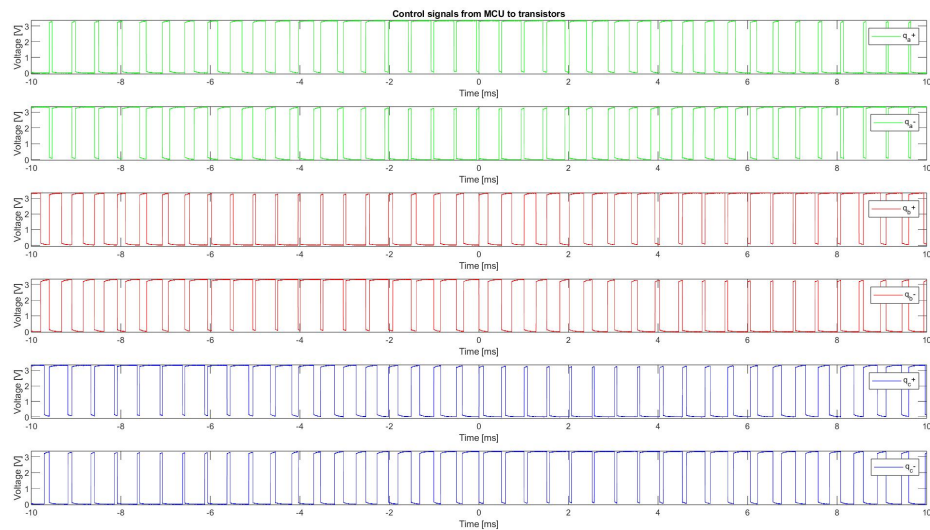
Etter at vi hadde bekreftet at dødtiden ble lagt inn, var neste steg å teste at styresignalene ved de ulike modulasjonsteknikkene ble generert på riktig måte. Vi har her benyttet et oscilloskop for å kunne visualisere styresignalene, og resultatene fra målingene er vist i figur 8.2-8.7.

## Trefase SVPWM



Figur 8.2: Styresignaler for trefase SVPWM  
( $m_a = 0.8$  ,  $f_{sw} = 1kHz$  ,  $f_1 = 50Hz$ )  
Kilde: plottet i MatLab med målinger fra Oscilloskop

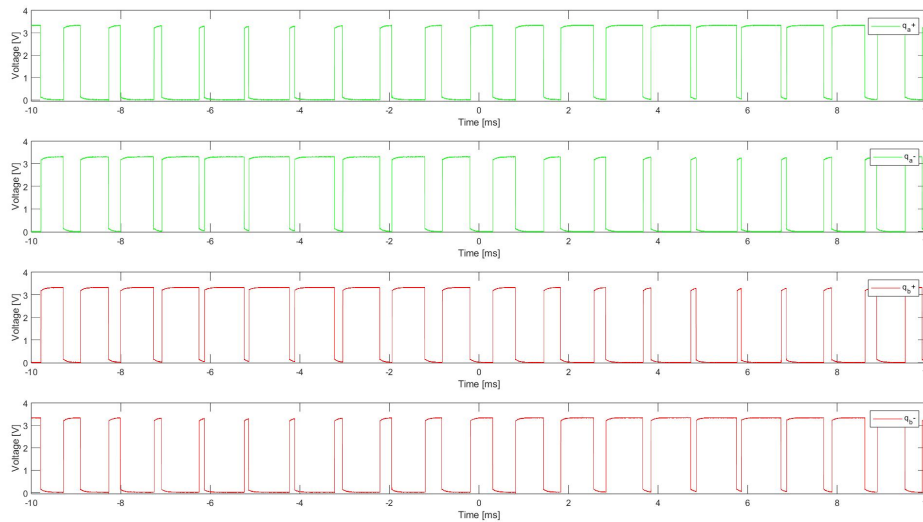
## Trefase SPWM



Figur 8.3: Styresignaler for trefase SPWM  
( $m_a = 0.8$  ,  $f_{sw} = 1kHz$  ,  $f_1 = 50Hz$ )  
Kilde: plottet i MatLab med målinger fra Oscilloskop



## Bipolar styring av enfase SPWM

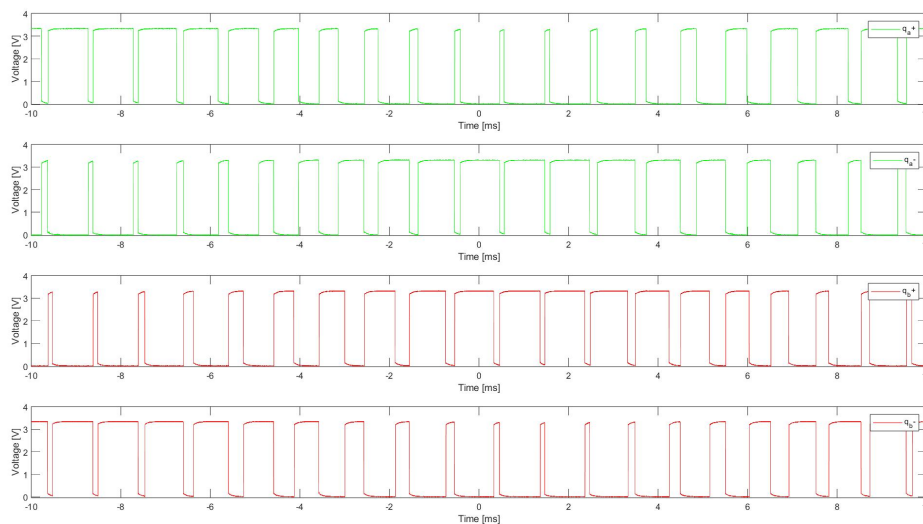


Figur 8.4: Styresignaler for bipolar kontroll av enfase SPWM

$$(m_a = 0.8, f_{sw} = 1kHz, f_1 = 50Hz)$$

Kilde: plottet i MatLab med målinger fra Oscilloskop

## Unipolar styring av enfase SPWM

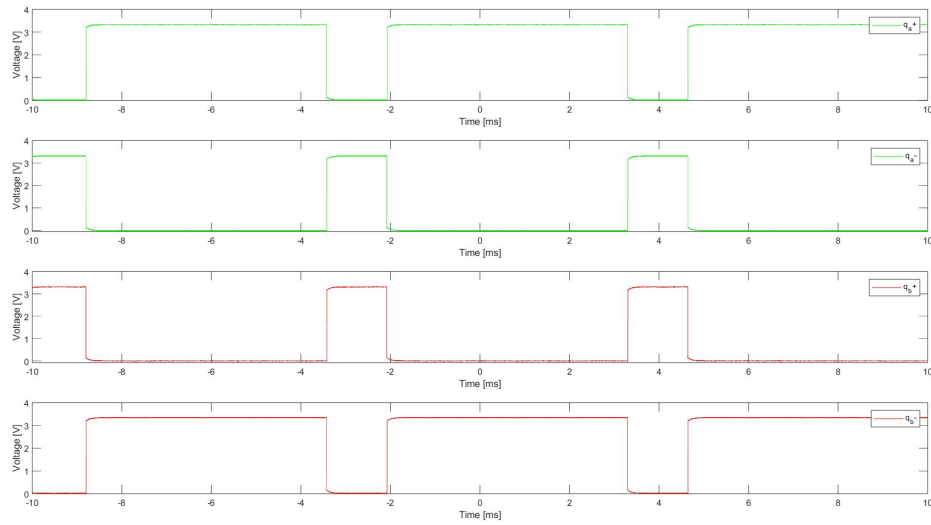


Figur 8.5: Styresignaler for unipolar kontroll av enfase SPWM

$$(m_a = 0.8, f_{sw} = 1kHz, f_1 = 50Hz)$$

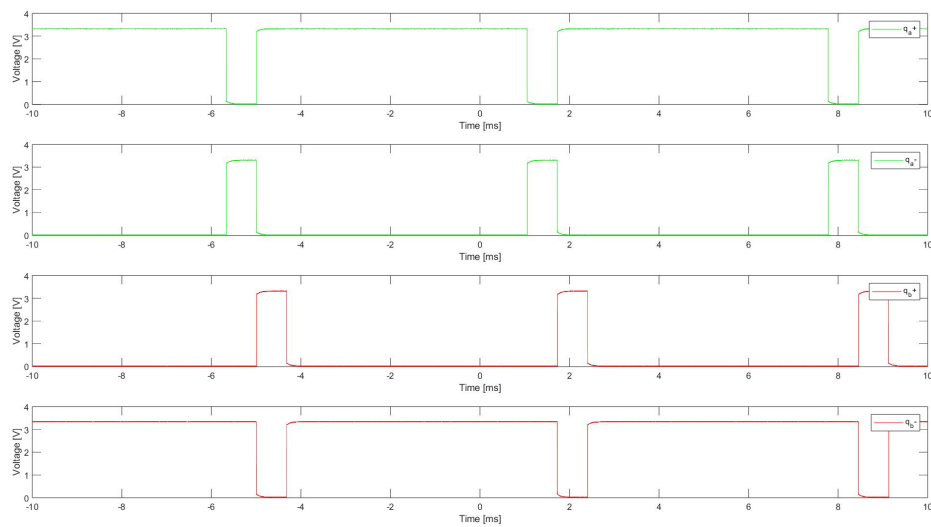
Kilde: plottet i MatLab med målinger fra Oscilloskop

## Bipolar styring av fullbro DC/DC omformer



Figur 8.6: Styresignaler for bipolar styring av fullbro DC/DC omformer  
( $D_{out} = 0.8$  ,  $f_{sw} = 150Hz$ )  
Kilde: plottet i MatLab med målinger fra Oscilloskop

## Unipolar styring av fullbro DC/DC omformer



Figur 8.7: Styresignaler for unipolar styring av fullbro DC/DC omformer  
( $D_{out} = 0.8$  ,  $f_{sw} = 150Hz$ )  
Kilde: plottet i MatLab med målinger fra Oscilloskop

## Måling av beregningshastighet for trefase SPWM

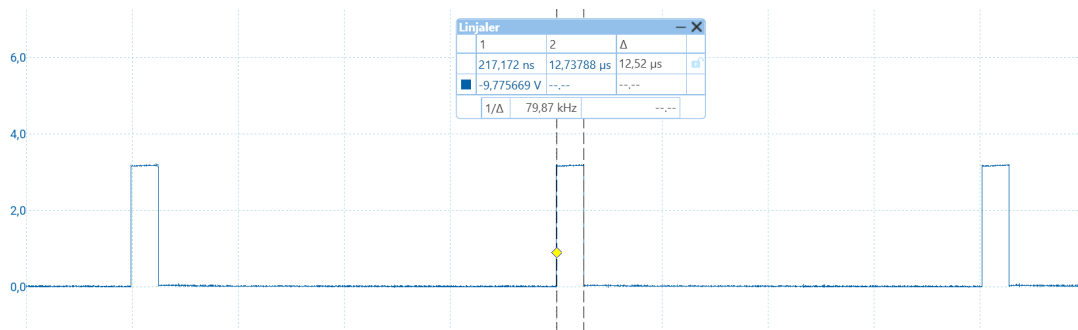
I kapittel 7 forklarte vi hvordan en kan benytte en tabell som inneholder sinus-verdier for én sinusperiode for å imitere sinuskurver. Et annet alternativ en kan benytte for å generere sinuskurver er å bruke den innebygde `sin()`-funksjonen fra `<math.h>`-biblioteket i STM32. Ved å bruke `sin()`-funksjonen kan en oppdatere compare-verdiene ved SPWM med følgende uttrykk:

$$U_{CH1_{compare}} := \frac{1}{2} \cdot (m_a \cdot k_{reload} \cdot \sin(2\pi \cdot f \cdot t_1) + k_{reload})$$

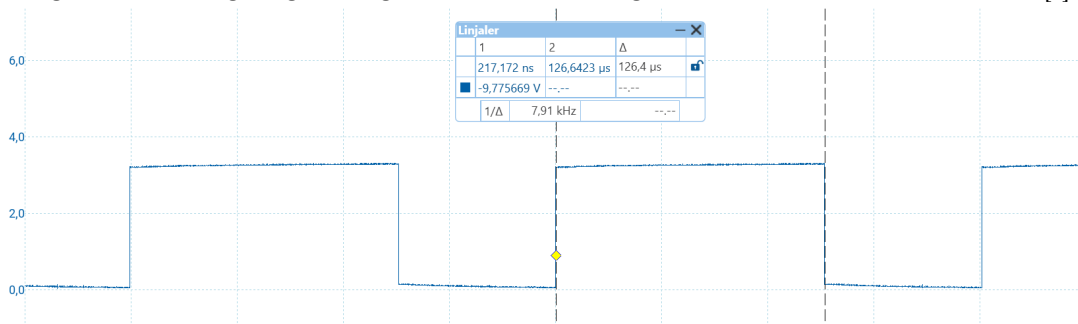
$$U_{CH2_{compare}} := \frac{1}{2} \cdot (m_a \cdot k_{reload} \cdot \sin(2\pi \cdot f \cdot t_2) + k_{reload})$$

$$U_{CH3_{compare}} := \frac{1}{2} \cdot (m_a \cdot k_{reload} \cdot \sin(2\pi \cdot f \cdot t_3) + k_{reload})$$

For å bestemme hvilken av disse to metodene som er raskest, utførte vi en hastighetstest. Denne testen baserte seg på å måle tiden som microcontrolleren bruker på å oppdatere compare-verdiene til TIM1 med sinustabell og med `sin()`-funksjonen. Vi målte tiden ved å sette en digital utgang høy i begynnelsen av ISR-funksjonen til interruptet som oppdaterer sinusverdiene, for å så sette utgangen lav igjen på slutten av ISR-funksjonen. På denne måten kan vi måle med et oscilloskop hvor lenge denne digitale utgangen er høy, som vil fortelle oss hvor lang tid det har tatt å gjennomgå koden som ligger inne i ISR-funksjonen. Figur 8.8 og 8.9 viser resultatene fra målingene.



Figur 8.8: Beregningshastighet for SPWM-algoritme ved bruk av sineTable[ ]



Figur 8.9: Beregningshastighet for SPWM-algoritme ved bruk av `sin()`-funksjonen fra `<math.h>`-biblioteket

*Kilde: skjermbilde av PicoScope 4000 Series oscilloskop*

## 8.1.2 Test av gate-motstand

Målet med denne testen er å observere hvordan ulike ohm-verdier av gate-motstanden, som er koblet i serie mellom styresignalet fra gatedriveren og gate på transistoren, påvirker svitsjingen til transistorene. Vi ønsker å teste hvordan svitsjetapene i transistorene påvirkes av ulik gate motstand. Vi ønsker også å måle temperaturen i transistoren ved ulik gate-motstand for å observere hvordan transistoren blir termisk påvirket av høyere svitsjetap.

Testene utføres på én fase, hvor omformerer påtrykkes en konstant spenning på 380VDC på DC-link. Styresignalene til transistorene er firkantpulser med 50 prosent arbeidssyklus og 4kHz svitsjefrekvens. Lasten er en konvensjons-ovn (resistiv last) som vi målte at hadde en motstand på  $70.43\Omega$ . Det vil si at med oppgitt spenning og arbeidssyklus vil lasten teoretisk sett trekke en effekt på  $512.6W$  med et strømtrekk på rundt  $2.65A$ . Denne lasten er holdt konstant under alle målingene. Alle målingene bortsett fra temperaturmålingen ble gjort med *PicoScope 4000 Series*. For å beregne gjennomsnittlig effekt omsatt i transistoren brukte vi MatLab. Måleinstrumentet for temperaturmålingene er et termisk kamera av typen *Fluke Ti27*. Alle målingene er gjort med en dødtid på  $12.3\mu s$ .

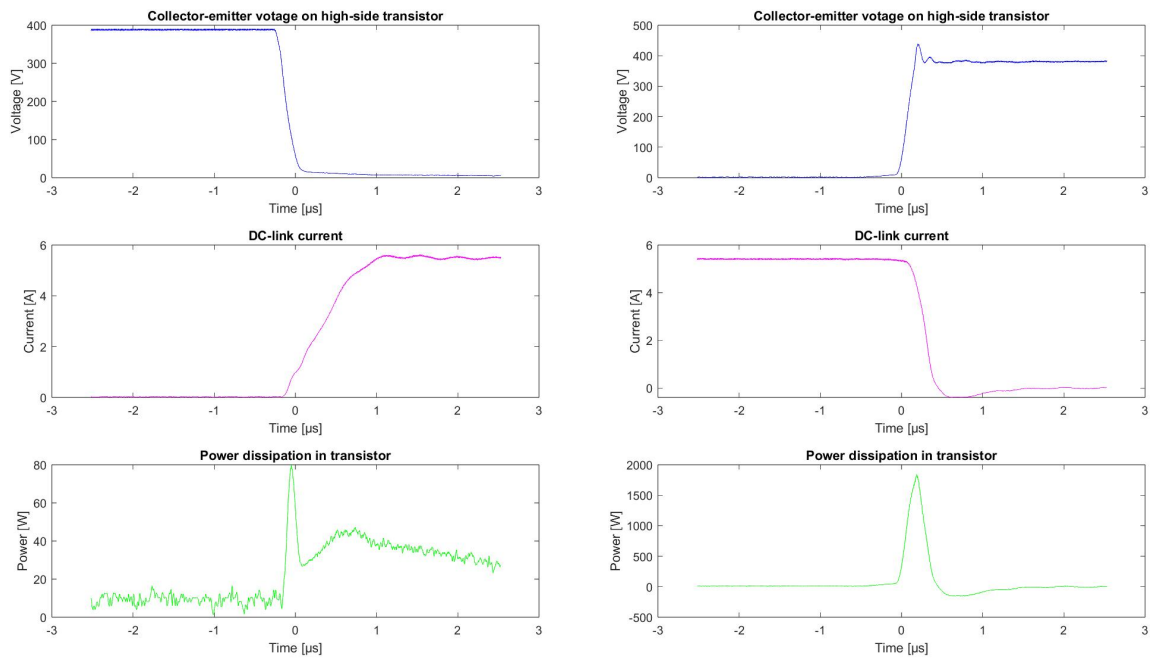
Målingene som vi tok med oscilloskop ved ulik gate motstand er vist i figur 8.10-8.13. Tabell 8.1 viser beregnet snitteffekt ved ulik gate motstand, og tabell 8.2 viser målt temperaturøkning ved de forskjellige verdiene av  $R_{gate}$ .

Tabell 8.1: Måling av effekttap i transistor ved ulik gate-motstand

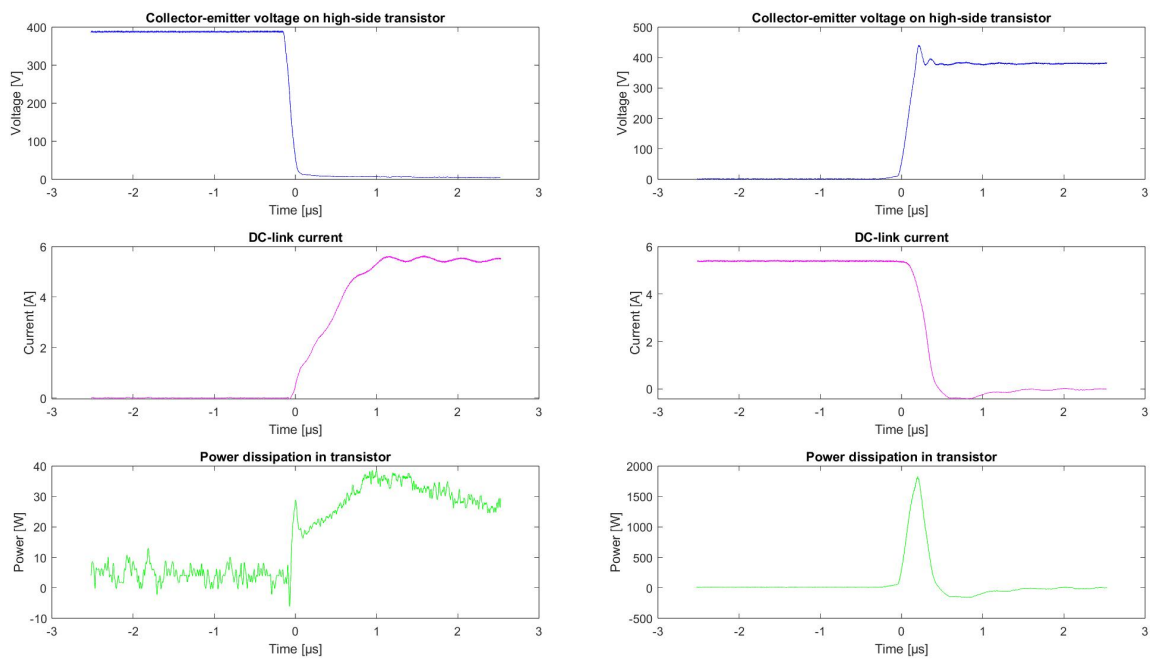
Gate-motstand	Effekttap i transistor ( $P_{SW} + P_{cond.}$ ) [W]
100 $\Omega$	14.4
49.9 $\Omega$	10.77
24.9 $\Omega$	3.99
10 $\Omega$	9.47
4.99 $\Omega$	1.8

Tabell 8.2: Termiske målinger med ulik gate-motstand men identisk last

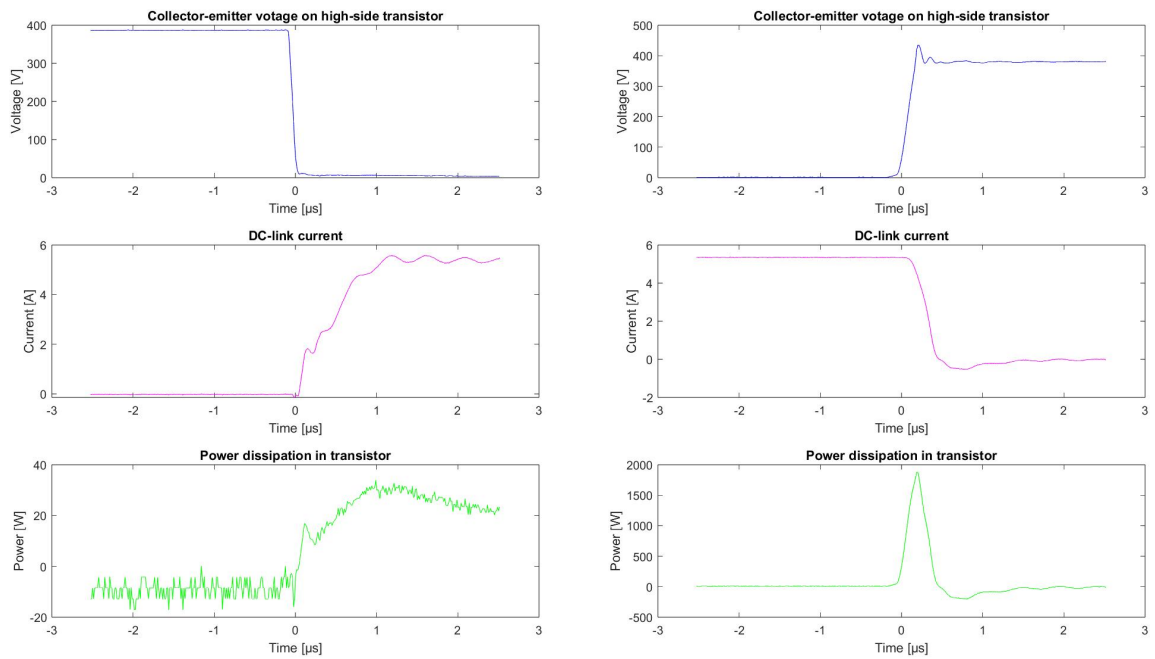
Gate-motstand	$T_{start}$ [°C]	$\Delta t$ [min]	$T_{slutt}$ [°C]	Avg. $\frac{\Delta T}{\Delta t}$ [°C/min.]
100 $\Omega$	30	7	52	3.143
49.9 $\Omega$	33.4	7	54.2	2.97
24.9 $\Omega$	33.6	7	53.0	2.77
10 $\Omega$	36.3	7	52.6	2.329
4.99 $\Omega$	40.6	7	54.7	2.014



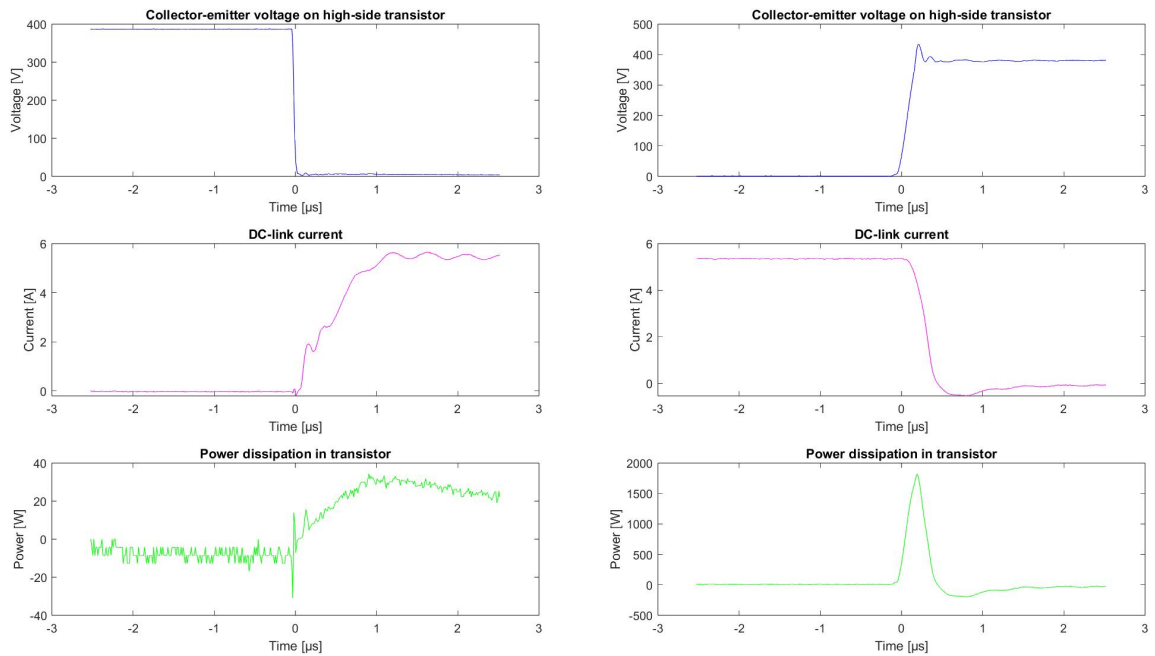
Figur 8.10:  $R_{gate} = 100\ \Omega$   
 Kilde: plottet i MatLab med data fra oscilloskop



Figur 8.11:  $R_{gate} = 50\ \Omega$   
 Kilde: plottet i MatLab med data fra oscilloskop



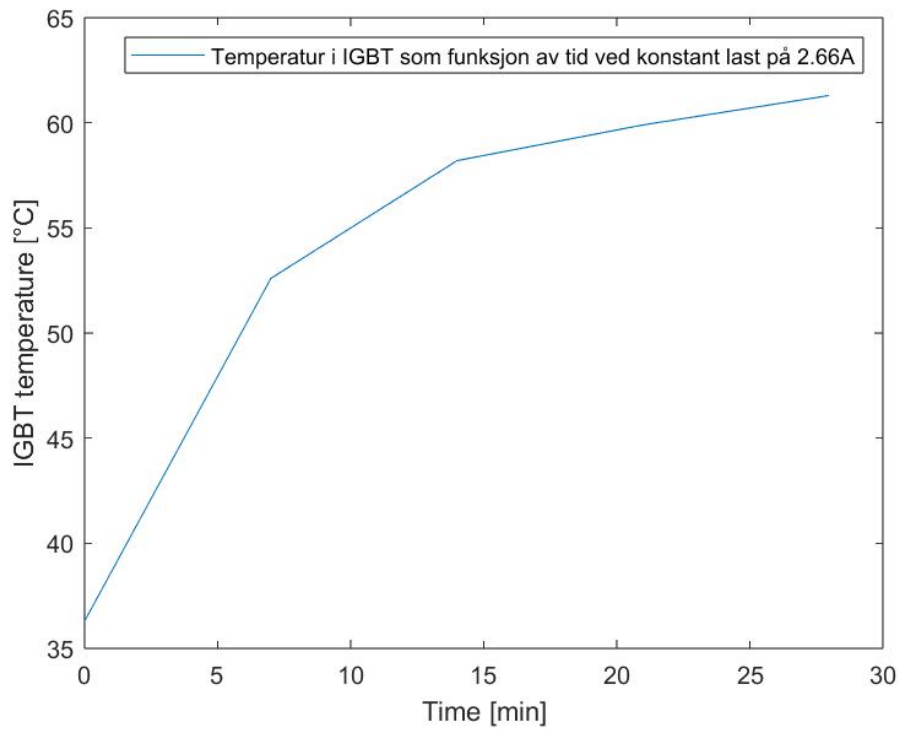
Figur 8.12:  $R_{gate} = 24.9\Omega$   
 Kilde: plottet i MatLab med data fra oscilloskop



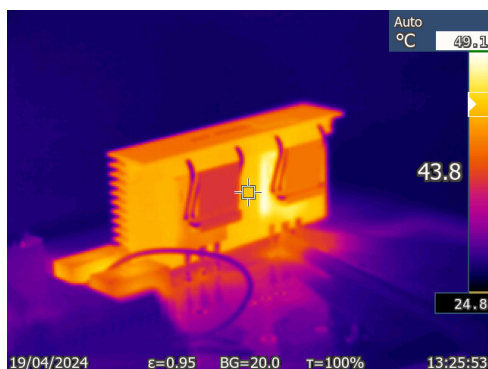
Figur 8.13:  $R_{gate} = 4.99\Omega$   
 Kilde: plottet i MatLab med data fra oscilloskop

Figur 8.15 viser to termografier som vi tok av transistorparet under testing. På disse termografiene var gate-motstanden  $100\Omega$ . For å forsikre at transistorene ikke ble overopphetet tok vi jevnlige termografier som dette.

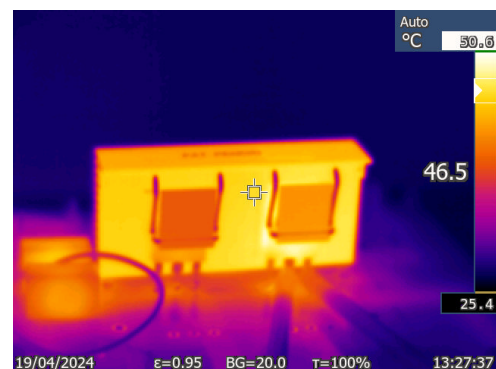
For å teste hvordan temperaturen i transistorene utviklet seg over tid hadde vi kretsen under drift i 30 min med samme last som er beskrevet over,  $512.6W$ . Gate motstanden var  $10\Omega$  under denne testen. I løpet av de 30 minuttene var den høyeste temperaturen vi målte  $63^\circ C$  8.14. Resultatet ble som vist i figur 8.14.



Figur 8.14: Varmeutvikling i IGBT som funksjon av tid ved  $I_{last} = 2.66A$  og  $R_{gate} = 10\Omega$



(a) Termografi av IGBT etter 5min



(b) Termografi av IGBT etter 7min

Figur 8.15: Kilde: Bilder tatt på kraftelektronikk lab

### 8.1.3 Test av H-bro som DC/DC omformer

Målet med å kunne drifte kretskortet som fullbro DC/DC omformer var for å kunne styre en DC-maskin. Før omformeren ble testet på en DC-maskin ble det utført grunnleggende tester av omformeren som fullbro DC/DC omformer. Disse testene er lagt fram i dette delkapittelet. I tillegg ble det gjort målinger av avgitt rippelspenning fra omformeren ved de to ulike styringsformene.

#### Grunnleggende test

Her har vi testet kretskortet i H-bro konfigurasjon ved både unipolar og bipolar styring. For å teste omformerens respons på ulike typer belastning ble det testet med både resistiv last og med induktiv last. Gate-motstanden var  $5\Omega$  under alle testene, og dødtiden var satt til  $12.3\mu\text{s}$ . Omformeren opererte konstant med følgende parameter under alle testene:

- $f_{sw} = 150\text{Hz}$
- $D = 0.8$
- $V_d = 400\text{V}$

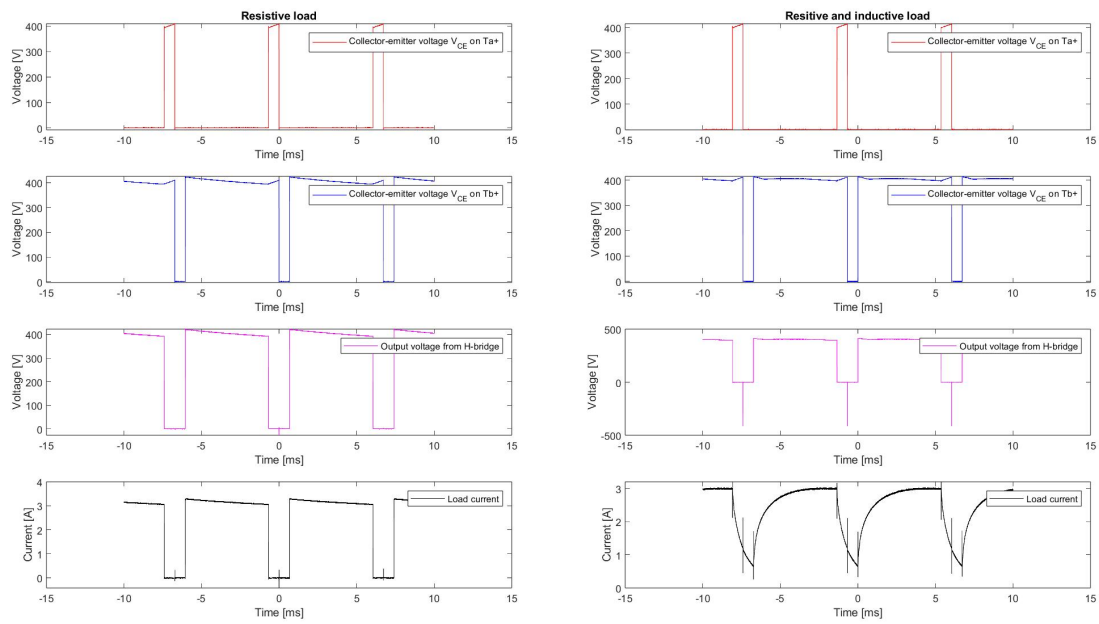
Som resistiv last benyttet vi en konveksjonsovn med en variabel motstand, hvor motstanden kan instilles til  $26.6\Omega$ ,  $42.0\Omega$  eller  $71.0\Omega$ . Videre hadde vi 2 stk effektmotstander, hvor den største var på  $81\Omega$  med maks effekt på  $790\text{W}$ , og den minste var på  $71\Omega$  med maks effekt på  $290\text{W}$ . For å simulere ulike laster koblet vi først inn varmeovnen i serie med de to bremsemotstandene.

Som induktiv last benyttet vi en *Terco Sweden Load Reactor MV1101*, som er en trefaset justerbar reaktor. Vi målte at induktansen per fase var cirka  $120\text{mH}$  på den laveste instillingen. For å redusere induktansen ytterligere koblet vi de tre spolene på reaktoren i parallell slik at den ekvivalente induktansen ble rundt  $40\text{mH}$ .

Figur 8.16 og 8.17 viser måleresultatene fra testen.



## Unipolar modulasjon

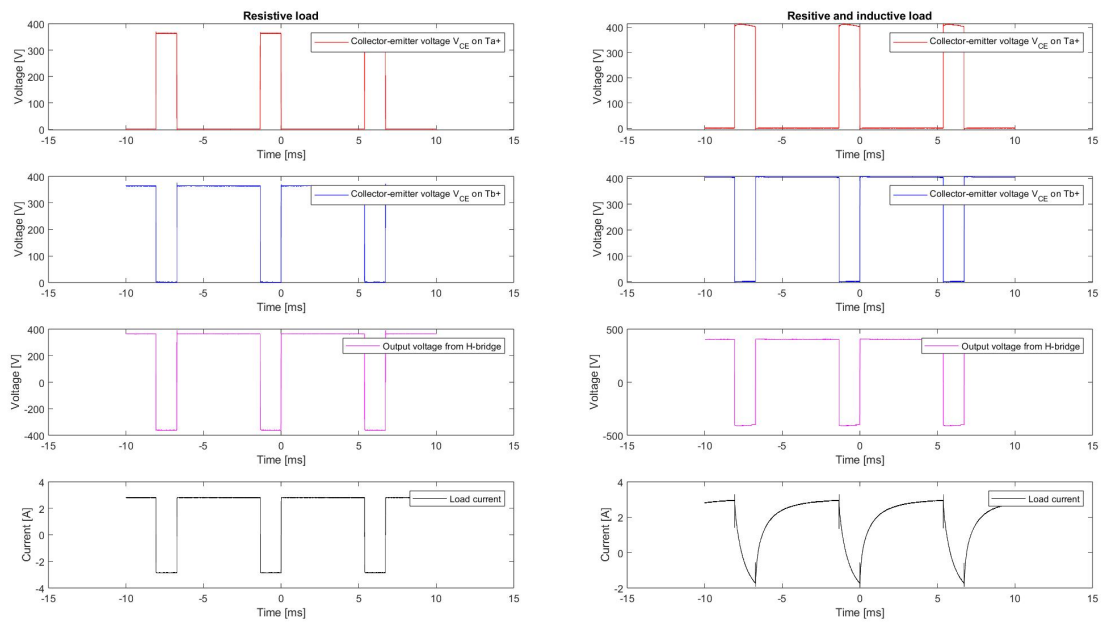


Figur 8.16: Målinger ved unipolar svitsjing med induktiv og resistiv last

$$(R_{load} = 126\Omega, L = 40mH)$$

Kilde: plottet i MatLab med data fra oscilloskop

## Bipolar modulasjon



Figur 8.17: Målinger ved bipolar svitsjing med induktiv og resistiv last

$$(R_{load} = 126\Omega, L = 40mH)$$

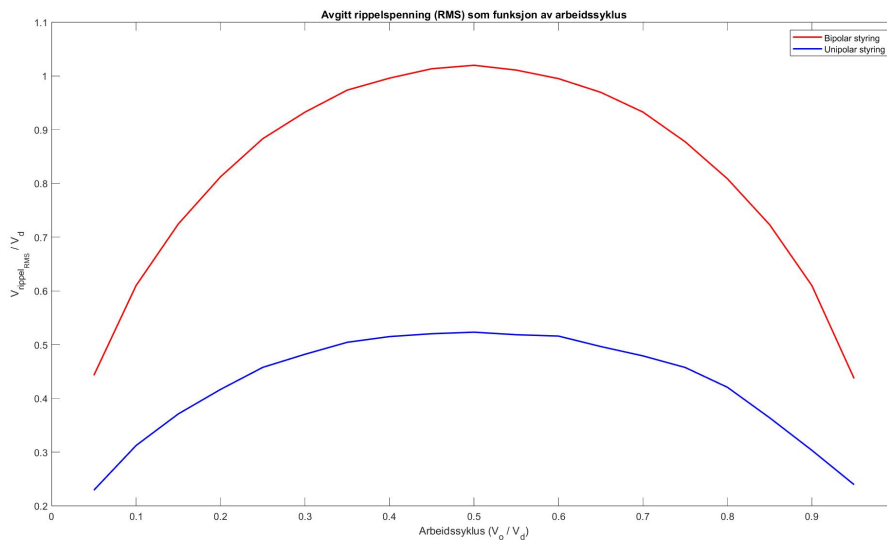
Kilde: plottet i MatLab med data fra oscilloskop

## Rippelspenning ved unipolar og bipolar styring

I denne testen har vi målt RMS- og snittspenning ved både bipolar og unipolar drift ved ulike arbeidssykluser, og deretter beregnet avgitt rippelspenning. RMS-verdien til avgitt rippelspenning kan beregnes etter følgende formel:

$$V_{ripple_{RMS}} = \sqrt{V_{out_{RMS}}^2 - V_{out_{AVG}}^2} \quad (8.1)$$

Under alle testene har vi holdt konstant DC-link spenning på 200V og svitsjefrekvens på 100Hz. Dødtiden var 12.3µs. Vi tok totalt 19 målinger for hver av de to styringsmetodene, hvor vi startet med en arbeidssyklus på 5% og økte med 5% for hver måling fram til vi nådde 95%. For å måle den avgitte snittspenningen og dens arbeidssyklus benyttet vi oscilloskop. Lasten var en konveksjonsovn. Til slutt plottet vi forholdet mellom målt rippelspenning og påtrykt DC-link spenning som funksjon av arbeidssyklus, og resultatet ble som vist i figur 8.1. Rådata fra målingene som ble gjort i denne testen er å finne i vedlegg G.1.



Figur 8.18: Avgitt rippelspenning (RMS) som funksjon av arbeidssyklus  
Kilde: plottet i MatLab med data fra oscilloskop

### 8.1.4 Test av H-bro som enfase DC/AC omformer

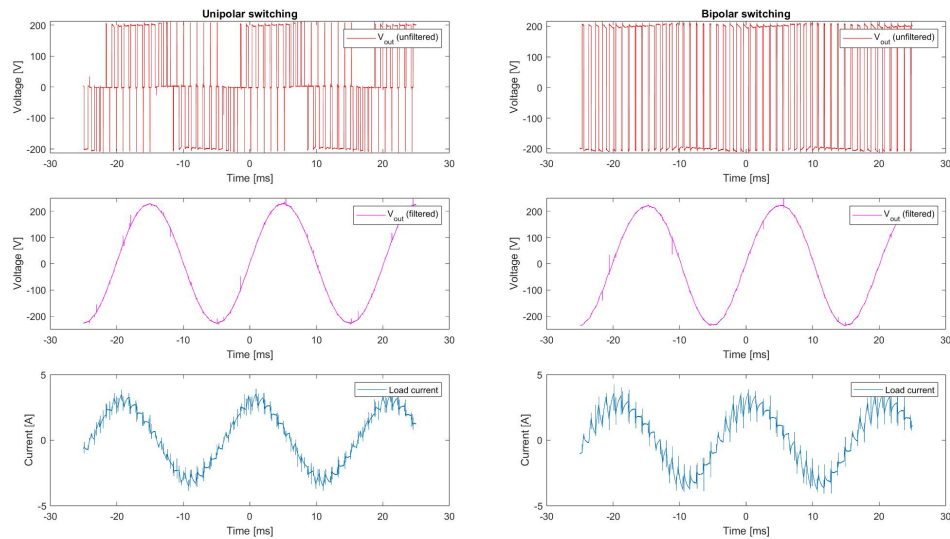
En H-bro kan også benyttes som en DC til enfase AC omformer. Modulasjonsalgoritmen som er valgt til testing er Sinus PWM (SPWM), og vi har valgt å teste med både bipolar og unipolar styring. For å tydeliggjøre grunnharmonisk avgitt spenning under måling, har vi her benyttet et LC-filter.

#### Grunnleggende test

Her har vi testet unipolar og bipolar styring av H-bro konfigurasjonen som enfase vekselretter. Testopsettet som vi benyttet her er gjennomgått i 8.2.1. Vi målte ufiltrert og filtrert avgitt spenning i tillegg til laststrømmen. Testen ble gjort med følgende parameter:

- $V_d = 200V$
- $f_{sw} = 1kHz$
- $f_1 = 50Hz$
- $L_{filter} = 40, mH, C_{filter} = 40, \mu F$
- $R_{last} = 200\Omega$
- $m_a = 0.8$

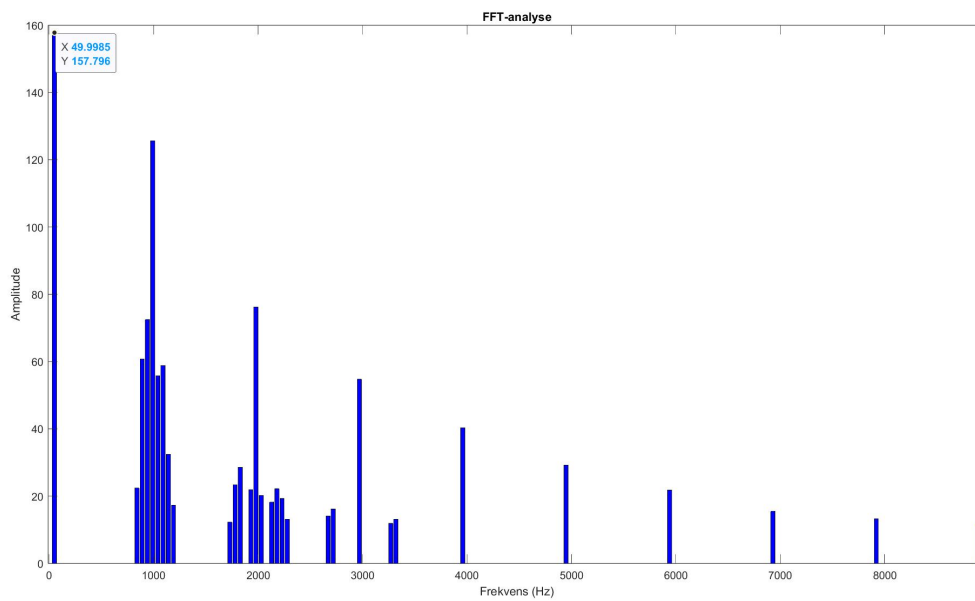
Resultat fra testene ble som vist i figur 8.19.



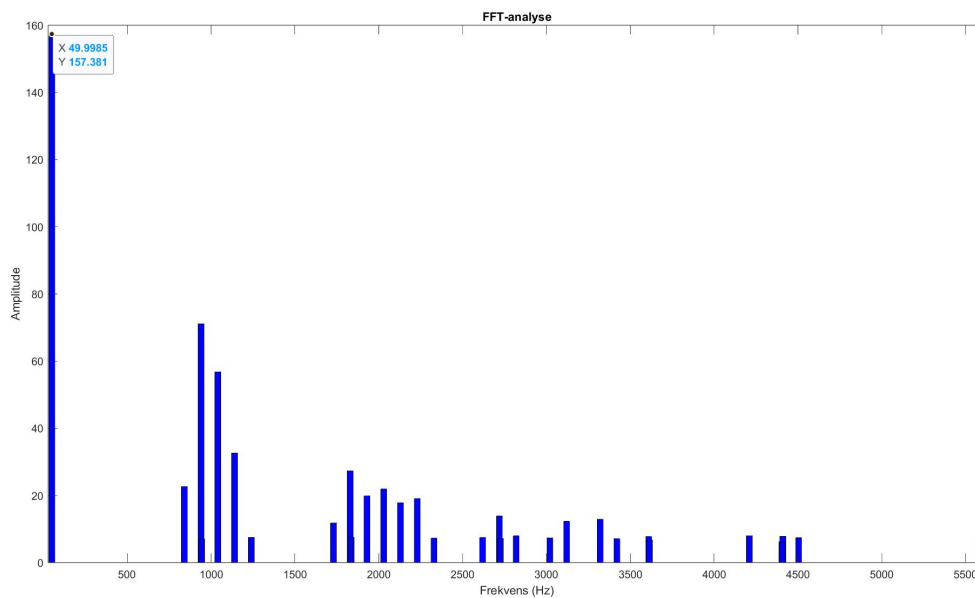
Figur 8.19: Målinger fra test av H-bro som enfaset vekselretter

*Kilde: plottet i MatLab med data fra Oscilloskop*

For å studere det harmoniske innholdet i avgitt spenning ved de to ulike styringsformene ble det foretatt en FFT-analyse av måledataene. MatLab ble benyttet til å utføre FFT-analysen, og koden som ble brukt er vist i vedlegg E.1.



Figur 8.20: FFT-analyse av **bipolar** svitsjing  
*Kilde: laget i MatLab med data fra oscilloskop*



Figur 8.21: FFT-analyse av målinger fra testing med **unipolar** svitsjing  
*Kilde: laget i MatLab med data fra oscilloskop*

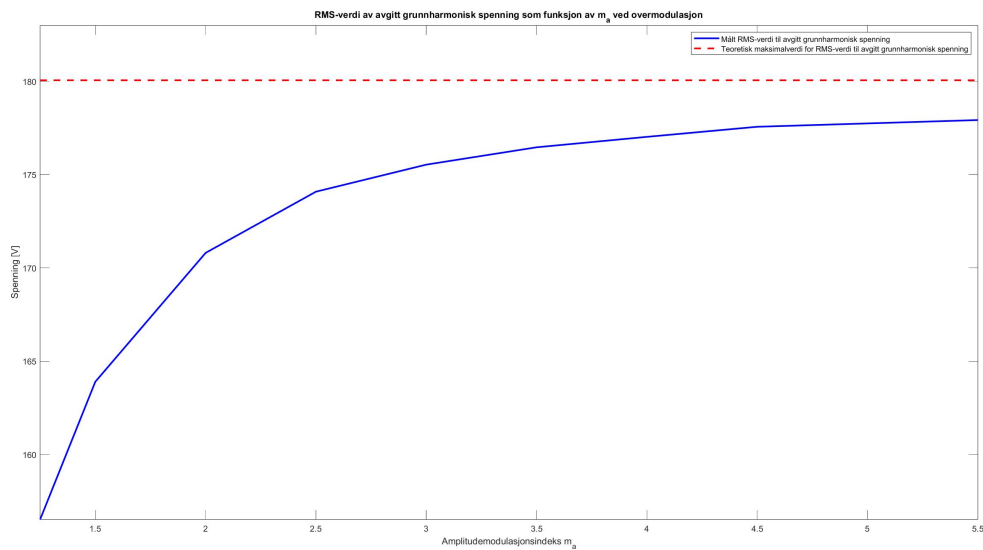
## Test av overmodulasjon

Målet med denne testen var å studere avgitt grunnharmonisk spenning og laststrøm ved overmodulasjon. Vi har benyttet bipolar styring under alle testene, og følgende parameter har blitt holdt konstant:

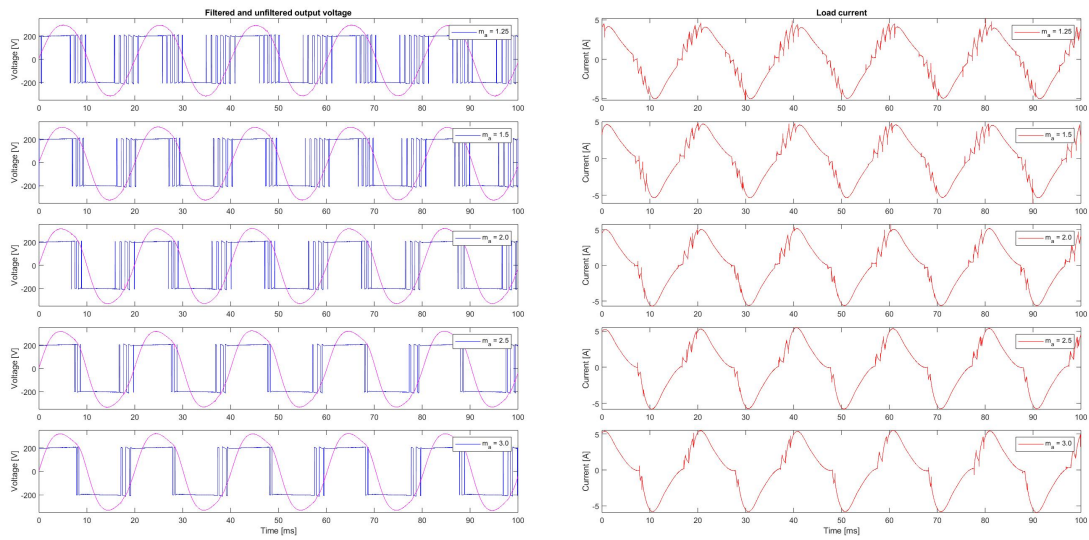
- $V_d = 200V$
- $f_{sw} = 1kHz$
- $f_1 = 50Hz$
- $L_{filter} = 40, mH, C_{filter} = 40, \mu F$
- $R_{last} = 200\Omega$

Vi startet med  $m_a = 1.25$ , og økte så  $m_a$  til vi nådde  $m_a = 5.5$ . Totalt tok vi 10 målinger. Figur 8.23 og 8.24 viser resultatene fra målingene som vi gjorde med oscilloskop.

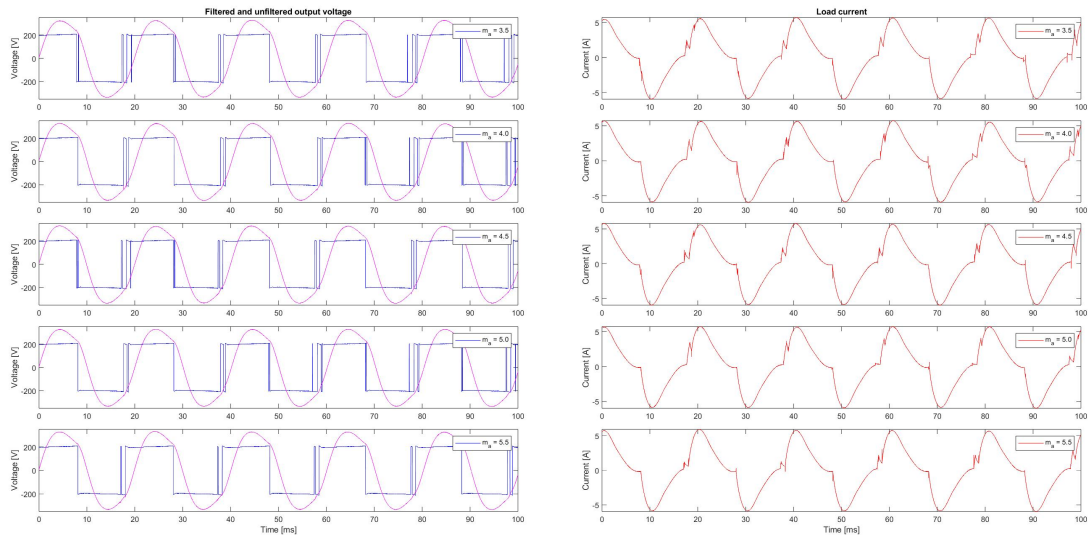
For hver av de 10 målingene benyttet vi MatLab-koden i vedlegg E.1 for å beregne RMS-verdien til den grunnharmoniske komponenten i avgitt spenning. Figur 8.22 viser resultatet av målingene. Vedlegg G.2 inneholder rådata fra målingene.



Figur 8.22: Avgitt grunnharmonisk spenning som funksjon av  $m_a$  ved overmodulasjon  
*Kilde: plottet i MatLab med data fra oscilloskop*



Figur 8.23: Ufiltrert avgitt spenning og filtrert laststrøm ved ulike  $m_a$   
 Kilde: plottet i MatLab med data fra oscilloskop



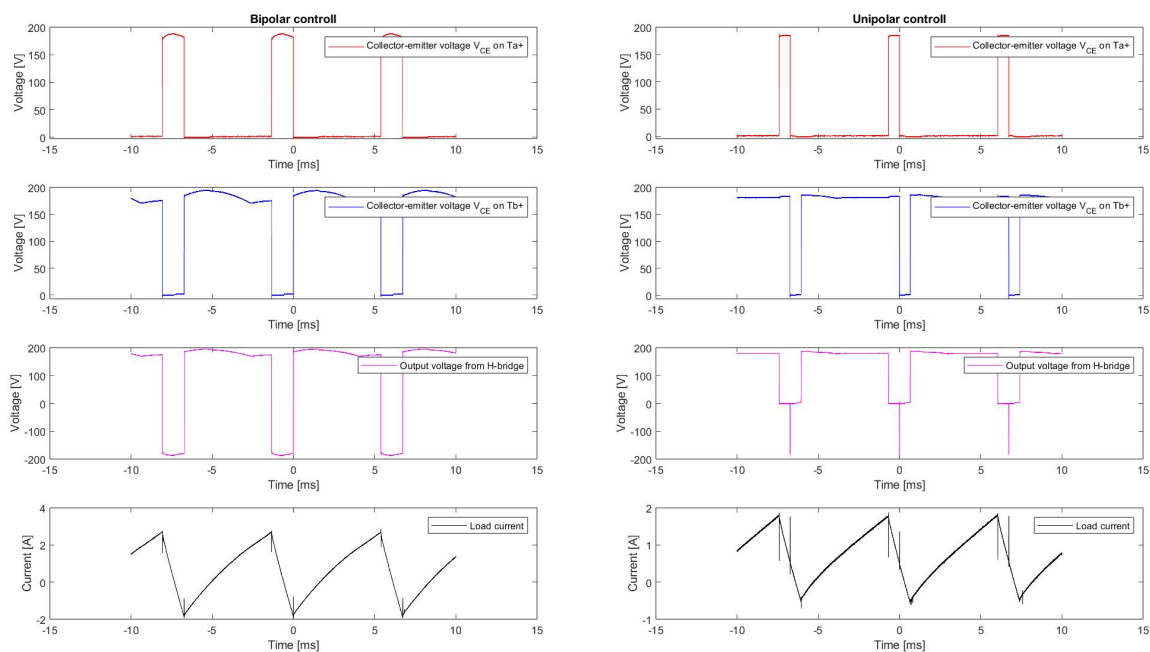
Figur 8.24: Ufiltrert avgitt spenning og filtrert laststrøm ved ulike  $m_a$   
 Kilde: plottet i MatLab med data fra oscilloskop

## 8.1.5 Test av DC-motor drift

Det er ønskelig å teste hvordan H-broen presterer med å drifte en DC-motor. Vi har her valgt å teste med både unipolar og bipolar styring. Det fysiske oppsettet for denne testen er beskrevet i 8.2.2. Omformerene opererte med følgende parameter både ved unipolar og bipolar styring:

- $V_d = 180V$
- $f_{sw} = 150Hz$
- $D = 0.8$
- $I_m = 200mA$  (Magnetiseringsstrøm)

Resultatene fra testene ble som vist i figur 8.25



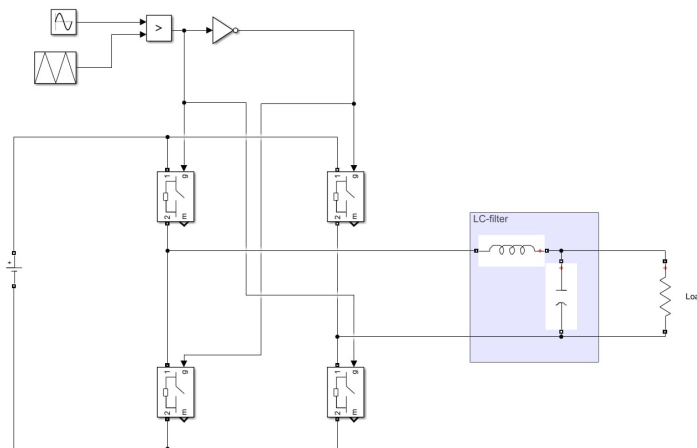
Figur 8.25: Målinger ved drift av DC-motor  
Kilde: plottet i Matlab med data fra oscilloskop



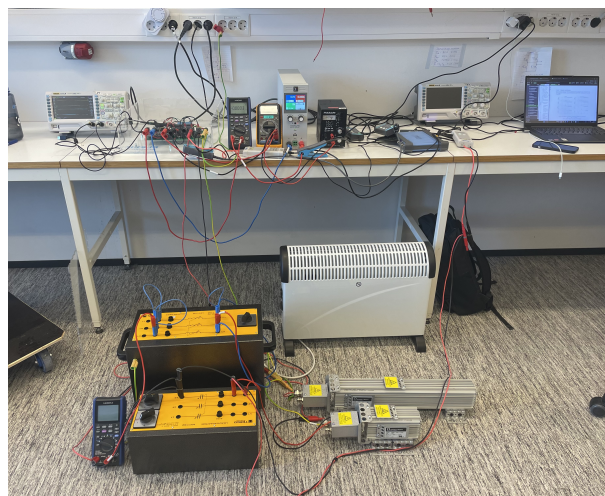
## 8.2 Testoppsett

### 8.2.1 Testoppsett for H-bro som enfaset vekselretter

For å teste H-broen som enfaset vekselretter gjorde vi først en simulering i SIMULINK som nevnt i kapittel 8.1.4. Der testet vi med ulike verdier for L og C, for å finne ut hvilket oppsett som ville gi oss best målinger.



Figur 8.26: SIMULINK modell av H-bro med LC filter



Figur 8.27: Praktisk oppkobling av testoppsett for H-bro med LC filter

På bildet 8.27 over ser vi 3 motstander koblet i serie, med en samlet motstand på  $225\Omega$ . I tillegg er det et LC filter med i oppkoblingen. Dette filteret fjerner de harmoniske komponentene med høye frekvenser, slik at man enklere kan lese av den grunnharmoniske komponenten med oscilloskop. På bildet ser man også 24V forsyning, og DC-link forsyningen. For å dokumentere data, bruker vi et oscilloskop som kobles til PC, for enkel overføring.

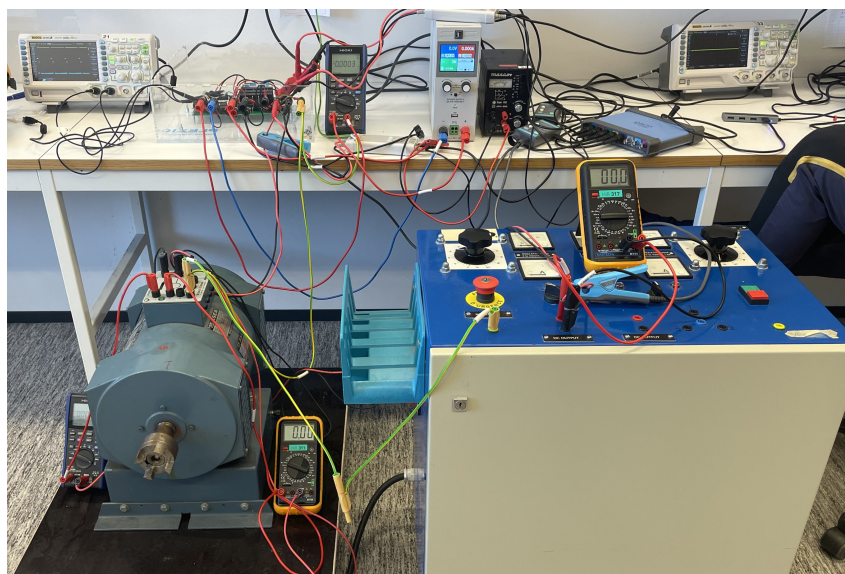
## 8.2.2 Drift av fremmedmagnetisert DC-Maskin

Testoppsettet under testing av DC-maskin besto av :

- Fremmedmagnetisert likestrømsmotor (DC-Maskin)
- DC - forsyning til å drive magnetiseringsstrømmen



Figur 8.28: Merkeskilt DC-maskin



Figur 8.29: Testoppsett ved drift av DC-Maskin

# Kapittel 9

## Analyse av testresultater

Dette kapitlet tar for seg analyse og drøfting av måleresultatene som ble lagt fram i kapittel 8.

### 9.1 Verifikasjon av PWM

#### 9.1.1 Verifikasjon av dødtid

Denne testen hadde som hovedformål å verifisere at det ble lagt inn en dødtid mellom svitsjingen av to transistorer på samme kanal (bein). Som forklart i kapittel om software så legges dødtiden inn som en sikkerhetsmekanisme for å forsikre at to transistorer på samme bein aldri er på samtidig. Figur 8.1 viser at dødtiden blir lagt inn skikkelig og at den innstilte dødtiden faktisk blir som forventet i praksis. I teorien skulle dødtiden ha vært  $12.3\mu\text{s}$ , men i praksis målte vi den til å være  $12.45\mu\text{s}$ . Vi anser dette som en suksess. Faktum at vi måler dødtiden til å være noe lengre enn forventet mener vi er bedre enn om vi hadde målt den til å være kortere enn forventet.

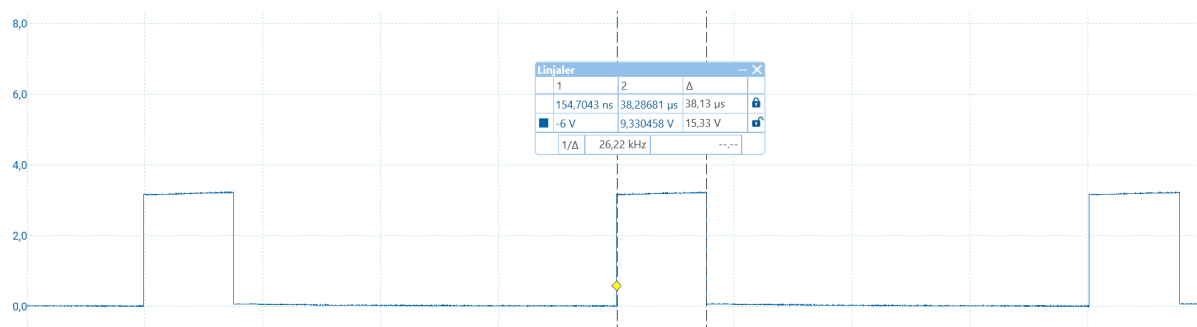
#### 9.1.2 Verifikasjon av styresignaler

I 8.1.1 ble det lagt fram måleresultater fra styresignalene ved de ulike modulasjonsteknikkene som vi hadde programmert inn i microcontrolleren. I kapittel 3 ble det lagt fram det teoretiske grunnlaget for styring av H-bro, og i kapittel 4 ble det lagt fram teorien for hvordan styresignalene ved trefase SPWM og SVPWM genereres. Det er tydelig at målte styresignalene i figur 8.2-8.7 genereres slik som teorien tilsier.

### 9.1.3 Måling av beregningshastighet

I 8.1.1 ble det lagt fram resultater fra testen vi gjorde for å måle beregningshastighet for trefase SPWM med sinustabell og `sin()`-funksjon. Resultatet fra testen som er vist i figur 8.8 viser at microcontrolleren bruker rundt  $12.5\mu\text{s}$  på å utføre en programiterasjon dersom en benytter oppslagstabell, mens fra figur 8.9 kan vi se at den bruker rundt  $125\mu\text{s}$  ved bruk av `sin()`-funksjonen. Dette tilsier at det er over 10 ganger raskere å benytte oppslagstabell framfor `sin()`-funksjonen. Dette føler vi at gir oss et solid grunnlag for å konkludere med at det er langt mer optimalt å benytte oppslagstabell enn sinusfunksjon.

For å kunne sammenligne hastigheten til SPWM-algoritmen og SVPWM-algoritmen utførte vi også en beregningshastighetstest på SVPWM-algoritmen. På dette tidspunktet hadde vi bestemt oss for å benytte oppslagstabell da vi ikke lengre anså `sin()`-funksjonen som et alternativ. Derfor ble det også benyttet oppslagstabell i koden for SVPWM-algoritmen. Figur 9.1 viser resultatet av hastighetstesten for SVPWM med oppslagstabell, som tilsier at denne algoritmen bruker rundt  $38\mu\text{s}$  på en iterasjon. Dette er over tre ganger så lang tid som SPWM-algoritmen med sinustabell. På forhånd av testen forventet vi at SVPWM skulle bruke lengre tid en SPWM da SVPWM krever flere og mer intensive beregninger.



Figur 9.1: Beregningshastighet for SVPWM-algoritme med sineTable[]

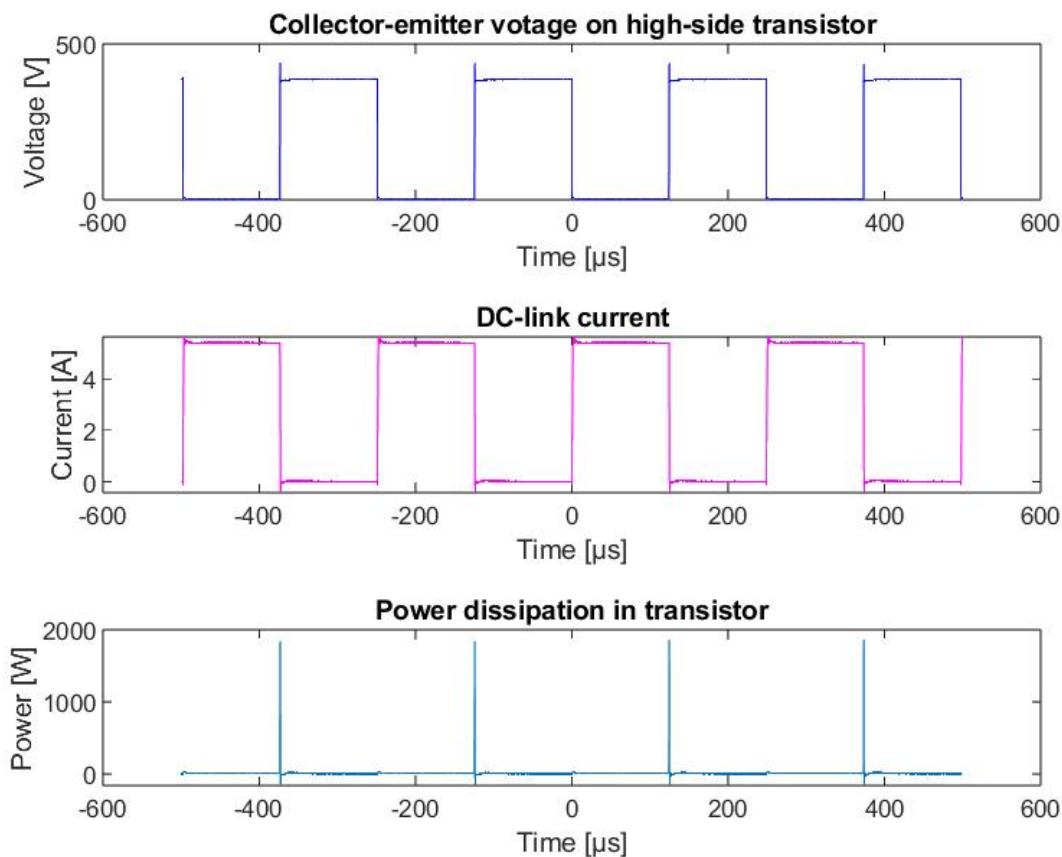
## 9.2 Test av ulik gate-motstand

Dette var den første testene vi gjorde etter å ha verifisert av omformerene fungerte som den skulle ved å sjekke dødtid og styresignal. Testen hadde som formål å bestemme hvilke gate motstand vi skulle benytte på hovedkortet. Vi hadde en forventning om at lavere gate motstand ville gi raskere svitsjing av transistorene, som følge av at gate-kapasitansen ville lade seg opp raskere. Dette forventet vi at skulle gi lavere svitsjetap i transistorene, og som følge av lavere svitsjetap skulle det derfor også bli mindre varmeutvikling i transistoren.

For å studere effekten av ulik gate motstand tok vi målinger av både stigende og fallende flanke på spenningen mellom kollektor og emitter på high side transistor. Når kollektor-emitter spenningen er 0V tilsier dette at transistoren leder strøm, mens når den er lik DC-link spenningen så tilsier det at transistoren er av. Figur 8.10, 8.11, 8.12 og 8.13 viser målingene vi gjorde med oscilloskopet ved ulik gate motstand. Ved å studere disse figurene ser man at effektomsetningen ved stigende flanke på  $V_{ce}$  ser ut til å være tilnærmet lik for alle målingene. Dette skjer fordi transistoren, som målingene viser, bruker tilnærmet like lang tid på å skru seg av. Men effektomsetningen ved fallende flanke på  $V_{ce}$  derimot, når transistoren skrur seg på, ser ut til å være påvirket i stor grad av de ulike gate motstandene. Studerer man målingen med  $R_{gate} = 100\Omega$  i figur 8.10 ser man at det er en stor transient i effekt-omsetningen rundt  $t = 0$  hvor den når rundt 80W i et lite øyeblikk. Dersom man sammenligner med målingen i figur 8.11 med  $R_{gate} = 50\Omega$  vil man se at denne transienten er kraftig redusert, med en toppverdi på rundt 30W. Ved  $R_{gate} = 24.9\Omega$  er transienten enda mer redusert, og ved  $R_{gate} = 4.9\Omega$  er den så og si eliminert.

Tabell 8.2 viser at det er en betydelig korrelasjon mellom størrelse på gate-motstand og temperaturøkning i transistoren. Ved hjelp av MatLab har vi beregnet snitt-effekten omsatt i transistoren ved ulik gate-motstand, disse målingene er vist i tabell 8.1 og viser en klar sammenheng mellom effekttap og gate motstand. Dette var som vi hadde forventet på forhånd.

Konklusjonen fra målingene er at effektomsetningen i transistoren er betydelig større når transistoren skrur seg av, i forhold til hva den er når transistoren skrur seg på. Effekten i transistoren når en max-verdi på rundt 1,7kW i det den skrur seg av, i forhold til max 80W når den skrur seg på. Effekttoppen på 1,7kW ved stigende flanke ble observert på alle målingene vi gjorde med forskjellige gate-motstander, som medfører at vi tolker dette som at den høye effektomsetningen ved stigende flanke er uavhengig av størrelsen på gate motstanden. Størrelsen på gate motstanden blir da i praksis kun avgjørende for effektomsetningen når transistoren skrur seg på, ikke når den skrur seg av. På basis av dette valgte vi å benytte gate motstand på  $4.99\Omega$  på sluttproduktet. Figur 9.2 viser effektomsetningen i high side transistoren fra testen i kapittel 8.1.2, men her med en bredere tidsakse. Figuren gir en god illustrasjon av signifikansen til effekten omsatt i transistoren når den går av versus på.



Figur 9.2: Effekttap i IGBT ( $R_{gate} = 100\Omega$ )  
 Kilde: plottet i Matlab med data fra oscilloskop

### 9.3 Test av H-bro som DC/DC omformer

Dette kapittelet har til hensikt å drøfte over resultatene som kom fra testene vi utførte med H-bro som fullbro DC/DC omformer. Disse testene er lagt fram i kapittel 8.1.3. Alt i alt kan vi konkludere med at testresultatene endte med å bli som det teoretiske grunnlaget tilsier.

Figur 8.16 og 8.17 viser tydelig hvordan laststrømmen påvirkes av den induktive lasten ved de to forskjellige styringsmetodene. Fra teorien vet vi at ved induktiv last kan ikke laststrømmen endres momentant slik som den gjør ved rein resistiv last. Denne teorien kan vi bekrefte med målingene som er lagt fram i figur 8.16 og 8.17, da vi ser at laststrømmen øker mye mer gradvis ved induktiv last.

Som vist i figur 8.16 observerte vi negative transienter på den avgitte spenningen ved unipolar styring og induktiv last. Transientene oppsto i det  $Ta+$  gikk på samtidig som  $Tb+$  gikk av. Vi har ikke klart å identifisere årsaken til dette, men vi vet at den induktive lasten har del i årsaken. Dette kan vi vite fordi vi ikke observerte noen som helst negative transienter ved rein resistiv last.

Den siste testen vi gjorde av H-bro som fullbro DC/DC omformer var å sammenligne

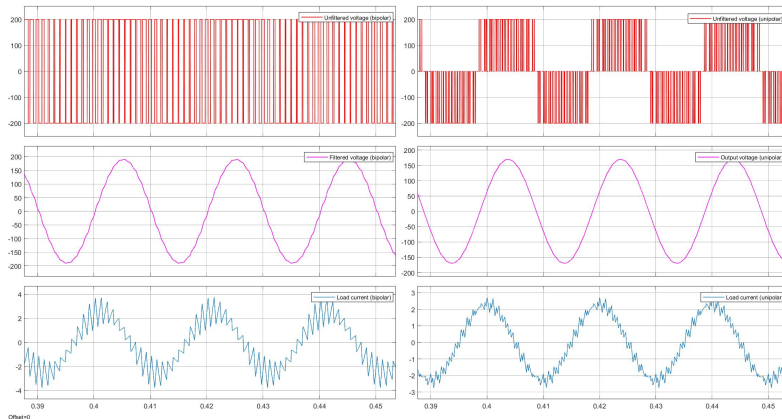
RMS-verdi til avgitt rippelspenning ved unipolar og bipolar styring. Denne testen er beskrevet i 8.1.3. Her forventet vi fra teorien at den avgitte rippelspenningen skulle være betydelig høyere ved bipolar styring enn hva den ville være med unipolar styring. Figur 8.18 illustrerer at teorien stemmer med praksis. Målingene vi gjorde viser at den høyeste rippelspenningen ved unipolar styring er 50% av DC-link spenningen, mens med bipolar styring når rippelspenningen 100% av  $V_d$ . Dette er et viktig resultat som sier noe om ulempene med bipolar styring. Til tross for at bipolar styring utnytter DC-link spenningen mer effektivt gir den altså høyere rippelspenning, noe som kan være et problem i samtlige tilfeller. Det er mulig å redusere denne rippelspenningen med et filter, men dette vil da føre til økte kostnader i praksis.



## 9.4 Test av H-bro som DC/AC omformer

### 9.4.1 Grunnleggende tester

I kapittel 8.1.4 ble det utført test av H-bro som enfase DC/AC omformer med både unipolar og bipolar styring. Figur 8.19 viser målingene vi gjorde med oscilloskop. For å verifisere målingene våre har vi simulert testen i SimuLink. Figur 9.3 viser resultatet fra simuleringen.



Figur 9.3: Simulering av test i kapittel 8.1.4

*Kilde: simulert i Matlab SimuLink*

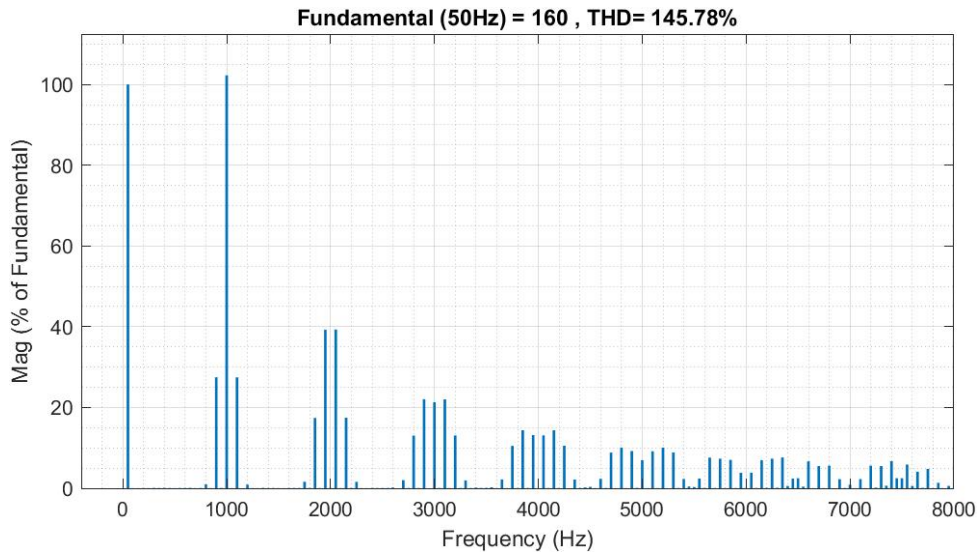
Dersom en sammenligner de reelle målingene i figur 8.19 med de simulerte målingene i figur 9.3 ser man at målingene er veldig like.

### FFT analyse

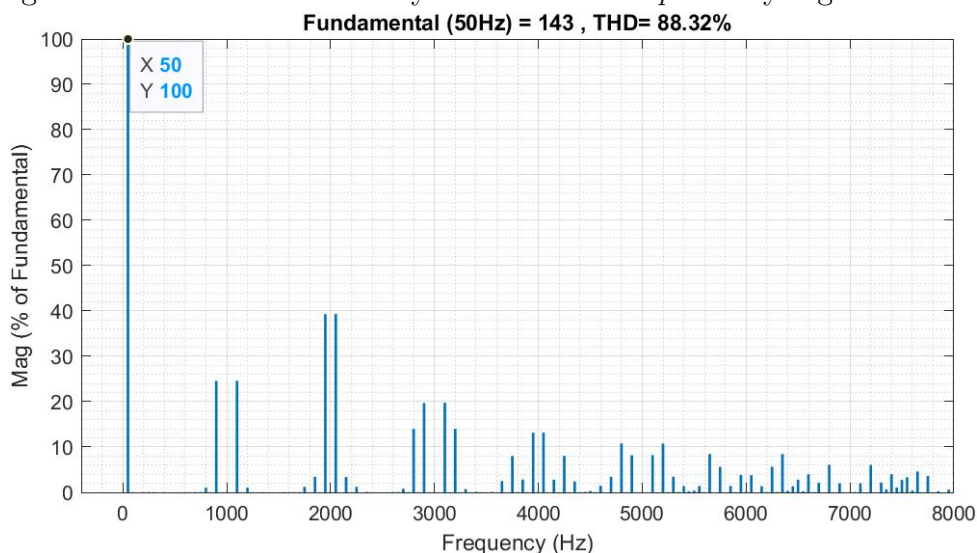
I kapittel 8.1.4 ble det lagt fram resultater fra FFT-analyse av målingene vi gjorde ved unipolar og bipolar SPWM. Figur 8.20 og 8.21 viser at begge signalene har grunnharmonisk frekvenskomponent på 50Hz med amplitude på rundt 157.3V. Dette er å forvente fra teorien ettersom at testene ble utført med  $m_a = 0.8$  og  $V_d = 200V$  som tilsier at amplitude på grunnharmonisk skulle bli  $0.8 \cdot 200V = 160V$ . Videre viser frekvensspekterene i figur 8.20 og 8.21 at de overharmoniske komponentene ligger seg som sidebånd på multipler av svitjefrekvensen (som var 1kHz), noe som også er å forvente fra teorien.

For å bekrefte FFT-målingene fra kapittel 8.1.4 gjorde vi tilsvarende FFT-analyser på SimuLink modellen. Figur 9.5 og 9.4 viser resultatene fra disse simulerte FFT-analysene. FFT-analysene viser at de overharmoniske komponentene på og rundt svitjefrekvensen er betydelig større ved bipolar styring i forhold til unipolar styring. Dette fenomenet er enkelt å observere både i de simulerte målingene i 9.5 og 9.4 samt i de reelle målingene i figur 8.20 og 8.21. Teoretisk sett vil svitjefrekvensen oppfattes som det dobbelte ved bipolar styring, noe som medfører at de laveste overharmoniske komponentene vil oppstå ved det dobbelte av den faktiske svitjefrekvensen [5, s. 217]. Dette forklarer hvorfor det er lavere harmonisk innhold ved unipolar styring.





Figur 9.4: **Simulert** FFT-analyse av test med *bipolar* styring i SimuLink



Figur 9.5: **Simulert** FFT-analyse av test med *unipolar* styring i SimuLink

*Kilde: laget i MatLab SimuLink*

## 9.4.2 Test av overmodulasjon

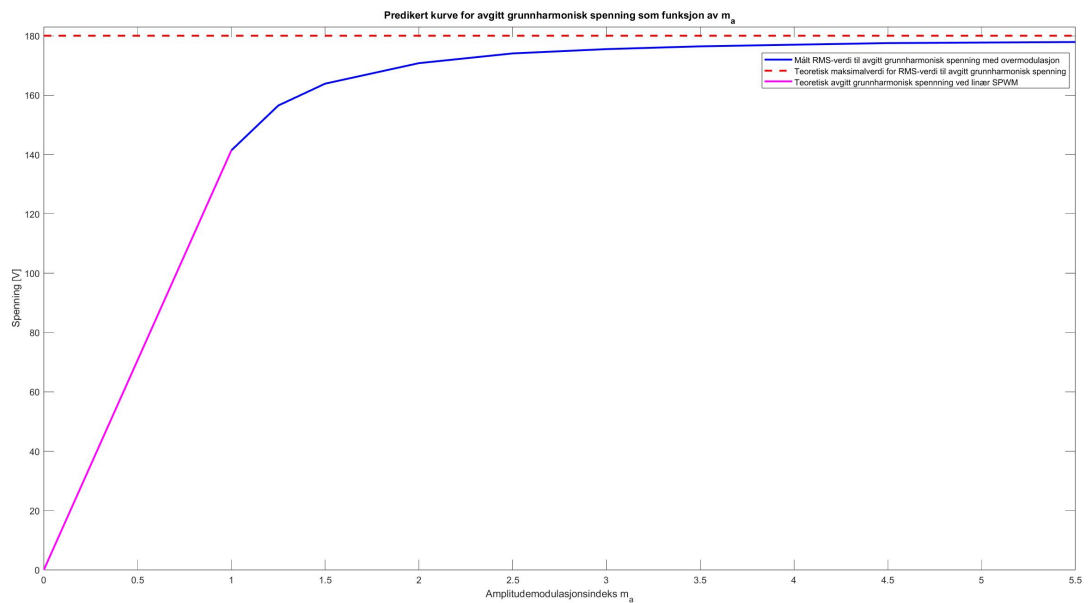
Denne testen hadde som formål å studere avgitt grunnharmonisk spenning som funksjon av amplitude-modulasjonsforholdet  $m_a$  ved overmodulasjon. Ved  $m_a > 1$  er det krevende å beregne hva den avgitte grunnharmoniske spenningen vil bli, og det var dette som var vår motivasjonen vår bak denne testen. Målingene vi gjorde av avgitt grunnharmonisk spenning (RMS) som funksjon av  $m_a$  ved overmodulasjon er vist i figur 8.22. Målingene viser at  $\frac{dV_{o1}}{dm_a}$  er høy mellom  $m_a = 1.0$  og  $m_a = 2.5$ . Etter  $m_a = 2.5$  gattes kurven ut, og den avgitte grunnharmoniske spenningen konvergerer mot maksimale teoretisk verdi ved ren firkantpuls, som er rundt 180V RMS ved  $V_d = 200V$ .

For å predikere hvordan kurven  $V_{o1}(m_a)$  ville ha sett ut dersom vi hadde tatt tilstrekkelig antall målinger av avgitt grunnharmonisk spenning ved lineær modulasjon også (noe vi

ikke gjorde), benyttet vi MatLab. Vi laget en vektor som inneholdt *teoretiske* verdier av avgitt grunnharmonisk spenning ved lineær modulasjon med  $V_d = 200V$ . Den teoretiske avgitte grunnharmoniske spenningen beregnet vi ut fra formelen:

$$V_{o1} = m_a \cdot V_d \cdot \frac{1}{\sqrt{2}} \quad (9.1)$$

Deretter plottet vi de teoretiske verdiene ved lineær SPWM i samme plot som de målte verdiene vi hadde av  $V_{o1}$  ved overmodulasjon. Resultatet ble som vist i figur 9.6. Den rosa kurven i figur 9.6 representerer de teoretiske verdiene for  $V_{o1}$  ved lineær SPWM, mens den blå kurven representerer de reelle målingene vi gjorde av  $V_{o1}$  ved overmodulasjon.



Figur 9.6: Predikert kurve for  $V_{o1}(m_a)$   
*Kilde: plottet i MatLab*

### 9.4.3 Test av DC-motor

Siden vi ikke kunne teste omformeren på en trefaset asynkronmaskin besluttet vi heller å teste den på en DC-maskin. Kapittel 8.1.5 legger fram resultatene. Resultatene ble mye som forventet, og motoren gikk som den skulle. Dette er vi veldig fornøyd med.

Ved unipolar styring av DC-maskinen observerte vi, som vist i figur 8.25, negative transienter på den avgitte spenningen fra omformeren. Dette minner veldig om målingene vi gjorde på RL-last med unipolar DC/DC. Som tidligere forklart antar vi at dette kommer av induktansen i motorviklingene, men dette er ikke noe vi kan si med fullstendig sikkerhet.

Dette prosjektet har hatt som hovedmål å videreutvikle en trefase DC til AC omformer ved å utarbeide et nytt kretskort, i tillegg til å videreutvikle software for styring av omformeren. Det nye kretskortet som ble utviklet i forbindelse med dette bachelorprosjektet viser en betydelig arealeffektivisering i forhold til kortet som ble utviklet våren 2023. Dette var et av hovedmålene med prosjektet, og vi føler nå at vi kan konkludere med at dette målet er oppnådd. Til tross for at alle tre fasene i kretskortet ikke ble testet samtidig, har vi fortsatt validert at alle tre fasene virker som de skal. Dette kan vi si fordi vi utførte gate-motstand testen på L1, mens H-broen var koblet mellom L2 og L3. Dermed har vi i praksis testet alle tre fasene og validert at de fungerer som forventet.

Etter mye laboratorietesting har vi validert at omformeren kan benyttes i H-bro konfigurasjon for drift av DC-maskiner. Med dette har vi utvidet bruksområdene for kretskortet. Den tidligere gruppen testet omformeren på en trefaset asynkronmaskin med både trefase S-PWM og SV-PWM. Dette fikk vi ikke muligheten til å gjøre nå i vår ettersom at den ene fasen (L1) ble defekt. Defekten skyldtes feil under lodding. Basert på målingene til den tidligere gruppen kan vi indirekte konkludere med at 2024-versjonen av omformeren også vil kunne benyttes til motordrift av asynkronmaskin. Vi har også validert at omformeren fungerer bra som fullbro DC/DC omformer, samt enfaset vekselretter. Med dette sagt kan vi konkludere med at produktet som vi har bidratt til å utvikle, kan være fin å bruke som laboratoriemodell for diverse laboratorietester. Laboratorietestene vil i all hovedsak være relevant for studenter innenfor elektro-utdanningene ved HVL.

Etter å ha sammenlignet unipolar og bipolar styring av H-bro som fullbro DC/DC omformer fikk vi god forståelse for disse to modulasjonsteknikkene. Vi så hvordan avgitt rippelspenning var ulik for de to styringsformene. I tillegg så vi hvordan avgitt laststrøm ble påvirket av induktiv og resistiv last. Testene vi gjorde på omformeren som enfaset vekselretter viste hvordan unipolar styring gir lavere harmonisk innhold i avgitt spenning enn hva bipolar styring gjør. Vi så også hvordan avgitt grunnharmonisk spenning ved overmodulasjon utviklet seg som funksjon av amplitude-modulasjonsforholdet, og med grunnlag i dette predikerte vi hvordan kurven  $V_{o1}(m_a)$  så ut.

En interessant observasjon vi gjorde oss under testingen var hvordan ulike svitsjefrekvenser skaper ulike lyder. Høy svitsjefrekvens gir naturligvis skarpe og høyfrekvente toner, mens lavere svitsjefrekvens gir mørkere og mer bråkete tone. Dette var noe vi ikke hadde tenkt over på forhånd av testingen, men som vi fort la merke til. Denne lærdommen ga oss bedre forståelse for valg av svitsjefrekvens i praktiske applikasjoner, og ga oss god forståelse for hvorfor det ofte er ønskelig å velge svitsjefrekvens over 20kHz slik at menneskeøret ikke oppfatter lydbølgene.

For å oppsummere er vi veldig fornøyd med resultatene som har blitt lagt fram i denne rapporten. Vi føler at vi har bidratt til å forbedre produktet i stor grad, og at vi sitter igjen med veldig mye ny kunnskap. Videre håper vi at studenter ved HVL vil nytte godt av produktet ved laboratorieøvelser, og vi oppfordrer med glede til at en ny framtidig bachelorgruppe vil ta for seg videreutvikling av dette produktet.

# Forslag til videre arbeid

Dette kapittelet har som formål å legge fram forslag til forbedringer av produktet.

## 9.4.4 Videreutvikling av HMI

HMI-kortet som vi utviklet ble aldri testet i praksis. Det ble heller aldri utviklet en kode for styring av omformeren via HMI. Årsaken til dette var mangel på tid. Dette har dermed stort forbedringspotensiale. Vi vil her gå gjennom forslag til hva HMI kortet kan brukes til.

### HMI-styring av fullbro DC/DC omformer

Et forslag her vil være å skrive en kode som gjør det mulig å justere avgitt spenning fra omformeren ved fullbro DC/DC omformer drift. Encoderen kan benyttes til å justere, slik at en øker den avgitte spenningen ved å rotere encoderen med klokken og reduserer den ved å vri encoderen mot klokken. Et annet forslag er å programmere en trykknapp til å endre polariteten på den avgitte DC-spenningen fra omformeren, slik at en kan endre dreieretningen på rotoren til en DC-maskin.

### HMI-styring av trefase vekselretter

Som nevnt tidligere i rapporten fikk vi aldri mulighet til å teste omformeren i trefasedrift. Her er det mange spennende muligheter, og et hav av ting som går an å programmeres inn. Vi har valgt å liste opp et par funksjonaliteter som vi mener hadde vært interessante å implementere.

- Utvikle softstarter-algoritme for å redusere startstrømmen til asynkronmaskin, ved å gradvis øke frekvensen i det omformeren starter opp
- Implementere 'mer avanserte' motorstyrings algoritmer som for eksempel felt-orientert kontroll (FOC), hvor en i tillegg til å måle fasestrømmer også måler rotorposisjon
- Gjøre det mulig å endre dreieretning på rotor ved å trykke på en av knappene på HMI kort

## Utvikling av menysystem for HMI

Ettersom at omformerer kan driftes med flere forskjellige modulasjonsteknikker, er et forslag å legge opp koden slik at brukeren kan velge hvilken modulasjonsteknikk omformerer skal driftes i fra HMI-kortet. En måte å gjøre dette på er ved å utvikle et menysystem. Et forslag er å lage en 'side' for hver av de forskjellige modulasjonsteknikkene.

### 9.4.5 Testing og programmering av målesystemer

Kretskortet er utrustet med målesystem for måling av DC-link strøm og spenning samt strøm i hver av fasene. Disse målesystemene fikk vi aldri tid til å teste i praksis. Det er dermed et stort rom for forbedring på dette området. Dette vil kreve utvikling av kode for å prosessere måledata fra sensorene. For eksempel kan en utvikle en algoritme for å kontinuerlig beregne RMS-verdien til målt DC-link strøm. I forbindelse med målesystemene er et forslag å programmere LCD-displayet på HMI-kortet til å vise måleverdier.

### 9.4.6 Videre testing og utvikling av bremsehopper

Kretskortet som ble utviklet våren 2023 av den tidligere gruppen var utrustet med en krets for å håndtere energi fra regenerativ drift av motor. Denne kretsen valgte vi å fjerne for å spare plass. Vi besluttet at denne kretsen skulle være på et eget kretskort, men dette kretskortet rakk vi aldri å utvikle. Dette gjør dermed at omformerer, slik designet er i 2024 versjonen, ikke er i stand til å håndtere overflødig energi fra generator drift av motor. Dette er ikke ideelt, og vi anbefaler derfor at det prioriteres å designe et kretskort for bremsehopper for å løse dette problemet.

### 9.4.7 Utvikling av likeretter

For å kunne forsyne omformerer fra en vekselstrømskilde, som for eksempel strømmettet, vil det naturligvis være behov for en likeretter. I den forbindelse vil et forslag til videre utvikling av produktet være å designe en likeretter. Da vil en kunne utvikle produktet til å bli en AC- til AC\* omformer, som per definisjon er en frekvensomformer. Forslaget vårt er å designe et eget kretskort for likeretteren. Det vil dermed være mulighet å designe en beholder bestående av to etasjer, hvor den første etasjen er likeretteren som forsynes via stikkontakt eller lignende og andre etasje er vekselretteren som forsynes fra likeretteren. HMI-kortet kan festes på siden av beholderen slik at brukeren enkelt har tilgang til det. I forbindelse med likeretteren er et forslag å designe et 'Power Factor Correction' system (PFC).

### 9.4.8 Utvikling av filter

For å redusere det harmoniske innholdet i avgitt spenning fra omformerer kan en benytte et filter. Dette er ønskelig i mange applikasjoner, og vi mener derfor at det ville vært nyttig å utvikle et filter. Forslag til filter kan være et LC filter eller et LCL filter. Vi mener det vil være naturlig å utvikle et eget kretskort for filteret som enkelt kan tilkobles output fra omformerer.

Del IV

Vedlegg

# Tillegg A

## Utgiftspost for prosjektet

Her er en liste over hvilke komponenter vi har bestilt inn våren 2024. De resterende komponentene vi har brukt som ulike motstander og kondensatorer er utstyr vi har fått fra lager på HVL. Listen under inneholder bestillingen vi gjorde, og antallet komponenter er basert på å kunne lage 2 eksemplarer med de delene som allerede var her fra den forrige bachelorgruppen.

Tabell A.1: Liste med komponenter brukt i omformereren

Produkt	Merke	Modell nr.	Ant:	Stk.pris:
Gatedriver	Infineon	2ED020I12F2XUMA1	3	60,64
IGBT transistor	Infineon	IKW40N120H3	6	88,36
Kjøleribbe	Ohmite	C247-075-3AE	3	99,24
DC-link kondensator	TDK	B32776G0146K00	2	122,5
DC/DC isolert	Maurta	MGJ2D241509SC	6	90,67
Strømmålingschip	Infineon	TLI4971-A025T5-E0001	4	61,56
Optisk isolert forsterker	Broadcom	ACPL-C87B-000E	1	90,67
DC/DC omformer	Maurta	NTE0505MC	1	91,13
Diode til gatedriver	STMicroelectronics	STTH112U	6	6,77
Schottkey diode	NEXPERIA	BAT165AX	6	1,92
Op-amp	Texas Instrument	595-OPA237UA	1	20,1
DC/DC omformer	Multicomp	MP-K7805T-500R3	2	33,8
RJ45	Hearting	09455511102	2	33,7
Level shifter	Texas Instruments	SNx4HCT245	1	13,36
Wago Terminal	WAGO	2604-3104	3	63,8
Encoder	Bourns	PEC11H-4225	1	29,57
IO expander	Microchip	MCP23016T-I/SO	1	22,87
Hovedkort	JLC PCB	Egendefinert	5	350
HMI kort	JLC PCB	Egendefinert	5	20
Totalpris				3834,31,-

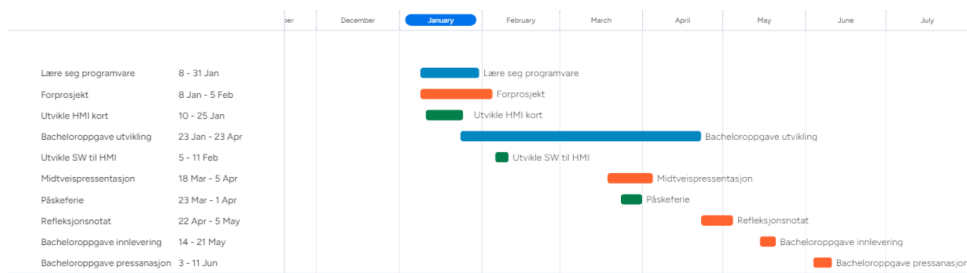


# Tillegg B

## Fremdriftsplan

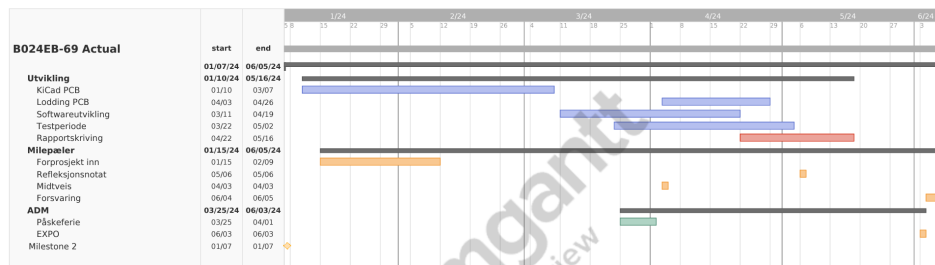
Vi valgte å benytte oss av gantt-diagram når vi skulle planlegge tidsbruken vår. Gantt-diagram gir oss en god og oversiktig oversikt over hvilke hovedoppgaver vi har forran oss, og hvor lang tid vi har tenkt å bruke på de ulike oppgavene. Under ser vi både estimert tidsbruk, og faktisk tidsbruk.

### B.1 Estimert fremdriftsplan



Figur B.1: Gantt skjema over estimert fremdrift

### B.2 Faktisk fremdrift



Figur B.2: Gantt skjema over faktisk fremdrift

## B.3 Logg for laboratoriarbeid

Tabell B.1: Logg for arbeid på laboratoriet

Logg dag:	Forklaring	Løsning:
4.april 2024	Oppstart lodding av kretskort. Kort introduksjon med veileder om hvordan vi skal lodde kortet. Fikk god kontroll på motstander og kondensatorer. Diverse chipper var mer krevende, men fikk kontroll der også etterhvert.	
10. april 2024	Da vi skulle lodde på level shifteren oppdaget vi at footprintet var feil, noe som gjorde at vi måtte lodde på ledninger.	
11.april 2024	Grunnleggende test av PWM fra mikrokontroller. Den fungerte fint. Vi målte også de to styresignalene til gate driveren. Styresignalet var firkantpulser med 4KH og 50% arbeidssyklus. Styresignalene fungerte og hadde 400 ns dødtid.	
24.april 2024	Hadde loddet ferdig hele kretskortet med tre faser, skulle teste at styringssignalene virket. MCU begynte å blinke med en gang vi koblet til 24V. Vi sjekket 24V->5V forsyning, denne virket som den skulle (ga ut 5V). Deretter testet vi å koble ut J6, da sluttet det å blinke og alt så ut til å virke helt fint. Vi lå merke til at spenning falt fra 24V til 19V når J6 ble koblet inn. Vi kunne nå vite at det enten er spenningsmålings kretsen eller levelhifteren det var feil med, da disse er de eneste tilkoblingene til J6. Vi koblet først ut levelshifteren, for å så legge inn J6 igjen. Da sluttet det å blinke og alt gikk som normalt, så vi kunne dermed vite at problemet innebar levelshifteren og dens oppkobling.	
25.april 2024	Fant ut at gårtdagens problemer skyldes strømbe- grensning på 24VDC forsyningen. Testet videre på L1, der det var alvorlige feil med gatedriveren grunnet feil under lodding.	Fjernet gatedriveren for L1 og fortsatte med test for L2. Den fungerte optimalt.
26.april 2024	Det ble bestemt at fordi L1 ikke funknet, skulle vi gå over til å lage en H-bro, og drive en DC-motor. Under testing av H-bro, ble det avdekket en feil på low-side på L3. Fortsatte med videre feilsøking.	En defekt kondensator med full kortslutning gjorde at ting ikke fungerte. Byttet den, og fikk deretter ting til å fungere.
29.april 2024	Testet med resistiv last på H-bro.	
30.april 2024	Test av unipolar styring	
1.Mai 2024	Sammenlikningstest av rippelspenning ved unipolar og bipolar styring.	
3.Mai 2024	Teste omformer på DC- maskin <sup>102</sup>	
13.Mai 2024	Teste omformer med overmodulasjon.	

# Tillegg C

## Forkortelser og programmer

### C.1 Forkortelser

**SPWM** Sinusoidal Pulse Width Modulation.

**SVPWM** Space Vector Pulse Width Modulation.

**AC** Alternating Current / Vekselstrøm.

**DC** Direct Current / Likestrøm.

**IGBT** Insulated-Gate Bipolar Transistor.

**MOSFET** Metal-Oxide-Semiconductor Field-Effect Transistor

**HMI** Human Machine Interface.

**VSI** Voltage Source Inverters.

**VSD** Variable Speed Drive.

**PCB** Printed Circuit Board / kretskort.

**ADC** Analog to Digital Converter.

**OC** Output Compare.

**TIMx** Timer x (intern timer i microcontroller).

**MCU** Microcontroller Unit.

**RMS** Root Mean Square.

**VSD** Variable Speed Drive

## C.2 Liste med programvare brukt i oppgaven

Tabell C.1: Programvare benyttet i oppgaven

Utvikler	Produkt	Formål
Kicad Developers Team	KiCad	PCB design
Time Base Technology Limited	Goodnotes	Grafikk
Microsoft	Excel	Management
Open source	InkScape	Grafikk
Microsoft	VS Code	MCU programmering
PlatformIO	PlatformIO	MCU programmering
STMMicroelectronics	STM32CubeIDE	MCU programmering
MathWorks	MATLAB	Simulering/grafer
TeamGantt	TeamGantt	Koordinering

# Tillegg D

## Liste med feil for PCB og software

### D.1 Liste med feil på PCB for hovedkort

- **Feil 1:** Bus-transceiver (SN54HCT245), som benyttes for å oversette 3.3V logikk fra MCU over til 5V logikk for gate drivere, passer ikke til footprinten i PCB. Under design av kretskort ble det antatt at SN54HCT245 var mindre enn hva den fysiske komponenten viste seg å være.

**LØSNING:** endre footprint for SN54HCT245 i KiCad til riktig størrelse.

- **Feil 2:** RST-pinnene på gatedrivere må gå via en 3.3V til 5V levelshifter, dette gjorde ikke på det nåværende designet.
- **Feil 3:** Mangler ohm-verdi på diverse motstander R50,51 37 38 OSV. Valgt 10KOhm, dette fungerte fint.

### D.2 Liste med feil på PCB for HMI-kort

- Feil 1: Mangelfull komponent-informasjon på silkscreen (mangler referanse designator)
- Feil 2: Mangler verdi på R4
- Feil 3: Running-LED mangler tilkobling til GPIO på kretskortet.

### D.3 Liste med feil i software

- Feil 1: Encoderen blir lest av på en dårlig måte, kan av og til minke telleren selv om vi vrir encoder mot høyre. Dette skjer fordi TIM3 av og til (men ikke ofte) begynner å telle nedover selv om en forventer at den skal telle oppover, vi vet ikke hvorfor..... feilen ligger mest sannsynlig i funksjonen `Bsp;nit_encoder;imer()` :”

- Feil 2: Bytte av faserekkefølge for å endre dreieretning på motor skjer momentant slik som koden er nå. For en motor i tomgang vil dette være ok, men i den virkelige verden går ikke dette. Ideelt må motoren bremse ned, endre fasefølge, og så akselerere opp igjen. Her er det mye forbedringspotensialer i koden.

# Tillegg E

## Kode

### E.1 MatLab-kode for å utføre FFT-analyse på måledata

```
1 datasett = readtable("
    SinglePhaseSPWM_Unipolar_f50_fsw1000_200V_R225_030524\
    SinglePhaseSPWM_Unipolar_f50_fsw1000_200V_R225_030524_2.csv", '
    ReadVariableNames', true, 'Decimal', ',');
2
3 t          = table2array(datasett(:,1));    % Tidsverdier
4 V          = table2array(datasett(:,3));    % y-verdier (
    spenning i dette tilfellet)
5
6 secondsPrTimeUnit = 1e-3;                % Hvilke
    tidsenhet er t oppgitt i? eks. millisekunder: 1e-3 s/TU
7
8 % Berenge samplingsfrekvens:
9 n_samples   = length(t);
10 n_TimeUnits = t(n_samples) - t(1);
11 T_sampling  = (n_TimeUnits*secondsPrTimeUnit) / n_samples;
12 f_sampling  = 1 / T_sampling;
13
14 % FFT-analyse og behandling av data:
15 Y          = fft(V);
16 f = (0:n_samples-1)*(f_sampling/n_samples);
17 P2 = abs(Y/n_samples);                    % To-sidig
    spekter
18 P1 = P2(1:n_samples/2+1);                 % Ensidig
    spekter
19 P1(2:end-1) = 2*P1(2:end-1);              % Dobler alle
    amplitudene unntatt DC og Nyquist
20 f = f(1:n_samples/2+1);                  % Reduserer
    frekvensakse til ensidig
21
```



```

22 % Velge topp N-komponenter
23 N = 30; % Antall
    komponenter du vil vise
24 [~, I] = sort(P1, 'descend'); % Sortere
    amplitudene i synkende rekkefølge og faa indeksene
25 significant_freqs = f(I(1:N));
26 significant_amps = P1(I(1:N));
27
28 % figure;
29 % bar(significant_freqs, significant_amps, 'blue');
30 % title('FFT-analyse');
31 % xlabel('Frekvens (Hz)');
32 % ylabel('Amplitude');
33
34 figure;
35 % Lag stolpediagram
36 hb = bar(significant_freqs, significant_amps, 'FaceColor', 'blue');
37
38 % Juster bredden paa stolpene
39 hb.BarWidth = 0.8;
40
41 % Legg til titler og akseetiketter
42 title('FFT-analyse');
43 xlabel('Frekvens (Hz)');
44 ylabel('Amplitude');
45
46
47 % Identifisere grunnharmonisk komponent
48 [~, maxIndex] = max(P1(2:end)); % Ignorerer DC-komponent
49 fundamentalIndex = maxIndex + 1; % Justerer for DC-forskyvning
50 fundamentalFreq = f(fundamentalIndex);
51 fundamentalAmp = P1(fundamentalIndex);
52
53 % Beregne THD:
54 harmonicsIndex = fundamentalIndex + (1:floor((n_samples/2)/
    fundamentalIndex)-1) * fundamentalIndex;
55 harmonicAmps = P1(harmonicsIndex);
56 THD = sqrt(sum(harmonicAmps.^2) / fundamentalAmp^2);
57
58 % Skrive ut THD
59 fprintf('THD of the signal: %.2f%%\n', THD * 100);

```

Listing E.1: MatLab kode for å utføre FFT-analyse

## E.2 MatLab-kode for å posserere samt plotte måledata fra picoscope

```
1 clc
2 clear
3
4 % Leser inn data fra Excel-ark + konvertere komma til punktum slik
   at MatLab
5 % kan tolke dataen:
6 dataset1 = readtable("H_Bridge_ResistiveLoadTest\Unipolar\300424\
   R125_f150_Rtest_300424\R125_f150_Rtest_300424_2.csv", '
   ReadVariableNames', true, 'Decimal', ',');
7 dataset2 = readtable("H-bro tester\
   H_Bridge_InductiveResistiveLoadTest\Unipolar\030524\
   R125_L40m_f150_UniPolar_030524\R125_L40m_f150_UniPolar_030524_2.
   csv", 'ReadVariableNames', true, 'Decimal', ',');
8
9 % Konvertere datasettene til selvstendige arrayer:
10
11 % Dataset 1:
12 t_dataset1      = table2array(dataset1(:,1)); % Tidsverdier
13 CH1_dataset1    = table2array(dataset1(:,2));
14 CH2_dataset1    = table2array(dataset1(:,3));
15 CH3_dataset1    = table2array(dataset1(:,4));
16 CH4_dataset1    = table2array(dataset1(:,5));
17 CH5_dataset1    = table2array(dataset1(:,6));
18
19 % Dataset 2:
20 t_dataset2      = table2array(dataset2(:,1)); % Tidsverdier
21 CH1_dataset2    = table2array(dataset2(:,2));
22 CH2_dataset2    = table2array(dataset2(:,3));
23 CH3_dataset2    = table2array(dataset2(:,4));
24 CH4_dataset2    = table2array(dataset2(:,5));
25 CH5_dataset2    = table2array(dataset2(:,6));
26
27 figure;
28 tiledlayout(4,2)
29
30 nexttile
31 plot(t_dataset1,CH2_dataset1,'r');
32 legend("Collector-emitter voltage V_{CE} on Ta+");
33 ylabel("Voltage [V]");
34 xlabel("Time [ms]");
35 xlim([-10 10]);
36 title("R-load");
37
38 nexttile
39 plot(t_dataset2,CH2_dataset2,'r');
40 legend("Collector-emitter voltage V_{CE} on Ta+");
```

```

41 ylabel("Voltage [V]");
42 xlabel("Time [ms]");
43 xlim([-10 10]);
44 title("RL-load");
45
46 nexttile
47 plot(t_dataset1,CH3_dataset1,'b');
48 legend("Collector-emitter voltage V_{CE} on Tb+");
49 ylabel("Voltage [V]");
50 xlabel("Time [ms]");
51 xlim([-10 10]);
52
53 nexttile
54 plot(t_dataset2,CH3_dataset2,'b');
55 legend("Collector-emitter voltage V_{CE} on Tb+");
56 ylabel("Voltage [V]");
57 xlabel("Time [ms]");
58 xlim([-10 10]);
59
60 nexttile
61 plot(t_dataset1,CH1_dataset1,'m');
62 xlabel("Time [ms]");
63 ylabel("Voltage [V]");
64 legend("Output voltage from H-bridge");
65 xlim([-10 10]);
66
67 nexttile
68 plot(t_dataset2,CH1_dataset2,'m');
69 xlabel("Time [ms]");
70 ylabel("Voltage [V]");
71 legend("Output voltage from H-bridge");
72 xlim([-10 10]);

```

## E.3 Kode for enfase SPWM

Dette vedlegget viser koden vi har utviklet for å kunne drifte omformereren som enfase vekselretter med både unipolar og bipolar SPWM. Dette er beskrevet i del kapittel 7.3.2.

```
1 void TIM2_IRQHandler(void)
2 {
3     uint32_t k_reload = LL_TIM_GetAutoReload(TIM1);
4     float      m_a = 1.25;
5     uint32_t sine_res = 101;
6
7     if(get_OutputVoltageType() == AC) // Single-phase DC to AC SPWM
8     {
9         static uint32_t i1 = 0;
10        static uint32_t i2 = 50;
11        compare_t comp;
12
13        switch (get_ModulationAlgorithm())
14        {
15            case BIPOLAR:
16                LL_TIM_OC_SetMode(TIM1, LL_TIM_CHANNEL_CH3 ,
17                LL_TIM_OCMode_PWM2);
18                LL_TIM_OC_SetMode(TIM1, LL_TIM_CHANNEL_CH3N ,
19                LL_TIM_OCMode_PWM2);
20                comp.T_a = (uint32_t)((m_a*k_reload*sine_table[i1] +
21                k_reload)/2.0);
22                comp.T_b = comp.T_a;
23                break;
24
25            case UNIPOLAR:
26                LL_TIM_OC_SetMode(TIM1, LL_TIM_CHANNEL_CH3 ,
27                LL_TIM_OCMode_PWM1);
28                LL_TIM_OC_SetMode(TIM1, LL_TIM_CHANNEL_CH3N ,
29                LL_TIM_OCMode_PWM1);
30                comp.T_a = (uint32_t)((m_a*k_reload*sine_table[i1]
31                + k_reload)/2.0);
32                comp.T_b = (uint32_t)((m_a*k_reload*sine_table[i2]
33                + k_reload)/2.0);
34                break;
35
36            default:
37                break;
38        }
39        LL_TIM_OC_SetCompareCH2(TIM1, comp.T_a);
40        LL_TIM_OC_SetCompareCH3(TIM1, comp.T_b);
41        i1++; i2++;
42        if (i1 >= sine_res) i1 = 1;
43        if (i2 >= sine_res) i2 = 1;
44    }
```

```
39     if (LL_TIM_IsActiveFlag_UPDATE(TIM2)) LL_TIM_ClearFlag_UPDATE(  
40         TIM2);  
}
```

## E.4 Kode for bipolar styring av fullbro DC/DC omformere

Dette vedlegget viser koden som ble utviklet for å kunne drifte omformeren som fullbro DC/DC omformer med bipolar styring, som er beskrevet i delkapittel 7.3.1.

```
1 void BipolarControl_DC(void)
2 {
3     // Inverter styresignal til CH3 i forhold til CH2
4     LL_TIM_OC_SetMode(TIM1, LL_TIM_CHANNEL_CH3, LL_TIM_OCMODE_PWM2)
5     ;
6     LL_TIM_OC_SetMode(TIM1, LL_TIM_CHANNEL_CH3N, LL_TIM_OCMODE_PWM2
7     );
8     float    dutyCycle = get_DutyCycle() / 100;
9     uint32_t k_reload  = LL_TIM_GetAutoReload(TIM1);
10            // Funksjon av svitsjefrekvens
11     uint32_t V_control = (uint32_t)round(k_reload * dutyCycle);
12     LL_TIM_OC_SetCompareCH2(TIM1, V_control);
13     LL_TIM_OC_SetCompareCH3(TIM1, V_control);
14 }
```

## E.5 Kode for unipolar styring av fullbro DC/DC omformer

Her vises koden som vi har utviklet for å kunne drifte omformeren som fullbro DC/DC omformer med unipolar styring. Dette er beskrevet i delkapittel 7.3.1.

```
1 void UnipolarControl_DC(void)
2 {
3     LL_TIM_OC_SetMode(TIM1, LL_TIM_CHANNEL_CH3, LL_TIM_OCMODE_PWM1)
4     ;
5     LL_TIM_OC_SetMode(TIM1, LL_TIM_CHANNEL_CH3N, LL_TIM_OCMODE_PWM1
6     );
7     uint32_t k_reload      = LL_TIM_GetAutoReload(TIM1);
8            // Funksjon av svitsjefrekvens
9     float    dutyCycle    = get_DutyCycle() / 100;
10
11     float    D_1          = 0.5 * (dutyCycle + 1);
12     float    D_2          = 1 - D_1;
13     uint32_t V_control_1  = (uint32_t)round(k_reload * D_1);
14     uint32_t V_control_2  = (uint32_t)round(k_reload * D_2);
15
16     LL_TIM_OC_SetCompareCH2(TIM1, V_control_1); // Bein A
17     LL_TIM_OC_SetCompareCH3(TIM1, V_control_2); // Bein B
18 }
```

## E.6 Kode for trefasedrift av omformer

```
1 void TIM2_IRQHandler(void)
2 {
3 // Parametre for PWM-algoritmer:
4 scheme_t algo = get_ModulationAlgorithm();
5 float m_a = get_AmplitudeModulationIndex();
6 float Vd = get_Vd();
7 uint32_t k_reload = LL_TIM_GetAutoReload(TIM1);
8 uint32_t sine_res = 101;
9 uint32_t counterLim = get_InterruptCounterLimit();
10 static uint32_t counter = 0;
11
12
13 if(counter >= counterLim)
14 {
15
16 static uint32_t i1 = 0;
17 static uint32_t i2 = 67;
18 static uint32_t i3 = 34;
19
20 compare_t compare;
21 float UAo1, UBo1, UCo1, Uk;
22
23 switch (algo)
24 {
25 case SPWM: // Sinus PWM
26 compare.L1 = (uint32_t)((m_a*k_reload*sine_table[i1] +
27 k_reload)/2.0);
28 compare.L2 = (uint32_t)((m_a*k_reload*sine_table[i2] +
29 k_reload)/2.0);
30 compare.L3 = (uint32_t)((m_a*k_reload*sine_table[i3] +
31 k_reload)/2.0);
32 break;
33
34 case SVPWM: // Space Vector PWM
35 UAo1 = m_a*(Vd/sqrt(3))*sine_table[i1];
36 UBo1 = m_a*(Vd/sqrt(3))*sine_table[i2];
37 UCo1 = m_a*(Vd/sqrt(3))*sine_table[i3];
38
39 Uk = (1/2.0)*(max(UAo1, UBo1, UCo1) + min(UAo1, UBo1,
40 UCo1));
41
42 compare.L1 = (uint32_t)(k_reload*((1/sqrt(3))*m_a*
43 sine_table[i1]-(Uk/Vd)) + k_reload/2.0);
44 compare.L2 = (uint32_t)(k_reload*((1/sqrt(3))*m_a*
45 sine_table[i2]-(Uk/Vd)) + k_reload/2.0);
46 compare.L3 = (uint32_t)(k_reload*((1/sqrt(3))*m_a*
47 sine_table[i3]-(Uk/Vd)) + k_reload/2.0);
```

```

42         break;
43
44     default:
45         break;
46     }
47
48     // Handle rotor direction
49     if (get_rotorDirection() == CCW)
50     {
51         uint32_t temp = compare.L2;
52         compare.L2    = compare.L3;
53         compare.L3    = temp;
54     }
55
56     // Set compare values:
57     LL_TIM_OC_SetCompareCH1(TIM1, compare.L1);
58     LL_TIM_OC_SetCompareCH2(TIM1, compare.L2);
59     LL_TIM_OC_SetCompareCH3(TIM1, compare.L3);
60
61     i1++; i2++; i3++;
62     if (i1 >= sine_res) i1 = 1;
63     if (i2 >= sine_res) i2 = 1;
64     if (i3 >= sine_res) i3 = 1;
65
66     counter = 0;
67 }
68
69 counter++;
70
71 if (LL_TIM_IsActiveFlag_UPDATE(TIM2)) LL_TIM_ClearFlag_UPDATE(
    TIM2);
72 }

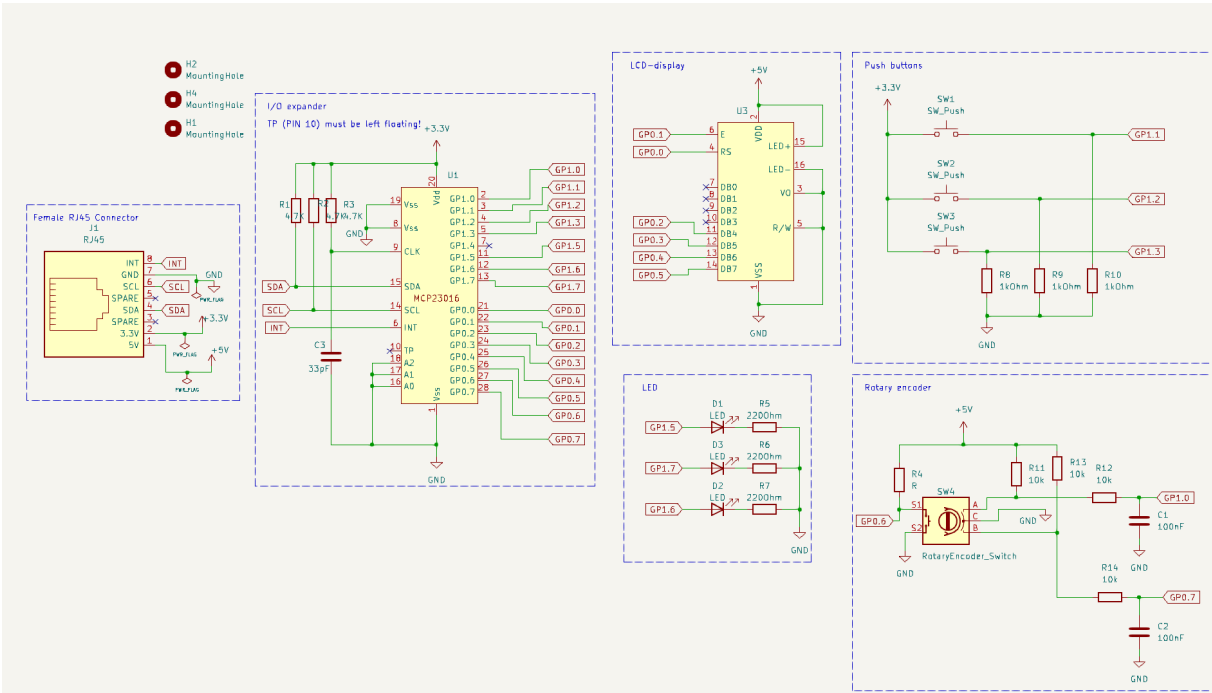
```



# Tillegg F

# Kretskortutlegg

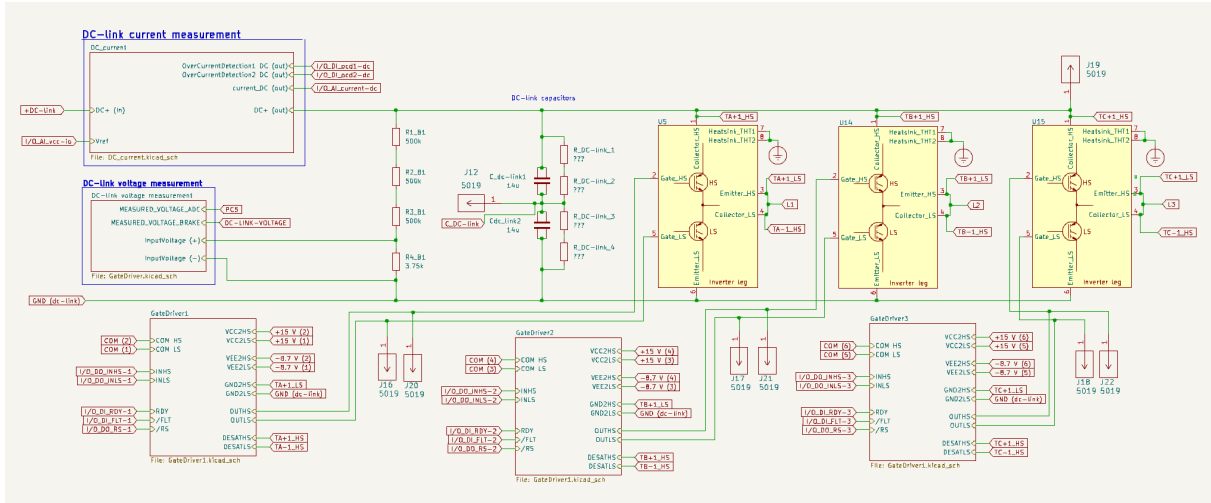
## F.1 HMI-kort



Figur F.1: Koblingskjema for HMI-kort

## F.2 Hovedkort

Figur F.2 viser koblingsskjema for 'hoveddelen' av kretskortet til omformeren. Det er langt flere koblingsskjema som tilhører hovedkortet. Ettersom at det er den tidligere gruppen som har utviklet koblingsskjemaene og ikke oss, så har vi valgt å ikke legge de som vedlegg i denne rapporten men heller referere til deres rapport [2].



Figur F.2: Koblingsskjema for hovedkort

## Tillegg G

### Måledata fra diverse tester

## G.1 Måledata fra test av spenningrippel ved unipolar og bipolar fullbro DC/DC

Tabell G.1 inneholder rådata fra test av avgitt rippelspenning ved unipolar og bipolar styring av fullbro DC/DC omformer. Testen er beskrevet i delkapittel 8.1.3.

Tabell G.1: Måling av spenningsrippel ved bipolar og unipolar styring ved ulike arbeids-  
syklus

$V_{out(RMS)}$ (Bipolar)	$V_{outsnitt}$ (Bipolar)	$V_{outRMS}$ (Unipolar)	$V_{outsnitt}$ (Unipolar)	Duty cycle
206	-186	46	4,2	5
206	-166	64	14	10
205	-145	78	24	15
205	-125	90	34	20
205	-104	102	45	25
205	-85	111	55	30
205	-64	120	65	35
204	-44	128	76	40
204	-23	135	86	45
204	-2	142	96	50
203	18	149	107	55
203	40	156	117	60
203	60	162	128	65
203	80	168	138	70
202	100	174	148	75
202	121	179	158	80
202	141	184	169	85
202	161	189	179	90
201	181	194	188	95

## G.2 Avgitt grunnharmonisk spenning ved overmodulasjon på enfase SPWM

Tabell G.2 inneholder data fra måling av RMS-verdi til avgitt grunnharmonisk spenning ved overmodulasjon. Testen er beskrevet i delkapittel 8.1.4.

Tabell G.2: Måling av avgitt grunnharmonisk spenning ved overmodulasjon

$m_a$	V
1.25	156.52
1.5	163.9
2.0	170.81
2.5	174.09
3.0	175.54
3.5	176.47
4	177.03
4.5	177.57
5	177.75
5.5	177.93

# Figurer

1.1	Ekvivalentskjema fremmedmagnetisert DC-maskin . . . . .	11
1.2	Asynkronmaskin[7] . . . . .	12
1.3	Avgitt dreiemoment fra asynkronmaskin som funksjon av sakking . . . . .	13
1.4	Børsteløst system . . . . .	14
2.1	Transistorer. [6] . . . . .	15
2.2	MOSFET symbol . . . . .	16
2.3	IGBT symbol . . . . .	17
2.4	Ekvivalentskjema for MOSFET . . . . .	18
3.1	Standard H-bro topologi . . . . .	19
3.2	Kontrollsystem for unipolar styring med $u_c = 0.6$ . . . . .	21
3.3	Simulering av unipolar styring med $u_c = 0.5$ og positiv avgitt spenning . . . . .	22
3.4	Simulering av unipolar styring med $u_c = 0.5$ og <b>negativ</b> avgitt spenning . . . . .	22
3.5	Styringslogikk for bipolar styring . . . . .	24
3.6	Simulering av bipolar styring med $u_c = 0.8$ og positiv avgitt spenning . . . . .	24
3.7	Simulering av bipolar styring med $u_c = -0.8$ og <b>Negativ</b> avgitt snittspenning . . . . .	25
4.1	Puls-bredde modulasjon (PWM) . . . . .	27
4.2	Generering av styringssignaler for trefaset SPWM med bipolar kontroll . . . . .	28
4.3	Overmodulasjon på enfase vekselretter med bipolar styring . . . . .	29
4.4	Trefase vekselretter topologi . . . . .	31
4.5	Vektordiagram for de 6 aktive vektorene ved SVPWM . . . . .	32
4.6	Kontrollsignal for SVPWM ved et gitt øyeblikk . . . . .	33
4.7	Romvektor i et gitt øyeblikk i et symmetrisk trefasesystem . . . . .	34
4.8	Simulerte styresignaler for SVPWM . . . . .	35
5.1	Blokkdiagram for Infineon 2ED020I12-F2 . . . . .	39
5.2	Maksimal strømføringsevne som funksjon av banebredde . . . . .	43
5.3	Blokkdiagram av styringskrets for transistorer . . . . .	44
5.4	Prototype av kretskort til omformer . . . . .	45
6.1	PCB-layout for HMI-kort . . . . .	50
6.2	3D modell av HMI-kort . . . . .	50
6.3	Sisste utkast av PCB-layout for hovedkort . . . . .	52
6.4	Prinsippskisse for PCB layout . . . . .	53
6.5	3D-modell av hovedkort . . . . .	54

6.6	Generert BOM via KiCad . . . . .	55
6.7	Bilde av omformerens bakside etter lodding . . . . .	55
6.8	Bilder av omformer sett ovenfra . . . . .	56
6.9	2023 utgave til venstre og 2024 utgave til høyre. . . . .	56
8.1	Spenning mellom gate og emitter for low side og high side transistorer . . .	66
8.2	Styresignaler for trefase SVPWM . . . . .	67
8.3	Styresignaler for trefase SPWM . . . . .	67
8.4	Styresignaler for bipolar kontroll av enfase SPWM . . . . .	68
8.5	Styresignaler for unipolar kontroll av enfase SPWM . . . . .	68
8.6	Styresignaler for bipolar styring av fullbro DC/DC omformer . . . . .	69
8.7	Styresignaler for unipolar styring av fullbro DC/DC omformer . . . . .	69
8.8	Beregningshastighet for SPWM-algoritme ved bruk av sineTable[] . . . . .	70
8.9	Beregningshastighet for SPWM-algoritme ved bruk av sin()-funksjonen fra <math.h>-biblioteket . . . . .	70
8.10	$R_{gate} = 100\Omega$ . . . . .	72
8.11	$R_{gate} = 50\Omega$ . . . . .	72
8.12	$R_{gate} = 24.9\Omega$ . . . . .	73
8.13	$R_{gate} = 4.99\Omega$ . . . . .	73
8.14	Varmeutvikling i IGBT som funksjon av tid ved $I_{last} = 2.66A$ og $R_{gate} = 10\Omega$	74
8.15	Kilde: Bilder tatt på kraftelektronikk lab . . . . .	74
8.16	Målinger ved unipolar svitsjing med induktiv og resistiv last . . . . .	76
8.17	Målinger ved bipolar svitsjing med induktiv og resistiv last . . . . .	77
8.18	Avgitt rippelspanning (RMS) som funksjon av arbeidssyklus . . . . .	78
8.19	Målinger fra test av H-bro som enfaset vekselretter . . . . .	79
8.20	FFT-analyse av <b>bipolar</b> svitsjing . . . . .	80
8.21	FFT-analyse av målinger fra testing med <b>unipolar</b> svitsjing . . . . .	80
8.22	Avgitt grunnharmonisk spenning som funksjon av $m_a$ ved overmodulasjon .	81
8.23	Ufiltrert avgitt spenning og filtrert laststrøm ved ulik $m_a$ . . . . .	82
8.24	Ufiltrert avgitt spenning og filtrert laststrøm ved ulik $m_a$ . . . . .	82
8.25	Målinger ved drift av DC-motor . . . . .	83
8.26	SIMULINK modell av H-bro med LC filter . . . . .	84
8.27	Praktisk oppkobling av testoppsett for H-bro med LC filter . . . . .	84
8.28	Merkeskilt DC-maskin . . . . .	85
8.29	Testoppsett ved drift av DC-Maskin . . . . .	85
9.1	Beregningshastighet for SVPWM-algoritme med sineTable[] . . . . .	87
9.2	Effekttap i IGBT ( $R_{gate} = 100\Omega$ ) . . . . .	89
9.3	<b>Simulering</b> av test i kapittel 8.1.4 . . . . .	91
9.4	<b>Simulert</b> FFT-analyse av test med <i>bipolar</i> styring i SimuLink . . . . .	92
9.5	<b>Simulert</b> FFT-analyse av test med <i>unipolar</i> styring i SimuLink . . . . .	92
9.6	Predikert kurve for $V_{o1}(m_a)$ . . . . .	93
B.1	Gantt skjema over estimert fremdrift . . . . .	100
B.2	Gantt skjema over faktisk fremdrift . . . . .	100
F.1	Koblingsskjema for HMI-kort . . . . .	116

F.2 Koblingskjema for hovedkort . . . . .	117
---	-----



# Bibliografi

- [1] *Advanced Electric Drives by Mohan Ned. 9781118485484. Innbundet - 2014 | Akademika.no.* URL: <https://www.akademika.no/teknologi/teknikk/advanced-electric-drives/9781118485484> (sjekket 06.05.2024).
- [2] Marius Midtbø Englund, Eirik Skorve Haugland og Ingrid Hovland. «3-phase DC/AC converter with SV-PWM». Accepted: 2023-11-30T14:13:28Z. Bachelor thesis. Høgskulen på Vestlandet, 2023. URL: <https://hvlopen.brage.unit.no/hvlopen-xmlui/handle/11250/3105460> (sjekket 24.01.2024).
- [3] Ned Mohan. *Power Electronics: A First Course*. 1st edition. Hoboken, N.J: Wiley, 18. okt. 2011. 288 s. ISBN: 978-1-118-07480-0.
- [4] *Om Høgskulen på Vestlandet*. 6. des. 2016. URL: <https://www.hvl.no/om/> (sjekket 24.01.2024).
- [5] *Power Electronics. 9780471226932. Innbundet - 2002 | Akademika.no.* URL: <https://www.akademika.no/teknologi/teknikk/power-electronics/9780471226932> (sjekket 28.01.2024).
- [6] *Power semiconductor device*. I: *Wikipedia*. Page Version ID: 1157290703. 27. mai 2023. URL: [https://en.wikipedia.org/w/index.php?title=Power\\_semiconductor\\_device&oldid=1157290703](https://en.wikipedia.org/w/index.php?title=Power_semiconductor_device&oldid=1157290703) (sjekket 16.01.2024).
- [7] Knut A. Rosvold. *asynkronmotor*. I: *Store norske leksikon*. 4. jan. 2024. URL: <https://snl.no/asynkronmotor> (sjekket 16.01.2024).
- [8] Lasse Sivertsen. *ELEKTRISKE MASKINER OPPBYGGING, VIRKEMÅTE OG DRIFT*. 2019, s. 1–202. ISBN: 978-82-450-3296-3.
- [9] Lasse Sivertsen. *ELEKTRISKE MASKINER, AC- OG DC- MASKINER*. 2023, s. 1–287. ISBN: 978-82-450-4604-5.
- [10] Andrzej M. Trzynadlowski. *Introduction to Modern Power Electronics*. Second Edition. Wiley India Pvt Ltd, 1. jan. 2011. 452 s. ISBN: 978-81-265-3380-0.