



Sjøkrigsskolen

Bacheloroppgave

Motordrive

– Undervisningsmodell –

av

Alf Inge Holm Fagerheim

Birger Engelsrud Andresen

Levert som en del av kravet til graden:

BACHELOR I MILITÆRE STUDIER MED FORDYPNING I
ELEKTRO OG AUTOMASJON

Innlevert: Mai 2018

Godkjent for offentlig publisering

Publiseringsavtale

En avtale om elektronisk publisering av bachelor/prosjektoppgave

Kadettene har opphavsrett til oppgaven, inkludert rettighetene til å publisere den.

Alle oppgaver som oppfyller kravene til publisering vil bli registrert og publisert i Bibsys Brage når kadettene har godkjent publisering.

Oppgaver som er graderte eller begrenset av en inngått avtale vil ikke bli publisert.

Vi gir herved Sjøkrigsskolen rett til å gjøre denne oppgaven tilgjengelig elektronisk, gratis og uten kostnader	<input checked="" type="checkbox"/> Ja	<input type="checkbox"/> Nei
Finnes det en avtale om forsinket eller kun intern publisering? (Utfyllende opplysninger må fylles ut)	<input type="checkbox"/> Ja	<input checked="" type="checkbox"/> Nei
Hvis ja: kan oppgaven publiseres elektronisk når embargoperioden utløper?	<input type="checkbox"/> Ja	<input type="checkbox"/> Nei

Plagiaterklæring

Vi erklærer herved at oppgaven er vårt eget arbeid og med bruk av riktig kildehenvisning

Vi har ikke nyttet annen hjelp enn det som er beskrevet i oppgaven.

Vi er klar over at brudd på dette vil føre til avvisning av oppgaven.

Dato: 25 – 05 – 2018

Birger Engelsrud Andresen
Kadett navn

Kadett, signatur

Alf Inge Holm Fagerheim
Kadett navn

Kadett, signatur

Forord

Bacheloroppgaven er skrevet av kadettene Alf Inge Holm Fagerheim og Birger Engelsrud Andresen fra linjen Marineingeniør Elektro og Automasjon, våren 2018.

Oppgaven er rettet mot de som har bakgrunn i elektro og er de som er interessert i effektiv motorstyring og læring rundt temaet. Spesielt er undervisningsmodellen vi har bygget ment for samspill mellom lærere og elever for å effektivisere læring rundt motorstyring. Hovedfokuset i læringen vi ønsker å tilby med modellen er faglig bredde. Derfor er fagtemaet tverrgående mellom motorlære, automasjon/programmering og hovedsakelig kraftelektronikk. Kraftelektronikken i modellen skaper mulighet for manipulering av form og hastighet på spenningen, mens programmeringen gjør det mulig å bestemme dets omfang. Medfølgende modellen er et oppgavehefte, simuleringsprogram, laptop og en AC-motor. AC-motoren er med slik at brukeren får erfart hvordan den vil fungere under de forholdene modellen skaper for motoren, ved hjelp av denne type kraftelektronikk. Hensikten med oppgaveheftet og simuleringsprogrammet er å understøtte undervisning med modellen.

Vi ønsker å takke vår veileder Rune Fosse for lån av utstyr i oppstartsfasen og faglig støtte gjennom hele perioden. Videre vil vi også takke førstemannuensis Alexander Sauter opp mot bruk og testing av utstyr vi har vurdert og implementert. Dette gjelder også Fredrik Haugsgjerd fra firmaet Wago som hjalp oss med en del testing av potensielt utstyr. Det har vært nødvendig å bruke en del måleinstrumenter, komponenter og spenningskilder under arbeidsperioden. For dette takker vi Arild Sæbø for bruk av hans lab. For evaluering og testing av det ferdige produktet, med gode tilbakemeldinger for revurdering, takker vi også Joakim Mork og Cato Solberg.

Bergen, Sjøkrigsskolen, 25-05-2018

Oppgaveformulering

EU har innført strenge krav til energieffektive løsninger. Dette gjelder også for elektriske motordrifter. En meget viktig komponent for å øke virkningsgraden i motordrifter er frekvensomformerer. Det er derfor viktig å forstå hvordan en frekvensomformer påvirker systemet den er satt i, helt fra inntak og ut til motoren som skal drive lasten vi ønsker å drive.

Elevene på Sjøkrigsskolen som studerer elektro og automasjon vil ende opp med å jobbe på fartøy i ulike størrelser med ulike effektbehov. Derfor søker denne bacheloroppgaven etter en måte å tilrettelegge for bred og effektiv læring av elektromotorens alternativer for drift og styring. Resultatet av dette er følgende problemstilling;

Det skal utarbeides en undervisningsmodell for elektriske motordrifter som kan brukes til demonstrasjoner og laboratorieundervisning.

Det skal utarbeides en undervisningsmodell for motorstyring ifra og med spenningskilde gjennom en type frekvensomformer til og med motor. Styring og overvåking skal foregå ved hjelp av mikrokontrollere. Modellen skal ha gode visuelle forhold med muligheter for ulike spenning- og signalmålinger, i den hensikt å kunne forstå dens virkemåte og effekt. Løsningen skal dimensjoneres, beregnes og bygges fysisk. Sammen med modellen skal det medfølge øvingsoppgaver og demonstrasjonseksempler som skal gi elevene flere dimensjoner av læring. I tillegg til dette skal et simuleringsprogram brukes til å knytte teori og praksis sammen for å få med flest mulige dimensjoner opp mot læring. Modellen skal enkelt kunne bli transportert og benyttet i alle rom med en fase 230V 50Hz vekselstrømuttak.

Sammendrag

Siden 2009 har EU satt krav til elektriske motorer og frekvensomformere med tanke på miljøvennlighet. Elektromotoren kan i dag drives på ulike måter. Den kan for eksempel kjøres på nominell frekvens og spenning kontinuerlig, spenningsreguleres med autotrafo, hastighetsreguleres med endring av aktiv motstand eller frekvens, eller ved bruk av en *variable speed drive* (VSD) som ofte får navnet frekvensomformer. Den mest effektive måten å styre elektriske motorer på i dag er ved bruk av frekvensomformere.

På bakgrunn av dette mener vi at forståelse og innsikt i effektiviserende tiltak for elektromotor-systemer er viktig for elever som skal jobbe på plattformer med disse type systemer. Interessen og relevansen bak dette har bidratt til valget av en type konstruksjonsoppgave basert på følgende problemstilling;

Det skal utarbeides en undervisningsmodell for elektriske motordrifter som kan brukes til demonstrasjoner og laboratorieundervisning.

Til å starte med ble det utarbeidet konsept, mål og design for en slik modell. Ut fra dette har materiell for drift, sikkerhet og overvåking blitt undersøkt. Etter bestilling av potensielt anvendelig utstyr, har det på bakgrunn av utprøving og testing blitt valgt ut hva som skulle brukes videre til konstruksjon.

Konstruksjon av modellen har hovedsakelig bestått av et kretskort for motorkontroll og to kort for styring, sammen med ledninger, knapper, sensorer og en skjerm. For å komplimentere modellen medfølger det også et krets-simuleringsprogram, motor og oppgavehefte for bruk i sammenheng med modellen. Oppgaveheftet inneholder blant annet teori som er relevant for kraftelektronikken i modellen og oppgaver av teoretisk, visuell og praktisk karakter.

Etter test av sluttproduktet ser vi at den har evnen til å kjøre en trefase AC-motor med varierende selvinnstilt frekvens, måle turtall og DC-buss-spenning, samt endre dreieretningen til motoren. I tillegg til dette tilbyr konstruksjonen visuelle, gode, og sikre forhold for signalmålinger. Med dette mener vi at den konstruerte modellen egner seg godt til demonstrasjoner og laboratorieundervisning med begrenset antall brukere på samme tid.

Innholdsfortegnelse

Figurer	1
Tabeller	4
Forkortelser	5
1 Innledning	6
1.1 Bakgrunn	6
1.2 Mål.....	7
1.3 Avgrensninger	8
1.4 Metode og struktur	8
1.4.1 V-modell.....	8
1.4.2 Fase én: Konseptutvikling	9
1.4.3 Fase to: Materiell	9
1.4.4 Fase tre: Software	10
1.4.5 Fase fire: Ferdigstilling av modell.....	11
1.4.6 Fase fem: Oppgaveutvikling.....	11
1.4.7 Fase seks: Oppgavetest.....	11
2 Teori.....	12
2.1 V-modell.....	12
2.1 Læring og kunnskap	12
2.1.1 Læring.....	12
2.1.2 Kunnskap.....	13
2.2 Frekvensomformer	14
2.2.1 Generelt	15
2.2.2 Strømforsyning	20
2.2.3 Likeretting	20
2.2.4 Filtrering – Glatting av signal.....	21
2.2.5 Vekselretting.....	22
2.2.6 Filtrering - Utbedring av signal til motor	22
2.3 Harmoniske signaler	24
2.4 Motorkontrollerkort: STEVAL-IHM023V3	25
2.5 Arduino.....	29
2.5.1 Generelt	29
2.5.2 Arduino-kort	29
2.5.3 Arduino IDE (programvare)	31
2.5.4 Hvordan programmere i Arduino IDE	31

2.6	OrCAD.....	32
3	Utviklet produkt.....	33
4	Utviklet software	38
4.1	Motorstyring	38
4.1.1	IGBT-styring med six step.....	38
4.1.2	Frekvensstyring.....	41
4.1.3	Motortilstander.....	41
4.2	Måling og overvåking	43
4.2.1	LCD-skjerm	43
4.2.2	Turtall på aksling (RPM)	43
4.2.3	Innstilt frekvens	45
4.2.4	Spenning på DC-bussen.....	45
5	Måleresultater	47
5.1	Gate-signaler.....	47
5.1.1	Måling for 50Hz, retning A	48
5.1.2	Måling for 50Hz, retning B.....	48
5.1.3	Måling for 20Hz, retning A	49
5.1.4	Måling for 20Hz, retning B.....	49
5.2	Strøm.....	50
5.2.1	Startstrøm (inrush)	50
5.2.2	Strømmåling ved endring av dreieretning.....	51
5.2.3	Strømmåling ved stopp	51
5.2.4	Kontinuerlig strømtrekk.....	52
5.3	Spenning	53
5.3.1	Fasespenning.....	53
5.3.2	Linjespenning.....	54
5.3.3	Common-mode-spenning.....	55
5.4	Effekter	56
5.5	Harmonisk støy	57
5.6	Temperatur	58
5.7	Hastighet på aksling.....	59
5.8	DOL-kjøring av motoren	60
6	Drøfting.....	63
6.1	Konsept	63
6.1.1	Komersiell frekvensomformer	63
6.1.2	Sammensetting av enkeltkomponenter	63

6.1.3	Modulbasert modell.....	64
6.2	Materiell	65
6.2.1	Valg av motor til modell.....	65
6.2.2	Motorkontrollerkort	67
6.2.3	Kontroll og styring	67
6.3	Software.....	69
6.3.1	Generelt	69
6.3.2	Six step-kode	69
6.3.3	Frekvensstyring	70
6.3.4	Motortilstander	71
6.3.5	Måling og overvåking.....	72
6.4	Ferdigstilling.....	74
6.5	Læringsutbytte	75
6.6	Målte resultater	76
6.6.1	Generelt	76
6.6.2	Gate-signaler.....	77
6.6.3	Strøm	78
6.6.4	Spenning	78
6.6.5	Effekter	79
6.6.6	Harmonisk støy.....	79
6.6.7	Temperatur	79
6.6.8	Hastighet på aksling	80
6.6.9	Sammenligning mellom DOL og modell	80
7	Konklusjon med anbefaling.....	81
	Referanser.....	82
	Vedlegg.....	86

Figurer

Figur 1 – Prosjektmodell (V-modell).....	8
Figur 2 – Konvensjonell frekvensomformer (VVSaktuell, 2013).....	14
Figur 3 - Prosess med frekvensomformer ut til AC-motor (Tech-flo, 2014)	14
Figur 4 - Kontroll-loop av en AC-drive med PWM-styrt frekvenskontroll (ABB Drives 2017a, 11).....	15
Figur 5 – Kontroll-loop av en PWM AC-drive med flux vector-styring (ABB Drives 2017a, 12).....	16
Figur 6 – Kontroll-loop av en AC-drive som bruker DTC (ABB 2017a, 13)	17
Figur 7 – Transistortønning i 180° six step-mode (Xplanator, 2017)	18
Figur 8 – DC-buss med transistorer ut til stjernekoblet last (Xplanator, 2017).....	19
Figur 9 - Motorens fase-spenning og strøm ved six step (Electrical Academia, 2018)	19
Figur 10 – Resultat av en likerettet sinus (Electronics, 2011).....	20
Figur 11 – Enfase likeretterbro (Wired, 2014)	20
Figur 12 – Glattet signal fra likeretterbro (Quora, 2016)	21
Figur 13 – Sinusfilter mellom drive og motor (ABB 2017c, 11)	23
Figur 14 - En ren sinusspenning og strøm inneholder ikke noe harmonisk forvrengning (ABB 2017b, 4).....	24
Figur 15 - Totale strømmen som resultat av den grunnharmoniske og 5. harmonisk forstyrrelse (ABB 2017b, 5).....	24
Figur 16 – STEVAL-IHM023V3 (ST 2014, 1)	25
Figur 17 – Inn- og utganger L6390 (ST 2018, 4)	26
Figur 18 – Skisse over tilførselskretsen i kortet (ST 2014, 11)	26
Figur 19 – Kablingsskjema for inngangssiden på L6390 (ST 2014, 13).....	27
Figur 20 – Kablingsskjema for utgangssiden på L6390 (ST 2014, 13).....	27
Figur 21 – NTC-motstand festet på kjøleribben til IGBTene (ST 2014, 23).....	28
Figur 22 – Arduino Uno mikrokontrollerkort (Wolfram, 2018).....	29
Figur 23 – Pull down-motstand (Arduino, 2018)	30
Figur 24 – Eksempelkode hentet fra Arduino IDE	31
Figur 25 – Bilde av ferdig produkt (koffert med utstyr).....	33
Figur 26 – Bilde av modellen med beskrivelse av punkter.....	34
Figur 27 – Bilde av modell (komponentoversikt).....	35
Figur 28 – Utklipp av simulert likeretterkrets	36
Figur 29 – Utklipp av utviklet likeretterkrets i OrCAD.....	36
Figur 30 – Utklipp av simulert vekselretterkrets	37
Figur 31 – Utklipp av vekselretterkrets i OrCAD.....	37
Figur 32 – Six step-sekvens med transistorer (Xplanator, 2017)	38
Figur 33 – Variablene som aktiverer transistorparene	39
Figur 34 – Utklipp av koden for styring av transistorpar	40

Figur 35 – Koden for hertzjustering	41
Figur 36 – Deklarering av knappene	41
Figur 37 – Øverste del av six step-koden med argument for å aktivere den	42
Figur 38 – Kodeloop for baklengs signalgivning for motsatt rotasjon	42
Figur 39 – Bilde av start/stopp-knapp, (til venstre), dreiretningsknapp og LED.....	42
Figur 40 – Bilde av LCD-skjermen som blir brukt til å vise resultater.....	43
Figur 41 – Utklipp av koden for flagg.....	44
Figur 42 – Utklipp av isr-funksjonen	44
Figur 43 – Utklipp av koden som initierer interrupt	44
Figur 44 – Bilde av telleskivemal	44
Figur 45 – Bilde av IR-sensor og telleskive	44
Figur 46 – Bilde av potmeteret som justerer innstilt frekvens	45
Figur 47 – Utklipp av koden for omgjøring til DC-buss-spenning	45
Figur 48 – Oversikt over oppkobling for måling av gate-signal.	47
Figur 49 – Måling av gate-signaler på oscilloskop (50Hz, retning A).....	48
Figur 50 - Måling av gate-signaler på oscilloskop (50Hz, retning B).....	48
Figur 51 - Måling av gate-signaler på oscilloskop (20Hz, retning A)	49
Figur 52 - Måling av gate-signaler på oscilloskop (20Hz, retning B).....	49
Figur 53 – Oversikt over oppkobling for måling av strømmer	50
Figur 54 – Resultat av inrush ved 50Hz (Skalering: 1A, 1s)	50
Figur 55 - Resultat ved endring av dreieretning to ganger ved 50Hz (Skalering: 1A, 1s)	51
Figur 56 – Resultat ved stopp, 50Hz (Skalering: 1A, 5s)	51
Figur 57 – Kontinuerlig strømtrekk ved 50Hz	52
Figur 58 - Kontinuerlig strømtrekk ved 35Hz.....	52
Figur 59 - Kontinuerlig strømtrekk ved 20Hz.....	52
Figur 60 – Fasespenning ved 50Hz	53
Figur 61 – Fasespenning ved 20Hz	53
Figur 62 - Linjepenning ved 50Hz	54
Figur 63 - Linjepenning ved 20Hz	54
Figur 64 – Common-mode-spenning ved 50Hz	55
Figur 65 - Common-mode-spenning ved 20Hz.....	55
Figur 66 - Effektmåling av motor ved 50Hz	56
Figur 67 - Strøm og spenning i forhold til hverandre ved 20Hz	56
Figur 68 – Strøm og spenning i forhold til hverandre ved 50Hz	56
Figur 69 – Harmonisk strøm ved 50Hz.	57
Figur 70 – Harmonisk spenning ved 50Hz.....	57
Figur 71 – Temperatur på motor (20Hz motordrift 5 min)	58
Figur 72 - Temperatur på motor (50Hz motordrift 5 min fra 48,7°C).....	58
Figur 73 - Måling av RPM ved 50Hz (IR-sensor)	59
Figur 74 - Måling av RPM ved 50Hz (Tachometer)	59
Figur 75 - Måling av RPM ved 20Hz (IR-sensor)	59

Figur 76 - Måling av RPM ved 20Hz (Tachometer)	59
Figur 77 - Måling av RPM ved 40Hz (Tachometer)	59
Figur 78 - Måling av RPM ved 50Hz (IR-sensor)	59
Figur 79 - Måling av RPM ved 30Hz (Tachometer)	59
Figur 80 - Måling av RPM ved 30Hz (IR-sensor)	59
Figur 81 - Oversikt over oppkobling for måling av DOL-kjøring.....	60
Figur 82 – Linjespenning på motoren DOL.....	61
Figur 83 – Fasepenning på motoren DOL	61
Figur 84 – Common-mode-spenning i DOL.....	62
Figur 85 – Harmonisk spenning i DOL	62
Figur 86 – Harmonisk strøm i DOL.....	62
Figur 87 – Effekter i DOL	62
Figur 88 – Volt/Amper/Hertz i DOL	62
Figur 89 - Motorskilt.....	66
Figur 90 - Motorkobling	66
Figur 91 - Kunnskapstrekant.....	75
Figur 92 – Sammenligning av gate-signaler ved dreieretning.....	77
Figur 93 – Sammenligning av høyest og lavest kontinuerlig strømtrekk (motordrift fra modell).....	78
Figur 94 – Effekter ved 50Hz	79
Figur 95 – Harmonisk strøm ved 50Hz	79

Tabeller

<i>Tabell 1 - Sannhetstabell for komparator L6390 (ST 2018, 14)</i>	<i>26</i>
<i>Tabell 2: Sannhetstabell for komparator L6390 (ST 2018, 14)</i>	<i>39</i>
<i>Tabell 3: Spenningsverdier og forholdstall.....</i>	<i>45</i>
<i>Tabell 4: Resultater for innstilt Hz mot opplevd Hz.....</i>	<i>73</i>

Forkortelser

A	Amper
AC	Alternate Current (Vekselstrøm)
DC	Direct Current (Likestrøm)
DOL	Direct On Line
FLIR	Forward Looking Infrared
HMI	Human Machine Interface
Hz	Hertz
I	Strøm
IGBT	Insulated Gate Bipolar Transistor
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
NTC	Negative Temperature Coefficient
PF	Power Factor
PWM	Pulse Width Modulation (Puls-bredde-modulasjon)
R	Resistanse
RMS	Root Mean Square
T	Transistor
TWh	Terawatttime
U	Spenning
VA	Voltamper
VAR	Voltamper reaktiv
VSD	Variable Speed Drive
W	Watt

1 Innledning

Innledningen beskriver bakgrunn for oppgaven, deretter målene vi selv har satt, og til slutt metoden som ble brukt for å tilstrebe oppnåelse av målene. Kapittel 2 inneholder teori som er nødvendig for å følge produktets utviklingsprosess, virkemåte, hensikt og medfølgende oppgavehefte. I kapittel 3 vil utviklet produkt bli presentert, og i kapittel 4 utviklet software. Målinger, tester og resultater rundt modellen vil bli presentert i kapittel 5. I kapittel 6 vil drøftingen av oppgaven finne sted. Kapittel 7 som er det siste kapittelet, inneholder konklusjon og videre anbefalinger for undervisningsmodellen. Oversikt over tabeller, figurer og forkortelser finnes innledningsvis, mens referanser og vedlegg finnes helt til slutt.

1.1 Bakgrunn

Elektriske motorer brukes i flere ulike redskap og systemer. For eksempel har pumper og ventilasjon lenge vært drevet med el-motorer på både sivile og militære fartøy. I følge konsulentfirmaet Propel sin tekniske rapport fra 2014 er det forventet at elektrisk fremdrift og hybridløsninger blir mer vanlig for flere skipstyper (Skipsutvikling, 2014). Nå som elektromotoren blir mer og mer implementert er det enda viktigere enn før å se på effektiviserende løsninger.

Siden 2009 har EU satt krav til disse elektriske motorene med tanke på miljøvennlighet. Rundt 70% av strømforbruket i industrien kommer fra elektriske motorer, og totalt samlet potensial for kostnadseffektiv forbedring har da vært på omlag 20-30%. Ved effektivisering kan dette gi energibesparelser på 135 TWh innen 2020 (NVE, 2016). Med dagens strømpris i Norge tilsvarer dette litt over 61,6 milliarder norske kroner.

Elektromotoren kan i dag drives på ulike måter. Den kan for eksempel kjøres på nominell frekvens og kontinuerlig spenning, spenningsreguleres med autotrafo, hastighetsreguleres med endring av aktiv motstand eller frekvens, eller ved bruk av en *variable speed drive* (VSD) som ofte får navnet frekvensomformer. Den mest effektive måten å styre elektriske motorer på i dag er ved bruk av frekvensomformere. Det har vært et lovmessig krav for IE2-motorer på 7,5-375kW i Europa, siden 2015, om å bli styrt med turtallsregulator (NVE, 2017)

Det er flere ulike metoder som blir brukt til forskjellige typer omformere for motordrift. Noen bruker enkel vekselrettingsteknikk med enkel overvåking, mens andre bruker svært

avanserte modeller til å utføre målinger og utregninger for å styre motoren på den måten brukeren ønsker. Noen parametere som blir målt og justert er spenning, frekvens, magnetisk fluks og moment.

Fordelene med disse er deres mulighet til å skreddersy drift og derav gjøre motorene mer energieffektive. Ulempene er deres evne til å skape harmoniske signaler på det elektriske nettet som kan skape blant annet forstyrrelser og varmegang i utstyr.

På bakgrunn av dette mener vi at forståelse og innsikt i elektromotor-effektiviserende systemer er viktig for elever som skal jobbe på plattformer med flere av disse type systemer. Interessen og relevansen bak dette har bidratt til valget av denne type konstruksjonsoppgave.

1.2 Mål

Opgavens mål har vært å produsere en undervisningsmodell av en type motordrive, bedre kjent som frekvensomformer, med tilhørende oppgavesett og simuleringsprogram som kan brukes til å tilby elever læring om hvordan en slik drive kan fungere; hvilke elementære komponenter de inneholder, dens egenskaper, fordeler og ulemper.

Praktiske mål har blant annet vært at den skulle ha mulighet for visuelt innblikk i komponenter, være lett å frakte og være funksjonell kun fra en standard enfase 230V stikkontakt. I tillegg skulle modellen være innkapslet for å gjøre den berøringssikker. Det var også ønskelig å ha med ulike sensorer for å måle parametere for å se på hvordan omgivelser, komponenter og programinnstillinger påvirker hverandre. For å få tilrettelagt enkle tiltak for læring med bakgrunn i praktisk arbeid har det også vært et mål å produsere et oppgavehefte med nødvendig teori og oppgaver som støttmateriell som medfølger modellen.

Oppsummering av konkrete krav til modellen:

- Gode visuelle forhold
- Lett å frakte
- Funksjonell med enfase stikkontakt
- Berøringssikker for strømførende deler
- Frekvensstyring
- Turtallsmåling
- Styring med mikrokontroller

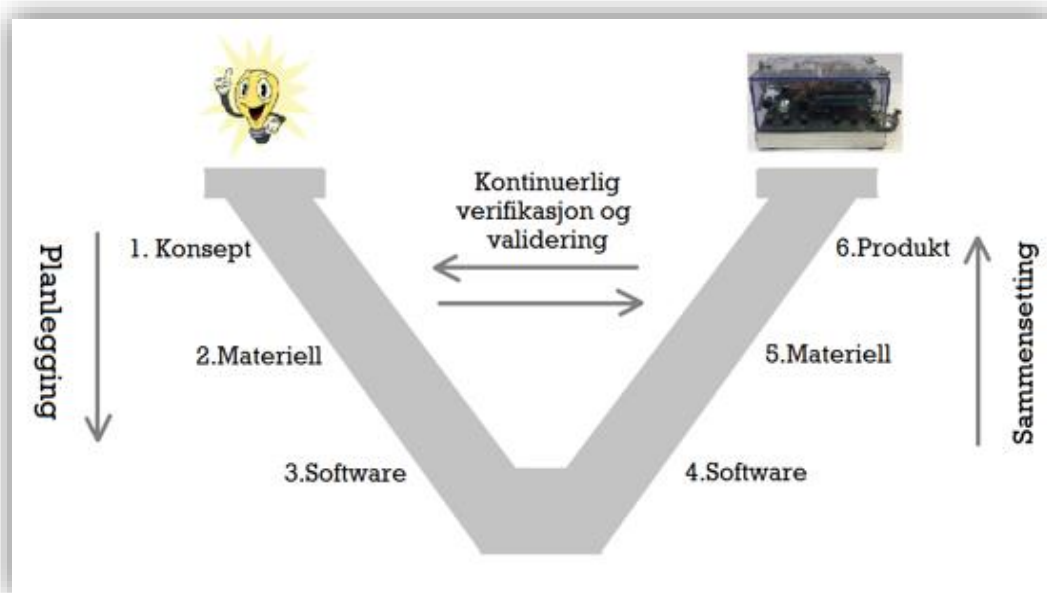
1.3 Avgrensninger

Til denne oppgaven har det blitt satt av et øvre budsjett på 20 000 kroner, og en tidsramme på fem måneder. Hovedfokuset i oppgaven ligger rundt de spesifikke kravene til modellen som var planlagt å konstruere.

1.4 Metode og struktur

Måten vi har jobbet med denne bacheloroppgaven på er ved å dele opp tidsperioden i ulike hovedfaser vi har jobbet ut ifra; Konseptfase, materiell, software, oppgaveutvikling, testing av oppgaver, og ferdigstilling. Disse fasene ble implementert i tidslinjalen vi jobbet ut ifra, som finnes i vedlegg D. I tillegg til dette har tankene rundt begynnelse og ferdigstilling av produktet vært basert på en prosjektmodell kalt V-modellen, som vil bli forklart nærmere i teoridelen av oppgaven.

1.4.1 V-modell



Figur 1 – Prosjektmodell (V-modell)

Innledningsvis stilte vi noen krav til hvordan modellen vår skulle se ut og hva som var hensikten med den. Det skulle være en lettfattelig og enkel design, som viste frem de ulike stegene i en VSD på en oversiktlig måte, for å tilrettelegge for best mulig læring.

Vi vil bruke V-modellen for å illustrere hvordan vi underveis i prosjektet hele tiden vurderte valgene våre ved å se tilbake på de kravene vi hadde stilt. Vi måtte også forsikre oss om at de ulike komponentene vi valgte oppfylte de samme kravene, og at de også var kompatible med hverandre. Sist men ikke minst så måtte vi også verifisere at det ferdige produktet tilfredstilte de viktigste kravene vi hadde satt oss i konseptfasen.

1.4.2 Fase én: Konseptutvikling

Innledningsvis diskuterte vi hvilken type oppnåelse som var ønskelig med en undervisningsmodell, og hva elever antageligvis ville satt pris på i en slik modell. Ut ifra dette ble det satt mål for hva som var ønskelig som sluttresultater for læring, sikkerhet, brukervennlighet og logistikk. Under konseptfasen ble det vurdert ulike måter å produsere en undervisningsmodell på.

Første vurdering var å bruke en kommersiell frekvensomformer ved å åpne den og spre delene utover et brett med overvåking av ulike parametere og styring av frekvensomformeren på vanlig vis via knappene på selve utsiden av omformeren eller ved tilhørende dataprogram for styring. Etter videre vurdering så det ut til at dette ville bli for enkelt og gitt modellen lite dynamikk med få muligheter for bred læring. Neste tanke var å kjøpe inn enkeltkomponenter for å bygge en helt enkel frekvensomformer bestående av et vekselstrøminntak, likeretter diodebro, et filter, og seks transistorer for vekselretting med selvvalgt frekvens ut til en motor. Ved senere vurdering ble det enighet om at det ville være for utfordrende og svært tidkrevende for brukere av modellen å foreta anskaffelse, plassering og rekonstruksjon av defekte enkeltkomponenter med et slikt oppsett.

Konklusjonen ble heller at det var ønskelig å gå for en middelvei ved å kjøpe inn større komponenter som allerede er satt sammen, som gir bedre fleksibilitet. Slik forsikret vi oss at levetiden til modellen lett kunne forlenges ved utilsiktet bruk. Dette betyr at det blir enkelt å erstatte komponentene hvis de blir ødelagt.

1.4.3 Fase to: Materiell

Valg av materiell for selve modellen ble det brukt lengst tid på å undersøke grunnet alle mulighetene som er på markedet. Materiell med store variasjoner i brukervennlighet, størrelse og kompleksitet ble undersøkt. Etterhvert ble det oppdaget et utviklerkort som kunne brukes til inntak, likeretting og vekselretting fra komponentfirmaet STMicroelectronics.

For styring av *insulated gate bipolar transistorene* (IGBT)¹ ble tre andre mikrokontrollerkort fra STMicroelectronics testet ut, men etter litt prøving ble enkle mikrokontrollere fra Arduino valgt til å implementeres i modellen.

Valg av motor belaget seg på fysisk størrelse og at vi ville ha en AC-motor, siden denne type motor er mest aktuelt i kraftelektronikk. En liten motor som allerede var på skolen som ble brukt til testing av modellen endte opp med å bli bestilt til å følge modellen.

For å konstruere selve modellen som brukervennlig og sikker ble alle ledende komponenter rammet inn i pleksiglass, slik at man fortsatt kunne se de. I skallet på pleksiglasset ble det montert start-, stopp-, dreieretningsbryter og potensiometer for styring av motorens tilstand og turtall. Trygge berøringssikre hunplugger hvor en enkelt kunne plugge i kabler for inntak til modell, uttak til motor, og testpunkter for analyse av det som skjer gjennom kretsene, ble valgt.

Det ble også kjøpt inn en liten laptop for å bruke med modellen for å gå inn på programmeringsplattformen for å se koden, i tillegg til simuleringsprogrammet for oppgaveløsning og simulering. For transport av modell, motor, laptop og kabler havnet valget på å benytte en koffert som enkelt kunne bæres alene.

1.4.4 Fase tre: Software

Som programvare var det viktig å benytte noe som var enkelt å forstå. Ideelt sett skulle det programmeres og skreddersys slik at fortløpende ideer kunne bli realisert. Hvorvidt det skulle benyttes et styringsprogram til modellen bygd fra bunn av, hente inn noe eksisterende å modifisere, eller kun bruke ferdigprogram, var under vurdering.

Styringskortene fra firmaet STMicroelectronics hadde flere alternativer til styringskort som brukte ulike programmeringsplattformer. De fleste programmeringsplattformene hadde eksempelkode og koder andre hadde laget klare til bruk. Kodene som ble funnet ville ikke passe AC-motorstyring, da de fleste kodene var egnet for DC-motorer. Dette gjorde at valget havnet på Arduino sine produkter med deres kodeplattform Arduino IDE (Integrated Development Environment)².

¹ En type transistor som kan tåle meget høye effekter

² Program for koding i Arduino

Det var to ulike vekselretting-metoder som ble vurdert å programmere; *six step*³-vekselretting og *pulse width modulation*-vekselretting (PWM)⁴. Målet var å kode begge typene, men det ble kun implementert six step. Slik ble resultatet fordi six step var en mindre kompleks styring å programmere enn PWM. I tillegg ble det ansett at six step ville være en god start for å tilegne seg grunnleggende forståelse for hvordan en drive kunne opereres, slik at det ville bli lettere for målgruppen å tilegne seg kunnskap om mer avanserte styremåter ved senere anledning.

1.4.5 Fase fire: Ferdigstilling av modell

For å verifisere at vi har bygget et ferdig produkt som oppfyller de kravene vi stilte innledningsvis, måtte vi selv kritisk vurdere vår motordrive-modell. Ved å foreta mange målinger og tester kunne vi få en formening om hvordan modellen presterte.

Siden det er lett for å bli inhabil valgte vi også å bruke en testperson til å validere det ferdige produktet opp imot de samme kravene.

1.4.6 Fase fem: Oppgaveutvikling

Det var viktig å få produsert et hefte med enkel teori, oppbygning, sikkerhet og brukerinformasjon for elevene, basert på designet og virkemåten til modellen og programmet. I tillegg til dette var det også ønskelig å produsere et sett med oppgaver som faglærer kunne bruke til å sette elever i gang med læring av praktiske oppgaver, tester og refleksjoner bak resultater. Sluttresultatet ble et kompakt hefte med både forklaringer og oppgaver slik at alt er samlet. Oppgavene ble basert på oppkobling og funksjonstest for å bli kjent med komponenter og virkemåte, signalmålinger, ulike spenningsmålinger, og effektmålinger.

1.4.7 Fase seks: Oppgavetest

Det var ønskelig å bruke en person i målgruppen til å lese heftet og utprøve oppgavene opp mot modellen. Hensikten var å få tilbakemelding på hvor eksplisitt innholdet måtte være, og hvorvidt innholdet og oppgavene ville øke elevens kunnskap innen fagene modellen omfavner. Testpersonen vurderte om oppgaveheftet var forståelig og gjennomførbart. Tilbakemeldingene fra personen var fokusert opp mot læring.

³ Six step er en metode man bruker til å vekselrette likespenning.

⁴ Puls Width Modulation er pulsbreddemodulasjon på norsk.

2 Teori

Siden dette er en oppgave om å planlegge og konstruere en undervisningsmodell er det spesielt viktig å få frem den underliggende teorien for hvordan vi har jobbet med å utvikle modellen, og hvilke forutsetninger for læring vi har lagt til grunn. Vi må også gi en grunnleggende forståelse for hva en frekvensomformer er, og hvilke utfordringer som medfølger. Samt utrede hvilke software og komponenter som har medgått i modellens konstruksjon og funksjon.

2.1 V-modell

V-modellen er en type modell som blant annet blir brukt i systemtenkning for et prosjekt. Den skal forsikre oss om at produktet som blir produsert er i henhold til kundens/brukerens krav, ønsker og behov. Under prosjekteringsfasen skal det forsørges om at konseptet og designen med alle undersystemer stemmer overens med de krav som er stilt tidligere. Om de ulike undersystemene som velges er kompatible med hverandre må også bli sjekket. Det kan være ekstremt kostbart å måtte endre på et produkt underveis i en byggeprosess, derfor er det veldig viktig med god planlegging og design i prosjekteringsfasen. Når prosjekteringen er ferdig og underveis i utviklingen, så må man hele tiden passe på at det som bygges stemmer overens med kravene som ble satt i designfasen. Tester kjøres underveis og helt til slutt for å verifisere at produktet leverer som bestilt (Bonnema, Venvliet, Broenink 2016, 18).

2.1 Læring og kunnskap

I dette delkapittelet vil teori om læring og kunnskap bli utredet, da det ansees som viktig å vite hvordan dette kan relateres til en utdanningsmodell som har til hensikt å tilby muligheter for læring og kunnskap.

2.1.1 Læring

I følge boken Læringspsykologi av Frode Svartdal og Magne Arve Flaten kan læring kjennetegnes som en relativt stabil og systematisk endring i atferd. En måte læring kan skje på er ved å få ny kunnskap, enten om noe, eller ved å lære en ny ferdighet (Læringspsykologi, 1998). I en artikkel fra 2009 om læringsstiler står det at Honey og Mumford

beskrev fire distinkt type elever; aktivister (lærer primært av erfaring), refleksjonister (lærer av reflekterende observasjon), teoretikere (lærer av å utforske foreninger og sammenhenger), og pragmatikere (lærer av å gjøre eller prøve praktiske ting med praktiske utfall).

Kort oppsummert var konklusjonen i artikkelen at det var begrenset forskning som korrelerte læringsstil til læringsutbytte, men at lærere bør konsentrere sin innsats i å undervise med en *multi-stil* som både når størst grad av elever og utfordrer alle til å utvikle seg (NCBI, 2009).

2.1.2 Kunnskap

Kunnskap kan deles i to hovedområder; reell kunnskap og intuitiv kunnskap. Med reell kunnskap menes det som er systematisk beskrevet i språk, det som kan meddeles eller diskuteres, altså begreper som er navngitt. Kilder for reell kunnskap kan være bøker.

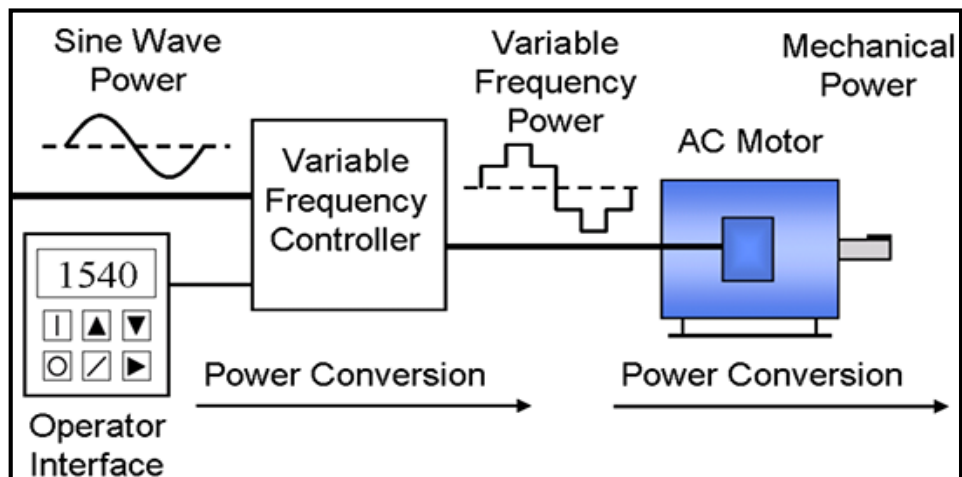
Ikke-språklig kunnskap er det intuitive man kan føle eller kjenne på kroppen, som det å lese/forstå når det er god eller dårlig stemning i et rom kommer inn i. Det kan ta lang tid før man opparbeider seg god intuitiv kunnskap om noe. To kilder for intuitiv kunnskap kan være erfaring og instinkt (Refleksjonsfilosofi, 2015).

2.2 Frekvensomformer

Her vil teori om frekvensomformere bli utredet, da dette er selve kjernen i oppgavens visjon.



Figur 2 – Konvensjonell frekvensomformer
(VVSaktuell, 2013)



Figur 3 - Prosess med frekvensomformer ut til AC-motor
(Tech-flo, 2014)

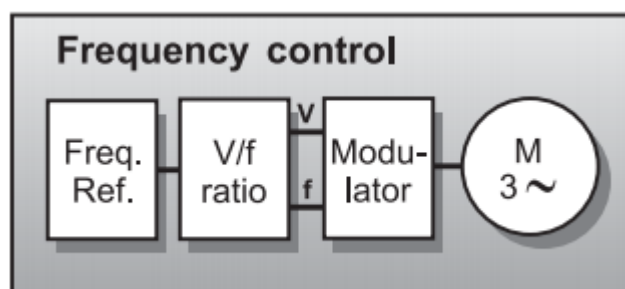
2.2.1 Generelt

Frekvensomformere, en type motordrive, kommer i ulike fysiske, parametriske og kapasitetsmessige størrelser med ulike styring- og justeringsmuligheter.

Hensikten med en motordrive er å kontrollere energien fra nettet og ut på motorens aksling. Det er hovedsakelig to komponenter vi ønsker å styre; hastighet og moment. I praksis så styres kun en av komponentene. Har du hastighetskontroll vil momentet styres av lasten, og ved momentkontroll er det farten som er bestemt av lasten. Til å begynne med så var det DC-motorer som ble brukt, fordi de har veldig godt moment og god fartsrespons uten å være avhengig av avansert elektronisk styring. Med tiden har det vært ønskelig å få tilsvarende karakteristikker på AC-motorer, siden de krever mindre vedlikehold og er billigere enn DC-motorene (ABB 2017a, 8).

Det finnes ulike prinsipper eller metoder for hvordan omformerne produserer og vedlikeholder ønsket spenningsnivå og frekvens ut til motor, og dermed moment. De forskjellige metodene har også ulike effekter, fordeler og ulemper. Under vil det nevnes noen av de mest vanlige typene gjennom tiden.

PWM-styringen i en AC-motorkontroller bruker spenning og frekvens som parametere til styring av motoren. Den har konstant fluks til nominell verdi ved å holde forholdet mellom spenningen og frekvensen konstant $U/F = \text{Konstant}$, ofte kalt skalarkontroll.

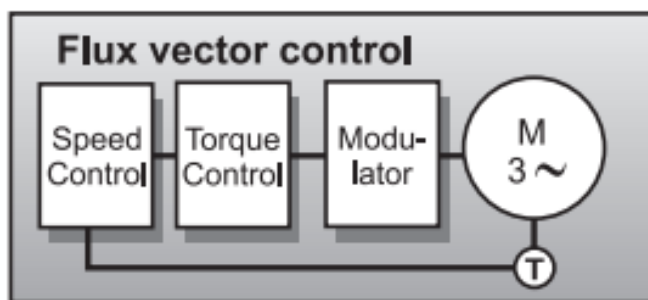


**Figur 4 - Kontroll-loop av en AC-drive med PWM-styrt frekvenskontroll
(ABB Drives 2017a, 11)**

Modulatorene mottar referanseverdier på spenning og frekvens, for så å lage et såkalt PWM pulstog som imiterer en AC-spenning som sendes ut på statorviklingene. Disse PWM signalene kan lages ved å sammenligne en sinusspenning og en trekantspenning gjennom en modulator. Det er en såkalt *open loop*-drive fordi den ikke har tilbakemelding på turtall fra akslingen. Driven bryr seg ikke om tilstanden til rotoren, det er lasten

på akslingen som dikterer momentet. Denne type drive egner seg best når det ikke stilles store krav til nøyaktighet og presisjon. Laster med kvadratisk moment slik som pumper og vifter vil egne seg godt til en slik type drive. (ABB Drives 2017a, 13)

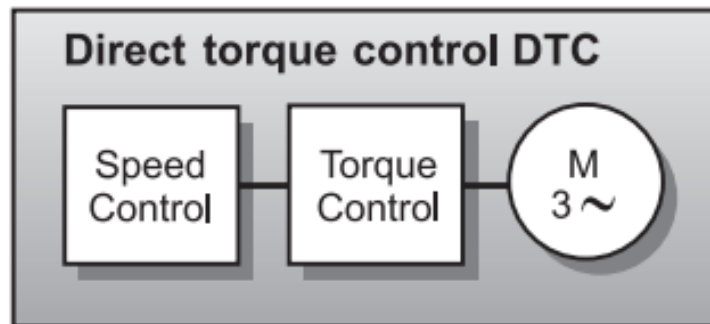
FOC (Field-Oriented Control) simulerer en DC drive, der motorens karakteristikk blir utregnet ved hjelp av mikroprosessorer. Det er en *closed loop*-drive som er avhengig av tilbakemelding fra motorens aksling.



Figur 5 – Kontroll-loop av en PWM AC-drive med flux vector-styring
(ABB Drives 2017a, 12)

For å oppnå *field orientation* må motoren først og fremst vite rotorens fart og vinkelposisjon i forhold til det statiske feltet i stator. Deretter blir de elektriske karakteristikk regnet ut ved hjelp av mikroprosessorer som simulerer en matematisk motormodell, som blir brukt til å prosessere dataene. Den varierer så spenning, strøm og frekvens for å oppnå riktig moment og turtall på akslingen. Momentet blir styrt indirekte. FOC-drives har god momentrespons og nøyaktig kontroll på akslingens fart. Den har fullt moment fra start og er tilnærmet like god som en DC-drive. FOC er dyrere og har en mer avansert styring enn U/F-drives (PWM). Dette skyldes blant annet fordi de krever en modulator og tilbakemelding, men til gjengjeldt så er det en mer nøyaktig drive med flere bruksområder. (ABB 2017a, 12)

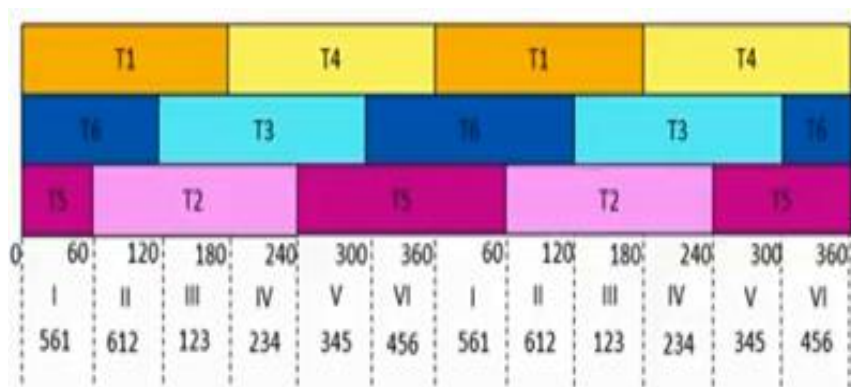
Direct torque control (DTC) er det nyeste innen drive teknologi. Det blir brukt i motor-drives for å kontrollere momentet, derav til slutt hastighet på AC-motoren. Måten dette skjer på er ved å kalkulere motorens magnetiske fluks og moment, basert på målt spenning og strøm, ved hjelp av avanserte raske matematiske beregninger. Metoden har høy effektivitet og lave tap, meget rask dynamisk momentrespons, og bidrar til lav støy fra motoren. Den oppnår field orientation uten modulator og tilbakemelding fra akslingen. Slik kutter den ut forsinkende mellomledd, og som følge av dette er momentresponsen ca. 10 ganger raskere enn noen annen AC- eller DC-drive. Den dynamiske fartsresponsen er ca. 8 ganger bedre enn noen annen open loop AC-drive.



Figur 6 – Kontroll-loop av en AC-drive som bruker DTC
(ABB 2017a, 13)

Six step-invertering er den styringen som blir brukt i produktet av denne oppgaven, altså undervisningsmodellen. Dette er også en måte å skape vekselstrøm fra likestrøm på, med mulighet til selvvalgt frekvens. Det er en lite effektiv metode, da det skapes veldig grove firkantpulser som har til hensikt å etterligne en sinusspenning.

Firkantpulsene er DC-spenning som slippes ut gjennom transistorer. Det finnes ulike metoder å sende signaler til transistorene på for å styre dem, som igjen gir ulike spenningsresultater ut i andre enden. Metoden går ut på å sende signaler til transistorene i en gitt rekkefølge for å få strømmen til å oppføre seg som en AC-strøm gjennom lasten. Det er to forskjellige operasjonsmoder som kan benyttes; 120° operasjonsmode, og 180° operasjonsmode. Med 120°-mode vil hver transistor være på i 120° i løpet av gitt periode, mens med 180°-mode vil de være på i 180° hver (Xplanator, 2017). Her vil 180°-moden bli forklart.



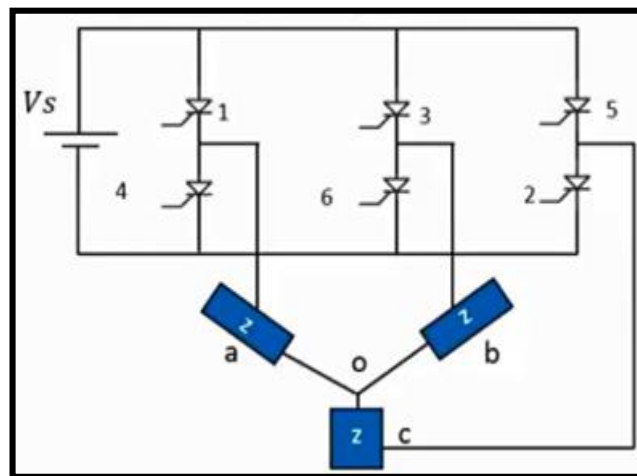
Figur 7 – Transistortiltenning i 180° six step-mode

(Xplanator, 2017)

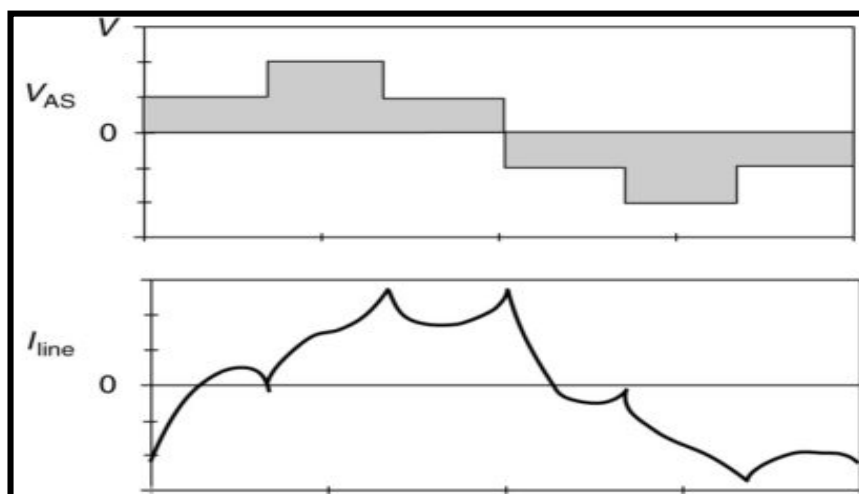
Kretsen opererer i seks trinn, hvor hvert trinn består av 60°. Hver transistor blir trigget hver 60°, og holder seg på/høy i 180° til den går av/lav igjen.

Figur 7 viser at transistor T1 starter ved 0° og varer til 180°, mens T2 tenner fra 60° og holder 180° til den når 240°-merket. Slik fortsetter det for alle transistorene. Under trinn 1 (0-60°), er T5, T6 og T1 tent. Under trinn 2 (60-120°), er T6, T1, og T2 tent osv.

Avhengig av hvilke transistorer som er tent, vil spenning over lastene **a**, **b** og **c** være positiv eller negativ. Resultatet av dette er vekselstrøm.



Figur 8 – DC-buss med transistorer ut til stjernekoblet last
(Xplanator, 2017)



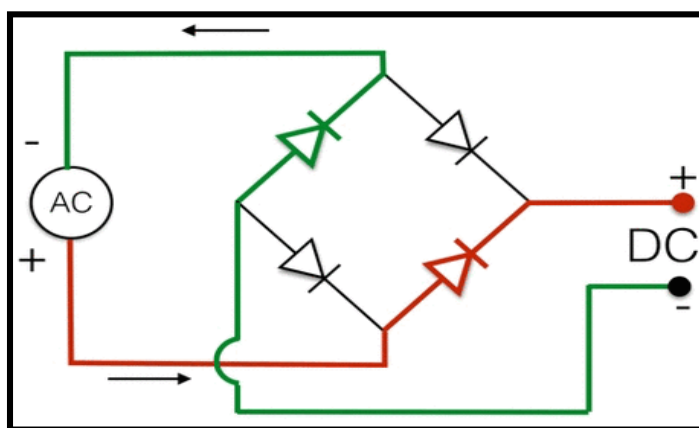
Figur 9 - Motorens fase-spenning og strøm ved six step
(Electrical Academia, 2018)

2.2.2 Strømforsyning

Forsyningen til de fleste frekvensomformere brukt i EU vil være på en- eller/og trefase 230V og trefase 400V 50/60Hz. De er oftest direktekoblede, på skip og off-shore brukes som regel trefase 690V.

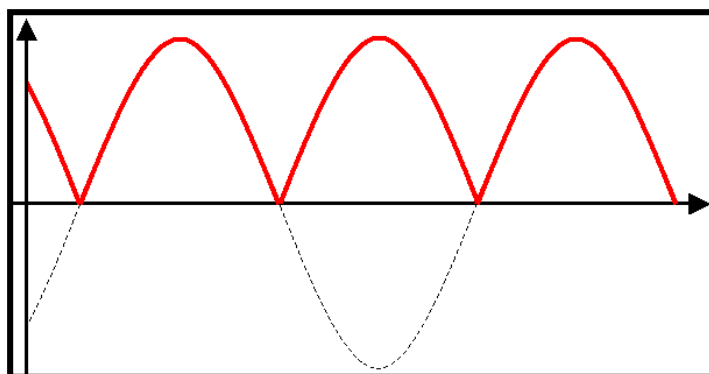
2.2.3 Likeretning

Med likeretning menes omgjøring av vekselspenning til likespenning. Dette gjøres ved hjelp av kraftelektronikk. Ofte skjer likeretning ved bruk av dioder, også kalt strømventiler, siden de kun tillater strøm til å gå i én retning ved normale tilstander i forhold til komponentkarakteristikken. Den fysiske ordningen av dioder for likeretning kalles ofte for likeretterbro eller diodebro.



Figur 11 – Enfase likeretterbro
(Wired, 2014)

Rett etter likerettingen av vekselstrømmen har funnet sted vil ikke spenningsnivået være konstant slik vi anser likestrøm å være fra andre kilder som for eksempel batterier. I stedet vil det se ut som bølger som svinger opp og ned, fra 0V til en viss positiv verdi.



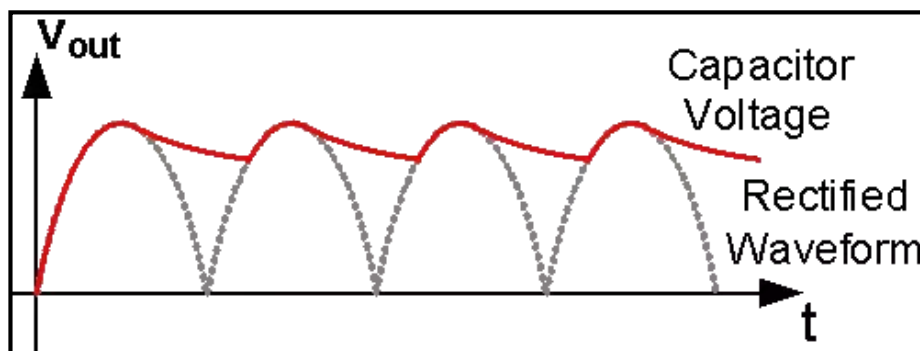
Figur 10 – Resultat av en likerettet sinus
(Electronics, 2011)

Grunnen til dette er fordi likeretterbroen har invertert den negative delen av spennings sinusbølge, fordi broen kun slipper gjennom bølgene som går til høyre, ifølge figur 11.

Utstyr som spenningen blir brukt på vil fortsatt kunne operere som om den får likespenning, på grunn av signalets *root mean square*-verdi (RMS⁵-verdi), men lite optimalt. Det optimale er å få glattet ut signalet slik at det holder seg mest mulig stabilt. Dette kan gjøres ved bruk av filter, normalt en kondensator.

2.2.4 Filtreiring – Glatting av signal

Ved å bruke filter rett etter den bølgede likespenningen, vil signalet kunne holde seg relativt stabilt. Det ønskelige er å redusere *rippelen*⁶ i bølgene mest mulig. Dette kan gjøres ved bruk av kondensatorer, som kan fungere tilnærmet slik batterier fungerer. Kondensatoren har egenskapen til å motsette seg raske endringer i spenninger, og til å lagre energi. Dette utnyttes til å glatte spenningssignalet. Ved bruk av riktige verdier på kondensatorene opp mot inn-signalets spenningsnivå og frekvens kan ut-signalet bli bortimot helt glatt uten svingninger.



Figur 12 – Glatte signal fra likeretterbro
(Quora, 2016)

Den glattede spenningen er 1.35 ganger større enn RMS tilførselspenning ved bruk av en-fase diodelikeretterbro (ABB 2011, 17). Signalet befinner seg nå i den delen av omformerens som kalles *DC-bussen*⁷ og går nå videre ut i frekvensomformerens krets for videre behandling.

⁵ Root Mean Square er det kvadratiske gjennomsnittet, også kalt effektivverdi i elektro. Det er denne verdien det som oftest implisitt snakkes om, og den som måles med voltmeter.

⁶ Dalene i spenningssignalet, fra minst verdi til størst verdi. Oppgis i prosent.

⁷ Hovedstrømskretsen i frekvensomformerens som består av likespenning

2.2.5 Vekselretting

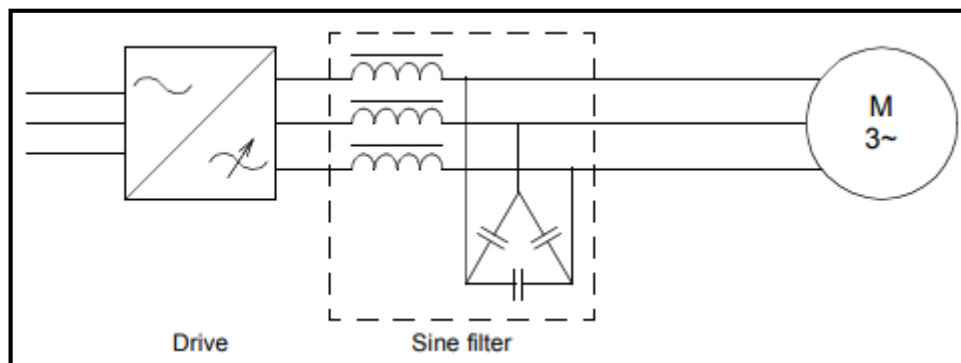
Vekselretting, det motsatte av likeretting, skjer ved at likespenningen fra DC-bussen blir sendt ut gjennom transistorer som blir styrt av og på av styresignaler. Både strøm- og spenningsignaler er vanlig, men IGBT bruker spenningsignaler. Disse styresignalene kan ha en fast eller varierende aktiveringssyklus ettersom hvordan hyppigheten til transistorenes av- og på-funksjon er ønsket. Rekkefølgen og hastigheten transistorene jobber med, samt spenningen ut og til lasten, bestemmer hvordan signalet vil se ut - hvor fint det er, og hvor raskt det svinger, altså frekvensen. Jo nærmere signalet ser ut som en glatt sinus, jo bedre. Enkle vekselrettere kan levere firkantpulser, mens andre kan levere jevnere mer sinuslignende signaler.

2.2.6 Filtrering - Utbedring av signal til motor

For å tilstrebe best mulig resultat av motordrifter, kan det være svært nyttig å bruke filtre. Det finnes ulike type filtre, noen mer effektive og dyrere enn andre. Ved prioritering av filter kommer det ofte an på hvilke utfordringer systemet har. Filtrene kan «rengjøre» signalene som skal ut til motoren for uønskede frekvenser og annen «sykdom» i signalet, typisk høyfrekvente harmoniske signaler og reflekterende spenningsignaler som kan bli så store at de ødelegger isolasjon. Disse filtrene monteres mellom driven og motoren. Under vil det nevnes noen typer filtre som er vanlig for AC motordrifter.

Sinusfilter eller sinusdrossel er et lavpassfilter som undertrykker de høyfrekvente komponentene ut fra omformeren. Det er spesielt viktig å gjøre dette ved ekstra lange kabelstrekke fram til motoren fra omformeren. Sinusfilteret:

- minimerer spenningsrefleksjoner, lagerstrømmer og harmoniske forstyrrelser.
- reduserer akustisk støy fra motoren i de fleste tilfeller.
- skaper et spenningsfall på typisk 10-15% ved nominell drift, slik at maks mulig motormoment synker (ABB 2017c, 12)



Figur 13 – Sinusfilter mellom drive og motor
(ABB 2017c, 11)

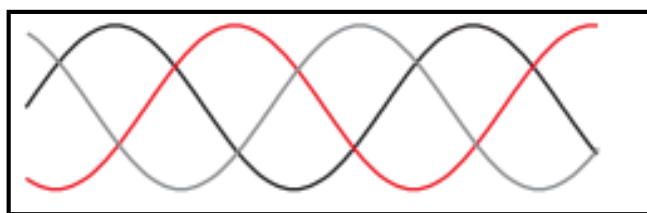
Du/dt-filter er også en type lavpassfilter som består av spoler og kondensatorer. Dets hensikt er å beskytte motoren mot høye du/dt -verdier, altså bratte/raske endringer i spenningen. Filteret er ment for systemer med kortere kabelstrek (<150 m) enn systemer hvor sinusfilter er anbefalt (150-300m). Du/dt-filteret:

- øker livsløpet til motoren ved å fjerne signaler som tærer på isolasjon.
- har mindre vekt, størrelse og pris enn sinusfilteret.
- eliminerer ikke akustisk støy fra motoren (Danfoss 2015, 7).

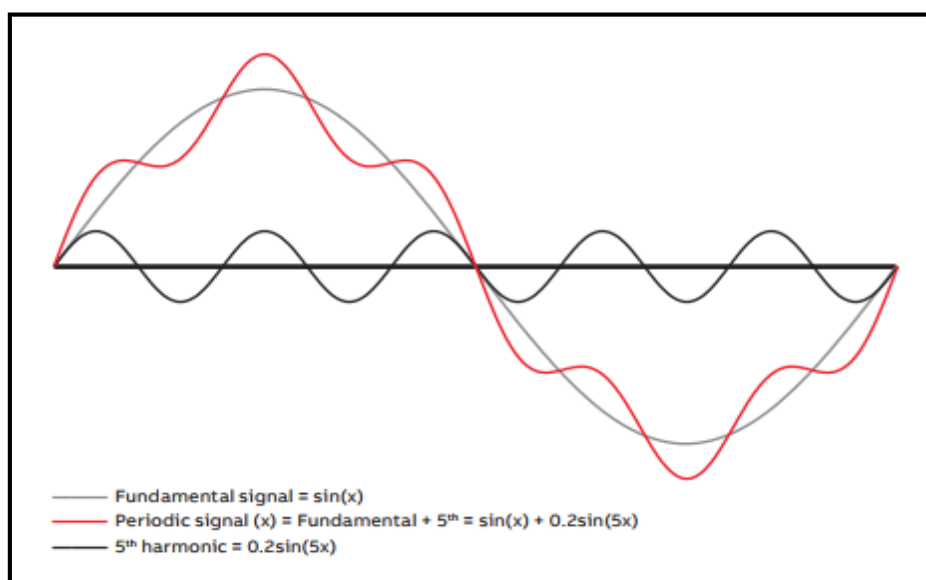
Common-mode-filter er et lavpassfilter med nanokrystallinske kjerner som reduserer høyfrekvent støy i motorkabelen. Hovedsakelig blir de brukt til å redusere motorens lagerstrømmer som kan genereres fra omformerens raskt vekslende spenningspulser med høy frekvens. Lagerstrømmene kan skape elektriske utladninger og etse motor og lagre. Slike strømmer blir også referert til som *common-mode*-strømmer (Danfoss, 2010).

2.3 Harmoniske signaler

Det som er negativt med frekvensomformere er at de er med på å produsere harmoniske signaler. Harmoniske signaler er grunnfrekvensen multiplisert med heltall, som kan være spenning og strøm i et elektrisk system. Hvis et signal inneholder en eller flere harmoniske signaler av grunnfrekvensen kalles det for en forstyrrelse, noe som ikke er ønskelig. Summen av alle harmoniske forstyrrelser kalles *Total Harmonic Distortion* (THD), og oppgis i prosent (Thorsen 2000, 80). Forstyrrelsene vil blant annet kunne føre til oppheiting i apparater og ledere, målefeil og lugging i motormoment. Reduksjon av dette er derfor ansett som ønskelig. Det er de ikke-lineære lastene vi finner i elektroniske apparater som vil resultere i dette, typisk halvlederkomponenter som manipulerer tilførselen fra nettet. Konkrete eksempler på apparater som vil føre til disse type signaler er typisk datamaskiner og printere, *light emitting diode*-lys (LED⁸-lys), batteriladere, og frekvensomformere (ABB 2017b, 4).



Figur 15 - En ren sinusspenning og strøm inneholder ikke noe harmonisk forvrengning (ABB 2017b, 4)



Figur 14 - Totale strømmen som resultat av den grunnharmoniske og 5. harmonisk forstyrrelse

⁸ En diode som sender ut fotoner (lys).

2.4 Motorkontrollerkort: STEVAL-IHM023V3

Kortet valgt for implementering i modell er STEVAL-IHM023V3. Det brukes til å lage VSDer for motorer opp til 1kW. Med evalueringskort menes det et kretskort man har mange muligheter til å koble og måle på selv for å oppnå det man måtte ønske istedenfor at det er like lukket som kortene i kommersielle komplette frekvensomformere. Hensikten med evalueringskort er å bli kjent med teknologi og utvikle kommersielle løsninger. Kretskortet inneholder flere komponenter som er med på å tilby ulike omgjøringer og filtreringer av strøm- og spenningsignaler fra inntaket, gjennom kretsene og ut til og med motoren. Som inntak brukes enfase (90-285VAC eller 125-400VDC), og som uttak til motor kan det velges mellom en- og trefase.

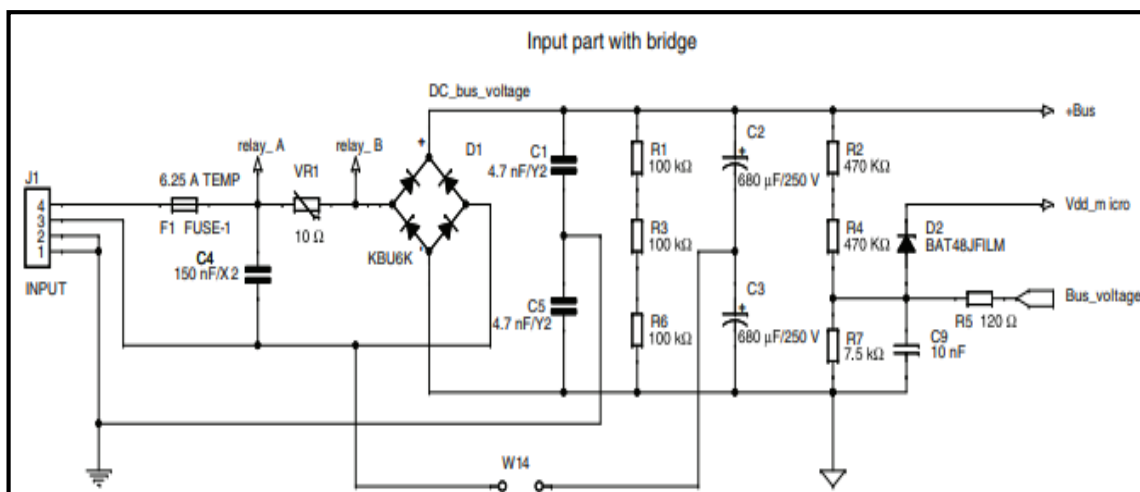


Figur 16 – STEVAL-IHM023V3

(ST 2014, 1)

Selv om det gis inntrykk av at det er veldig åpent og fritt fram med bruk og drift av kortet, er det innlagte beskyttelser for overstrøm, overtemperatur og anti-kortslutnings-forrigling mellom transistorene. Det fungerer slik at *gate*-signalene ut til de seks IGBTene går gjennom komparatorer⁹ (L6390) som passer på at kombinasjoner av IGBTer som kan føre til kortslutning blir umulig. Kortet har en god del *pins* man lett kan komme til for å gi signaler fra en kontroller. Det består av mange komponenter, men de vi anser som mest relevante er de som direkte bidrar til likeretting, filtrering, vekselretting og sikkerhet.

⁹ En komparator sammenligner signaler og gir output på bakgrunn av sammenligningens resultat.



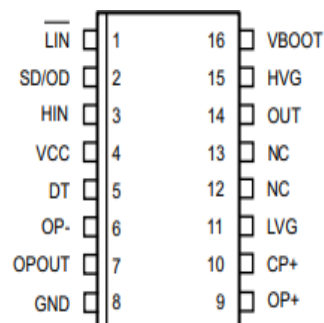
Figur 18 – Skisse over tilførselskretsen i kortet
(ST 2014, 11)

Figur 18 viser hvordan hovedkretsen for likeretting er bygget opp fra **J1**-koblingspunktet, hvor man kobler til driftsspenningen og beskyttelsesjord, og ut til DC-bussen gjennom likeretterdiodene/diodebroen. Kondensatorene i kretsen fungerer som filter og reduserer rippelen¹⁰ til DC-spenningen. Sikring **F1** på 6.25A bidrar til sikkerhet ved overstrøm. **VR1** er en *negative temperature coefficient*-motstand (NTC¹¹-motstand) som har til hensikt å eliminere input *peak*¹² av strøm når kondensatorene lades opp. For mer effektivitet er det også mulig å forbykoble denne etter oppstartsfasen. Komparatorne, som passer på

Tabell 1 - Sannhetstabell for komparator L6390
(ST 2018, 14)

Input			Output	
\overline{SD}	\overline{LIN}	HIN	LVG	HVG
L	X ⁽¹⁾	X ⁽¹⁾	L	L
H	H	L	L	L
H	L	H	L	L
H	L	L	H	L
H	H	H	L	H

1. X: don't care.



Figur 17 – Inn- og utganger
L6390 (ST 2018, 4)

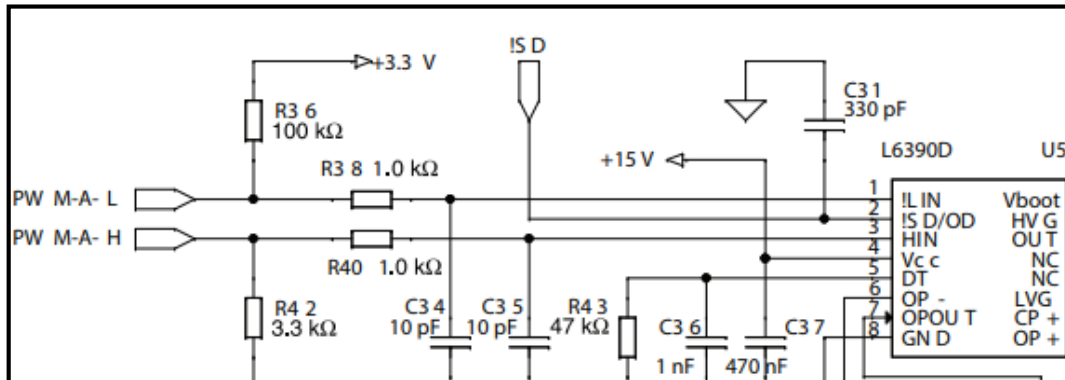
¹⁰ Forskjellen mellom de høyeste og laveste kontinuerlige vekslende verdiene.

¹¹ Når temperaturen øker, vil motstanden gå ned.

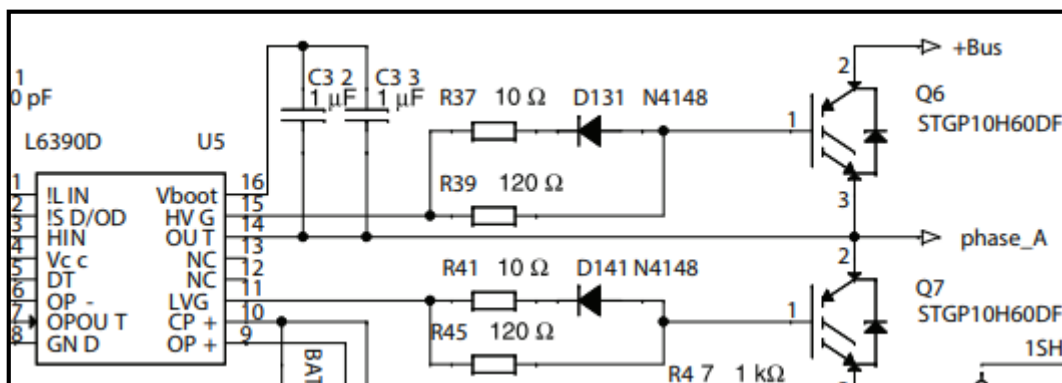
¹² Amplitudeverdi/Toppverdi til en bølge.

at kortslutning mellom transistorene ikke forekommer, heter L6390 og har koblingspunktene som vises i figur 17.

Tabell 1 viser når utgangene vil være høye eller lave, avhengig av inngangene. **Smart shutdown** (SD) er en inngang som legger til rette for umiddelbar sperring for utgangene. **Low-side driver input** (LIN) og **high-side driver input** (HIN) trenger spenningsignal (+3.3 til +5V) for å tenne output **high-side driver output** (HVG) og **low-side driver output** (LVG), vist i figur 19. +15V fra kortets interne krets forsyner komparatoren på input **Vcc**.



Figur 19 – Koblingsskjema for inngangssiden på L6390
(ST 2014, 13)



Figur 20 – Koblingsskjema for utgangssiden på L6390
(ST 2014, 13)

LVG og HVG-utgangene vil gi ut signaler (+15V) til transistorene, vist i figur 20. Komparatorene er det tre av i kortet, en til hvert transistorpar. I figur 19/20 ser vi en av dem.

I dette kortet er IGBTene i hovedstrømskretsen kalt **Q6**, **Q7**, **Q8**, **Q9**, **Q10** og **Q11**.

Q6 og Q7 er et slags transistorpar siden den ene komparatoren styrer begge disse.

Det samme gjelder for Q8 og Q9, samt Q10 og Q11. Tilsammen er det tre komparatorer.

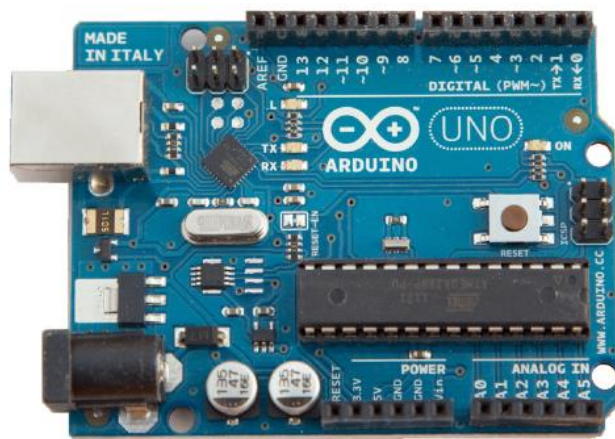
For å redusere sannsynlighet for overoppheting på transistorene er disse festet på en kjøleribbe. For å hindre at transistorene blir overopphetet og ødelagt er kjøleribben utstyrt med en NTC-motstand som vil gi signal til komparatoren om å stoppe når temperaturen blir for høy (70°C) (ST 2014, 23).



**Figur 21 – NTC-motstand festet på kjøleribben til IGBTene
(ST 2014, 23)**

2.5 Arduino

I dette delkapitlet vil det bli forklart hva Arduino er, hva som kan gjøres med det, og hvordan man kan få det til med det programmeringsspråket Arduino forstår.



Figur 22 – Arduino Uno mikrokontrollerkort
(Wolfram, 2018)

2.5.1 Generelt

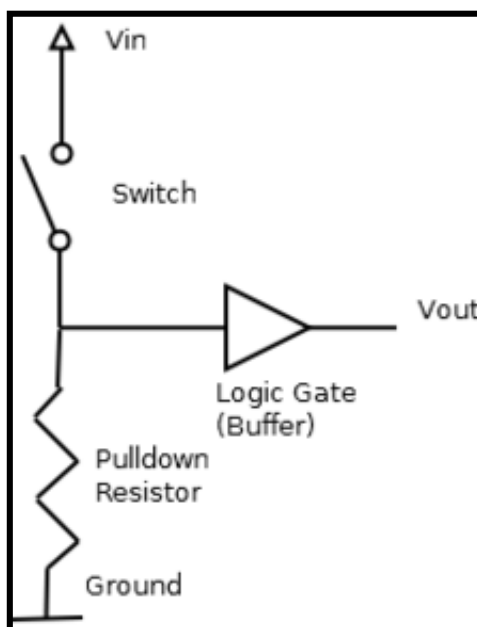
Arduino er en plattform brukt til å lage prosjekter av elektronikk, og belager seg på åpen kilde. Plattformen består av små fysiske programmerbare kretskort, et programmeringsspråk med et nettbasert utviklingsmiljø, og sensorer og aktuatorer for å bygge grensesnitt mot den fysiske verden (NTNU, 2017). Noe som er litt mer kjent for elektro-utdannede er *programmerbar logisk styring* (PLS). Man kan si at Arduino kan brukes til å realisere samme typer operasjoner som en PLS gjør. Forskjellen er noe ulike kapasiteter som for eksempel hastighet, spenningsnivå, brukervennlighet, antall innganger og utganger, samt friheten til å velge disse selv.

2.5.2 Arduino-kort

Kretskortene er som små datamaskiner med noe ulike innganger og utganger i både antall og funksjon. Inn- og utgangene på kortene kan sammenlignes med et menneske, hvor inngangene kan være sansene, og utgangene kan være signaler vi sender til musklene for å utføre noe fysisk. Disse inn- og utgangene kan være både analoge og digitale. De digitale vil enten være lav/av ved 0V eller høy/på ved 3.3 til 5V avhengig av kortet. Analoge innganger derimot, vil kunne lese en spenning mellom 0 og 5V og gjøre det om til et tall mellom 0 og 1023 som kan leses og brukes i programmet man laster opp til

mikrokontrolleren, og motsatt med utgangene. Det skal nevnes at disse egentlig ikke er rene analoge inn- og utganger, men bruker PWM til å realisere de ulike spenningene fra 0 til 5V RMS (Arduino, 2015).

Skal man koble til utstyr som brytere, transistorer og releer er det viktig å bruke *pull-down*-motstand i kretsen. I utgangspunktet vil ikke utgangenens potensiale gå helt tilbake til 0V etter den har vært 5V. Den vil ligge å “flyte”¹³ så lenge den ikke er tilkoblet referansejord eller 5V. Motstanden brukes som en veldig høy motstand til jord slik at potensialet går til referansejord 0V når kretsen til komponentene ikke er lukket.

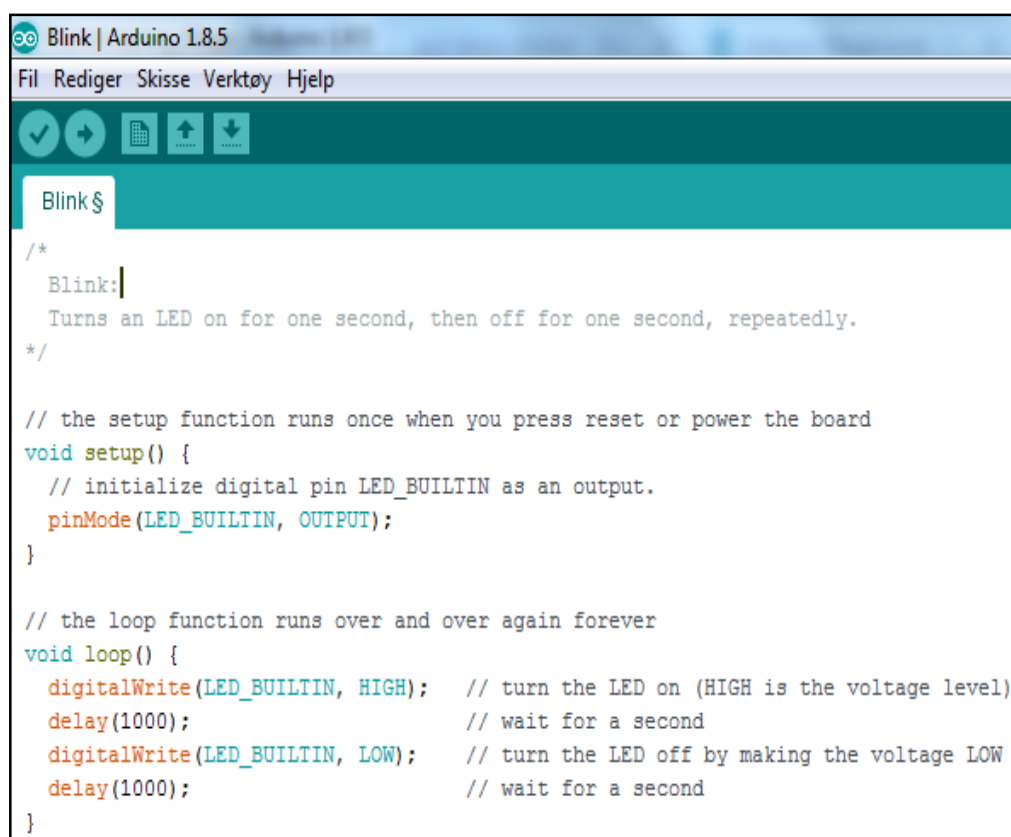


Figur 23 – Pull down-motstand
(Arduino, 2018)

¹³ Potensialet har ikke en konstant tilstand på 0 V eller spenningen som definerer digital 1 (TRUE).

2.5.3 Arduino IDE (programvare)

Arduino IDE er utviklingsmiljøet hvor man kan skrive koder for å laste opp til mikrokontrolleren på kretskortet. Programmeringsspråket er en forenklet versjon av C++¹⁴, slik at det er lett å lære. I motsetning til de fleste PLS-programmer, kan det kun programmeres i strukturert tekst uten alternativ for noen grafisk programmeringsmetode. Overføringen av koden som er skrevet lastes opp til kortet ved bruk av en USB-kabel fra PC. Nedenfor er et eksempel på en kildekode i programmet.

The image shows a screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.8.5". The menu bar includes "Fil", "Rediger", "Skisse", "Verktøy", and "Hjelp". Below the menu bar is a toolbar with icons for a checkmark, a right arrow, a grid, an upload arrow, and a download arrow. The main editor area shows the code for the "Blink" program. The code is as follows:

```
/*
  Blink:
  Turns an LED on for one second, then off for one second, repeatedly.
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Figur 24 – Eksempelkode hentet fra Arduino IDE

2.5.4 Hvordan programmeres i Arduino IDE

Hvert Arduino-program har to hovedfunksjoner. Disse er deler av et dataprogram som kjører spesifikke kommandoer. De har unike navn og blir *kalt opp*¹⁵ når de trengs. Hovedfunksjonene kalles for *setup()* og *loop()*. Deklarering av disse er nødvendig for å for-

¹⁴ C++ er et kjent programmeringsspråk.

¹⁵ Bedt om å utføre kodingen omfavnet med { }.

telle Arduino hva som skal skje i dem. Etter funksjonens deklarerer skrives det to krøllparenteser, en som betyr start { , og en som betyr slutt }. Koden som skrives innenfor disse parentesene vil bli utført når funksjonen blir kalt opp.

Oppretting av variabler eller konstanter du skal jobbe med i kodingen gjøres øverst, før `setup()`. For eksempel: `int x = 5;`¹⁶. `Setup()` vil kun kjøre én gang, når kortet starter opp. Her er det vanlig å definere hvilke pinner på kortet som skal være innganger og hvilke som skal være utganger. For eksempel: `pinMode(3, OUTPUT);`¹⁷. Etter `setup()` er ferdig vil `loop()` starte. Denne funksjonen vil kjøre kodene innenfor dens parenteser kontinuerlig. For eksempel `led = HIGH;`¹⁸.

Det finnes flere funksjoner og mange argumenter som kan utgjøre en kildekode. Ovennevnte er kun det helt grunnleggende med enkle eksempler (Fitzgerald 2013, 36).

2.6 OrCAD

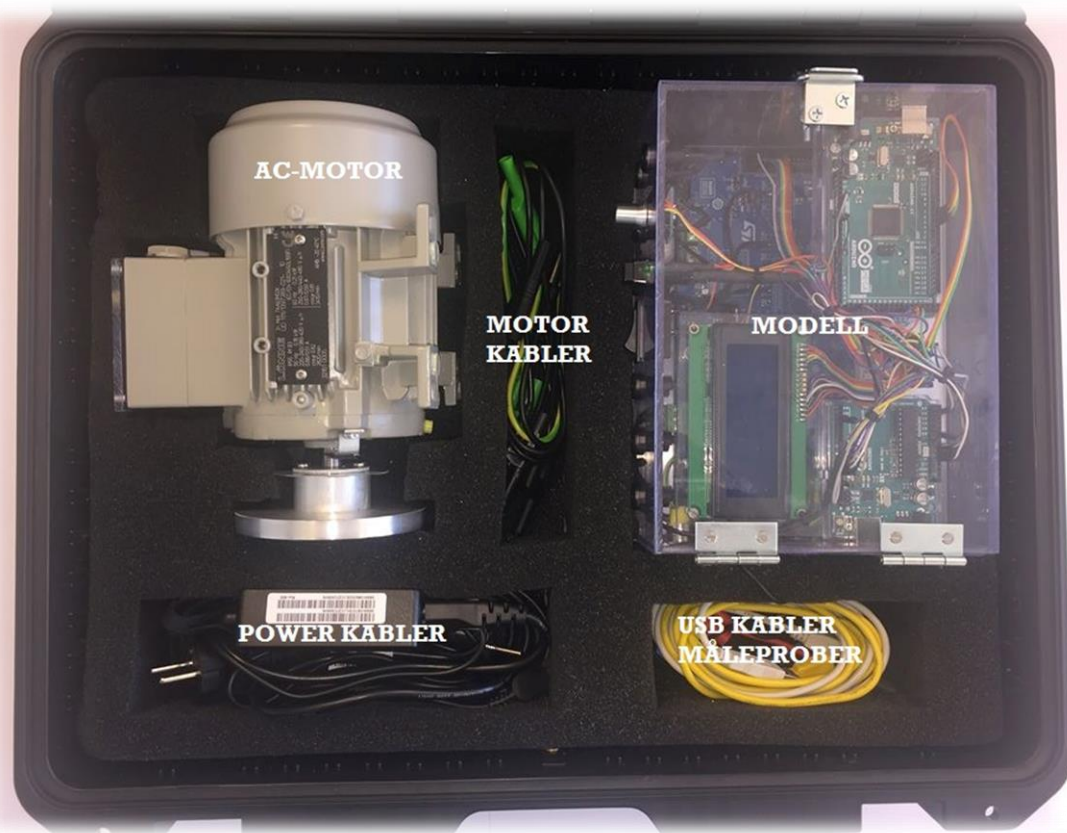
OrCAD er et dataprogram for tegning og simulering av elektriske kretser. Med dette programmet er det mulig å sette inn utallige komponenter på et skjema, definere verdier på deres parametere, koble de opp mot hverandre og kjøre beregninger ved hjelp av simulering. For å beregne og måle ønskede punkter i kretsen settes digitale måleprober ut over ønskede målepunkter. Etter simuleringen vil grafer med resultatet komme opp, hvor flere innebygde matematiske funksjoner gjør det mulig å finne ut av det meste om hvordan kretsen ville fungert i virkeligheten (OrCAD, 2008).

¹⁶ Variabelen “x” er lik tallet 5. Datatypen `int` (integer) gjør at variabelen kun kan være et heltall.

¹⁷ Pin nr. 3 på kortet skal være en utgang.

¹⁸ Variabelen `led` settes til `HIGH`. Er variabelen en utgang vil den gi 5 volt ut av pinnen - til for eksempel en LED.

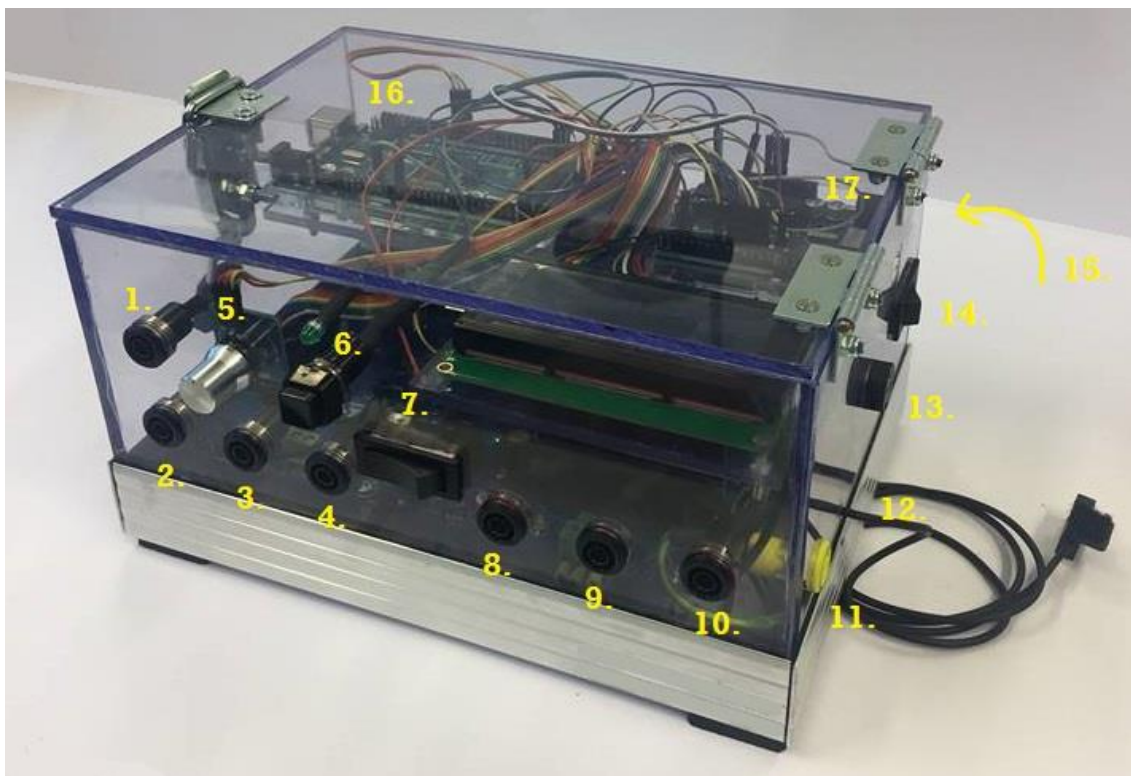
3 Utviklet produkt



Figur 25 – Bilde av ferdig produkt (koffert med utstyr)

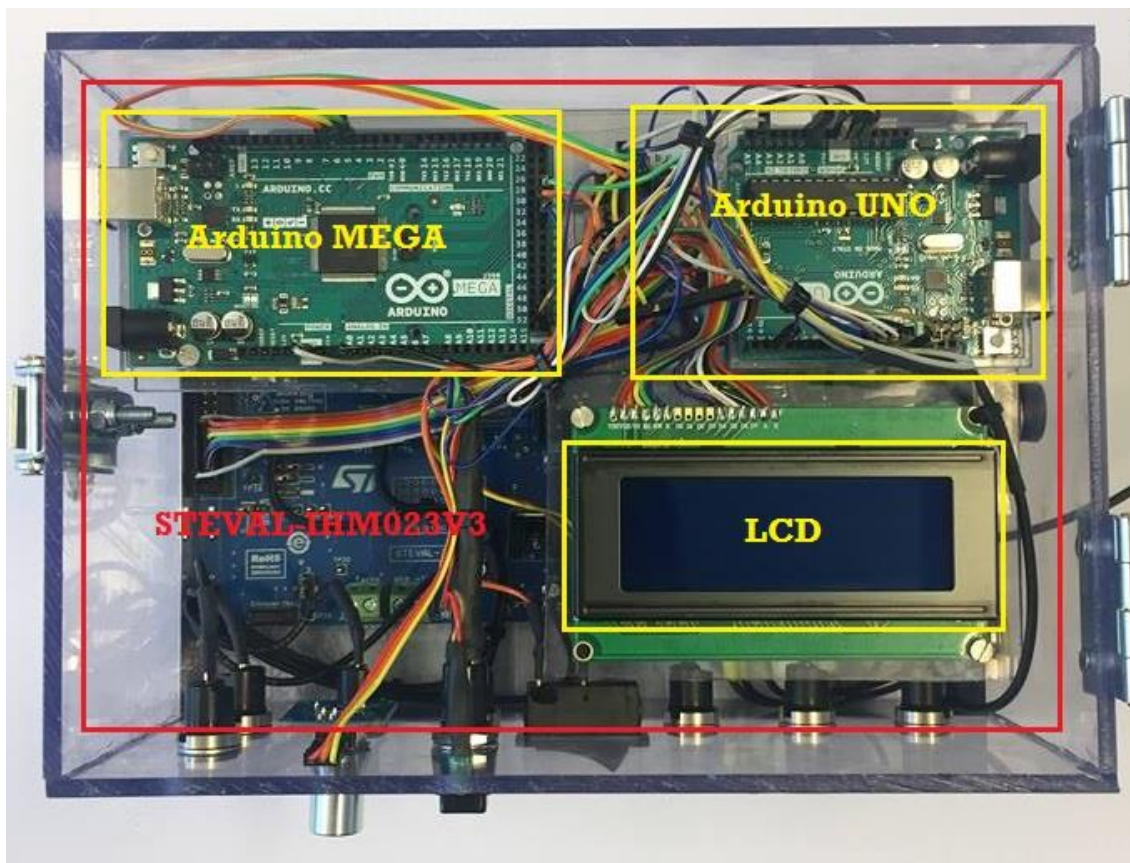
Her ser man det ferdige komplette fysiske produktet av oppgaven. Selve modellen med tilhørende kabler og motor.

For å beskytte alt utstyret har vi valgt å plassere det i en robust plastkoffert som er lett å bære med seg. Det er skjært ut spor i det isolerende skummet slik at brukerne enkelt kan pakke utstyret sammen, og for å isolere mot støt/slag.



Figur 26 – Bilde av modellen med beskrivelse av punkter

1. Referanse gate-signal
2. Gate-signal T1 og T4
3. Gate-signal T3 og T6
4. Gate-signal T5 og T2
5. Potmeter for å endre frekvens
6. Start/Stopp bryter med grønn LED
7. Velgebryter dreieretning
8. Uttak til motor fase A
9. Uttak til motor fase B
10. Uttak til motor fase C
11. PE Jordleder til motor
12. IR sensor
13. Sikring 6A smeltesikring
14. Hovedbryter
15. Tilførsel
16. USB-inngang til Arduino Mega
17. USB-inngang til Arduino Uno



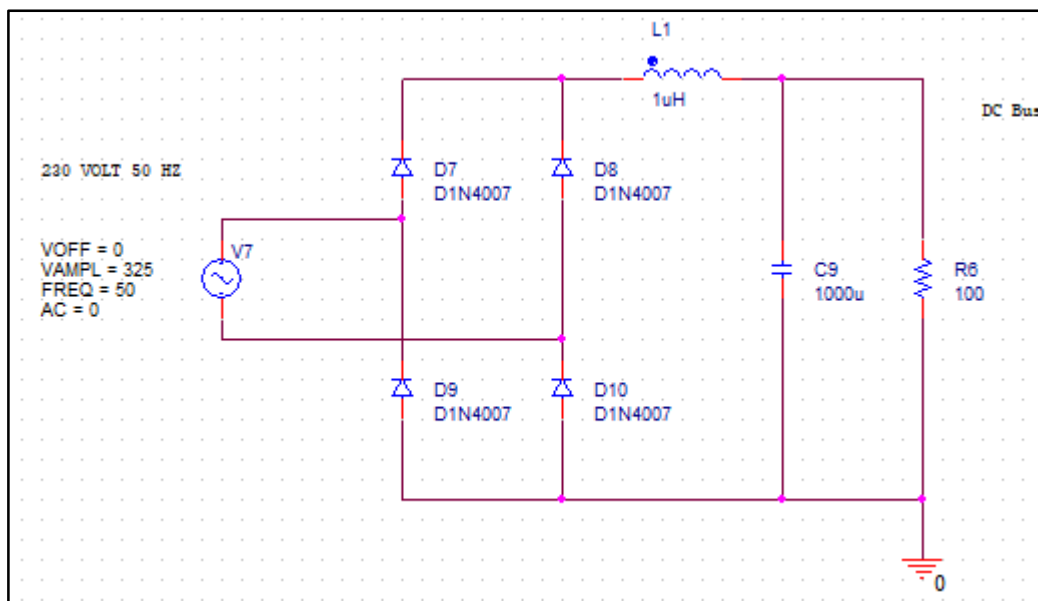
Figur 27 – Bilde av modell (komponentoversikt)

Her er en oversikt over de største og viktigste komponentene vi har valgt. I rød omriss er motorkontrollerkortet fra STMicroelectronics, som likeretter og vekselretter tilførselen og sender spenningen ut på motoren.

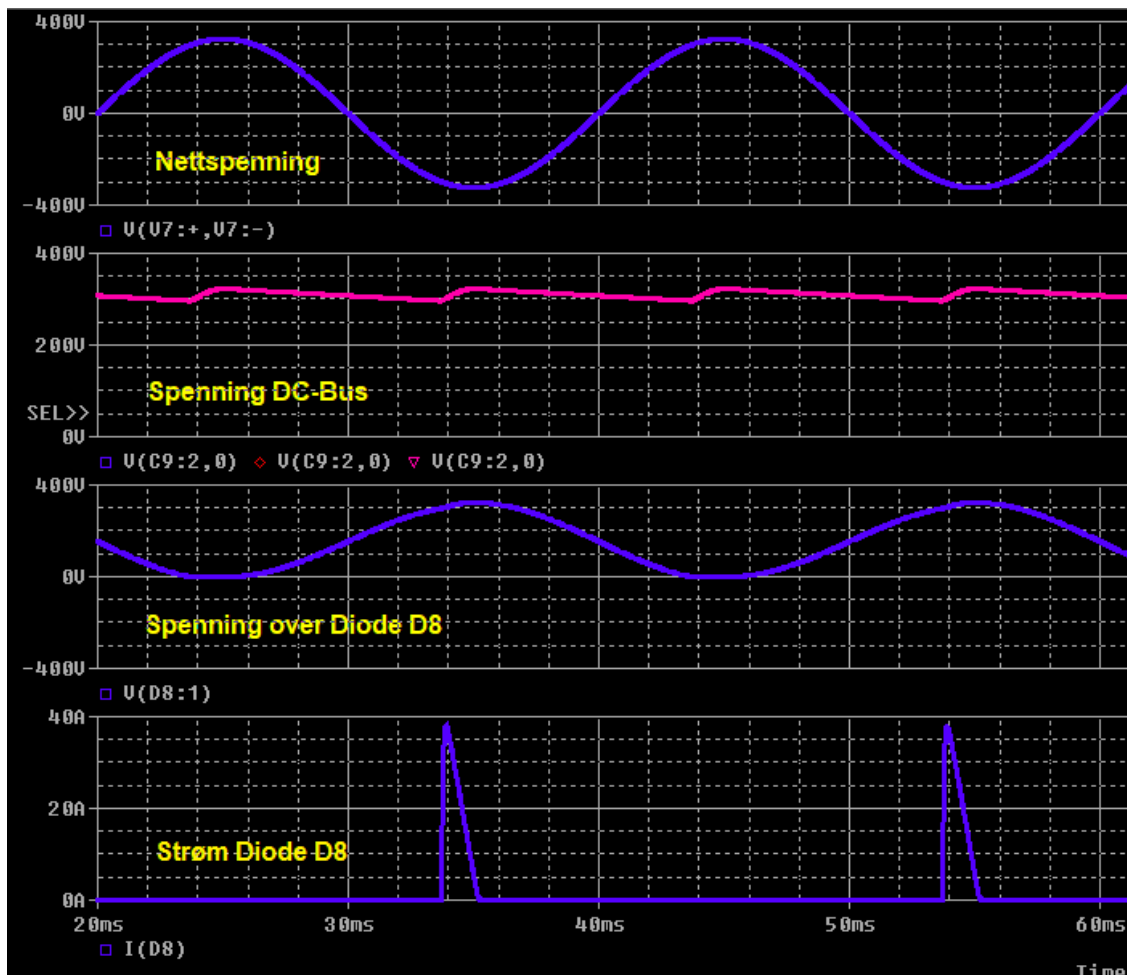
I gule omriss er Arduino Mega som tar for seg styringen av vekselrettingen, og Arduino Uno som overvåker DC-buss, frekvens og turtall og sender det til LCD-skjermen som viser verdiene.

Kapslingen av modellen er laget i plexiglass festet med komponent lim. Den er i tillegg utstyrt med hengelås som en ekstra sikring mot utilsiktet berøring. I henhold til NEK 410-1: 2008 har vi anslått modellen til å ha kapslingsgrad *ingress protection* 31 (IP31¹⁹) der det første sifferet angir at den er beskyttet mot faste fremmedlegemer større enn 2,5mm. Det andre sifferet sier at den er beskyttet mot vertikalt dryppende vann (NEK 2008, 10).

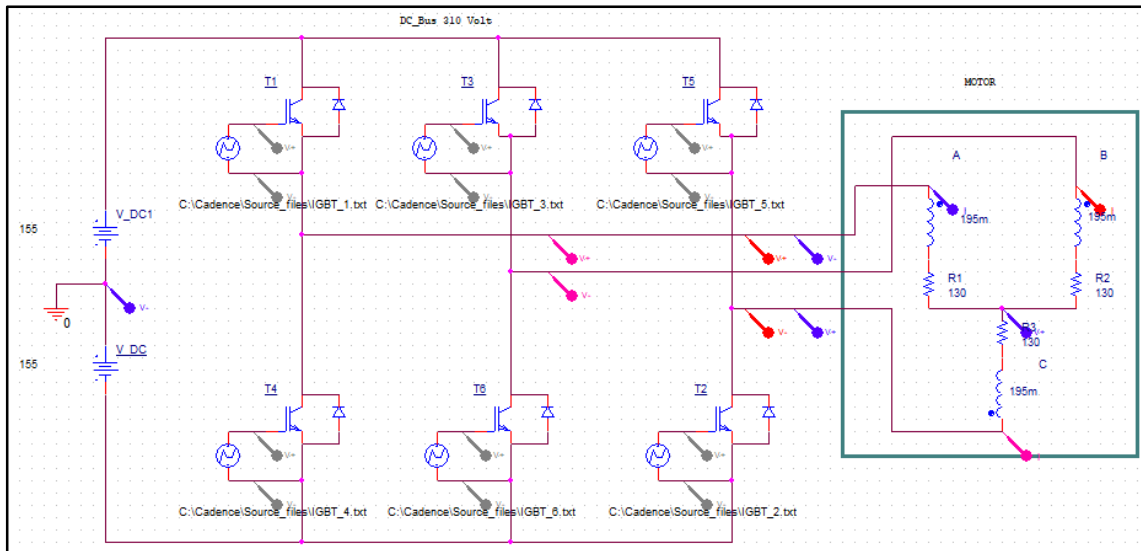
¹⁹ Oppgir beskyttelsesgraden til et utstyr.



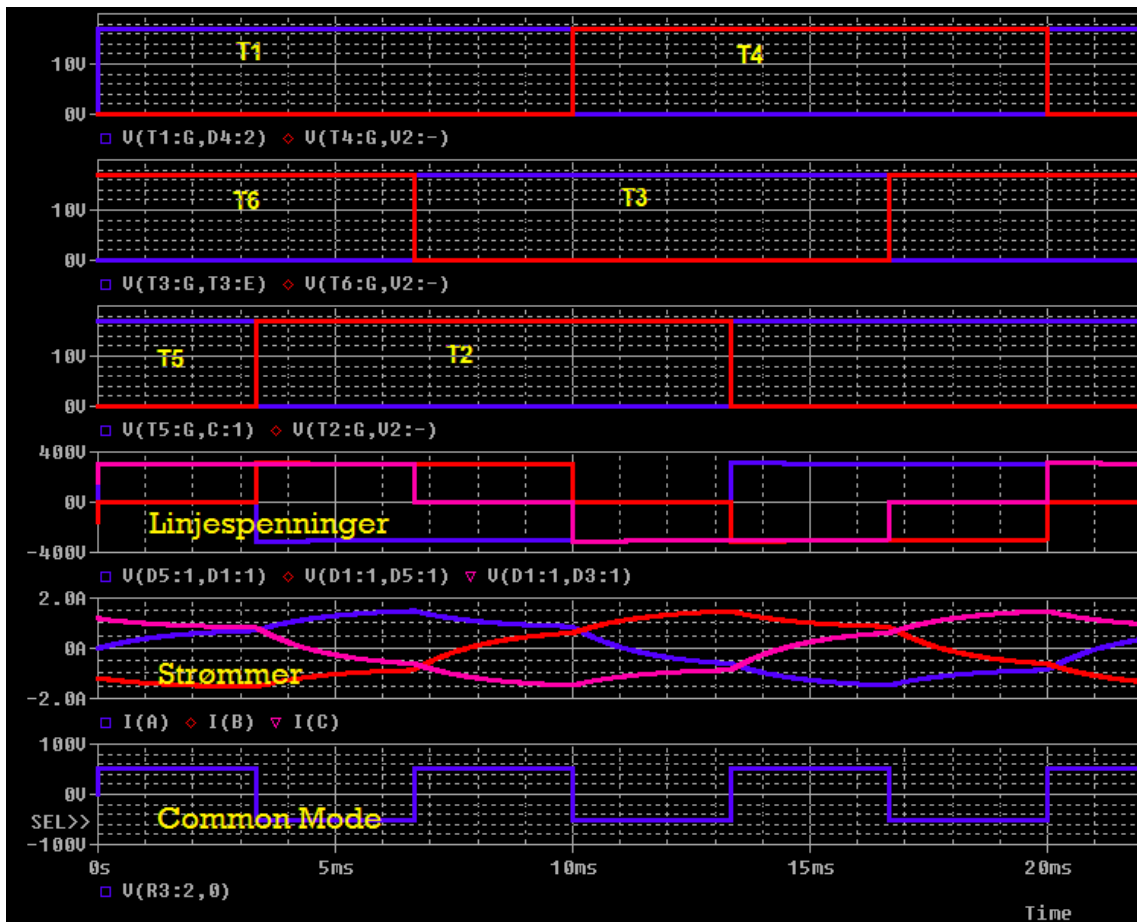
Figur 29 – Utklipp av utviklet likeretterkrets i OrCAD



Figur 28 – Utklipp av simulert likeretterkrets



Figur 31 – Utklipp av vekselretterkrets i OrCAD



Figur 30 – Utklipp av simulert vekselretterkrets

4 Utviklet software

Dette kapitlet vil vise og forklare programvarene som er designet til styringen i undervisningsmodellen. Det brukes to ulike programmer; et program for motorstyring og et program for overvåking, hvert program til ett av de to Arduino-kortene som blir brukt. Programmet for motorstyring er på Arduino Mega, mens programmet for overvåking er på Arduino Uno.

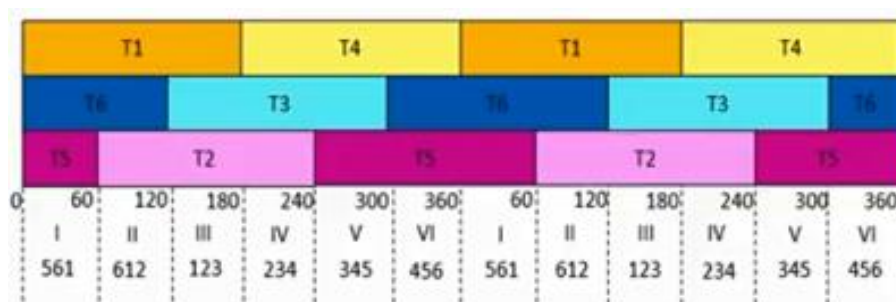
4.1 Motorstyring

Delkapitlet vil ta for seg hvordan motorstyring er realisert gjennom fysiske komponenter, koding, og kommunikasjon. Alt dette skjer gjennom Arduino Mega.

4.1.1 IGBT-styring med six step

Det er denne styringen som får motoren til å gå rundt. Vekselretting av DC-bussen skjer med six step-metoden. De seks transistorene blir styrt av digitale utganger fra Arduino Mega som vil åpne eller lukke transistorenes, slik at strømmen kan passere ut til motoren. Argumentene i programmet som blir brukt for å åpne og lukke transistorene i riktig rekkefølge til rett tid, er boolske variabler som endrer status mellom HIGH og LOW. Tidsforsinkelser (delay) er også lagt inn mellom hver bolk med kode i denne delen.

Utgangene fra Arduino vil sende signaler til komparatorene (L6390), som videre vil sende signal til transistorenes *gate-pin*. Gate-pinnen vil tenne/slukke og slippe gjennom DC-spenningen relativt til hva tiden på delay-koden er satt til, som igjen vil skape spenningens frekvens ut til motoren. Måten dette ble realisert på var å bruke en skisse for hvordan transistorene i en six step-sekvens vil se ut med 180° mode.



Figur 32 – Six step-sekvens med transistorer
(Xplanator, 2017)

Ut ifra figur 28 ble det antatt at mikrokontrolleren måtte sende ut seks signaler for å få til en slik sekvens på transistorene, ettersom det er seks transistorer. Etter undersøkelse av hvordan motorkontrollerkortets transistorer var koblet, endret dette seg. Sannhetstabellen for når transistorene blir tent gjennom komparatorene sine koblingspunkter viser til hvordan signaler den skal ha for å aktivere transistorparene.

Tabell 2: Sannhetstabell for komparator L6390
(ST 2018, 14)

Input			Output	
\overline{SD}	\overline{LIN}	HIN	LVG	HVG
L	X ⁽¹⁾	X ⁽¹⁾	L	L
H	H	L	L	L
H	L	H	L	L
H	L	L	H	L
H	H	H	L	H

1. X: don't care.

Fra figur 28 er transistorparene henholdsvis T1/T4, T3/T6 og T5/T2 som tilsvarende STEVAL-kortets Q6/Q7, Q8/Q9, og Q10/Q11. De to nederste linjene i sannhetstabellen viser at output LVG og HVG endres fra L (LOW) til H (HIGH) motsatt av hverandre så lenge LIN og HIN endrer seg likt. Etter dette ble oppdaget var ideen å gi ut samme signal på både LIN og HIN fra styringssignalet, altså kun ett og samme signal til hver komparator, som indirekte alltid ville gi motsatte signaler til LVG og HVG på samme transistorpar. I koden kalles utgangssignalene til parene *t1_t4*, *t3_t6*, og *t5_t2* for at det skal være enkelt å forstå og se visuelt opp mot figur 28 som har vært utgangspunktet for kodingen. De fysiske utgangssignalene sendes fra pin 4, 5, og 6 på Mega.

```

Mototrstyring_Mega
//180 GRADERS SIX STEP INVERTER

//Her deklarereres alle variabler

const int t1_t4 = 4;
const int t3_t6 = 5;
const int t5_t2 = 6;

```

Figur 33 – Variablene som aktiverer transistorparene

For å få til riktig tidsintervaller mellom hvert par har det blitt lagt inn tidsforsinkelser på en tid som representerer 60°, altså hver gang endringer i signaler skal skje. Tiden avhenger av frekvensen man ønsker spenningen ut til motoren skal svinge med. Løsningen på dette var å programmere et regnestykke som regner ut hvor lang tid 60° er ved angitt frekvens. Variabelen i koden kalt *hz* kan endres med potmeter²⁰. For eksempel hvis plasseringen av potmeteret representerer 50hz vil regnestykket regne ut at et 60° steg av 50hz har en tid på 0.00333 sekunder.

Regnestykket ser slik ut: $\text{steg} = ((1.0/\text{hz}) * 1000.0) / 6.0$

Eksempel: $\text{steg} = ((1.0/50) * 1000.0) / 6.0 = 3.333$

(Grunnen til at det multipliseres med 1000 er å få svaret i millisekunder)

Videre brukes verdien i variabelen *steg* som forsinkelse mellom hver endring av signaler.

Variablene til de digitale utgangene skrives som HIGH eller LOW. Når *t1_t4* er HIGH vil ifølge sannhetstabellen (tabell 2) T1 være HIGH, mens T4 være LOW. Når *t3_t6* er LOW, vil T3 være LOW, mens T6 er HIGH. Når *t5_t2* er HIGH, T5 være HIGH og T2 LOW.

Som indikert i figur 28 for 0-60° er T1, T6 og T5 HIGH. Dette stemmer med utredelsen over. Når *steg*1000* i mikrosekund har passert, vil neste trinn (60-120°) ta over. Dette gjelder helt ned til og med 300-360° og helt på nytt igjen siden koden er i *loop*-delen.

Denne metoden vil skape en momentan stopp på motoren når startknappen deaktiveres. Det skjer fordi IGBT t4, t6 og t2 vil få et høyt signal, når både HIN og LIN er lave. Som følge av dette vil motorviklingene kortsluttes og fungere som en bremsemotstand.

```
//0-60GRADER
digitalWrite(t1_t4, HIGH);
digitalWrite(t3_t6, LOW);
digitalWrite(t5_t2, HIGH);
delayMicroseconds(steg*1000);
//60-120GRADER
digitalWrite(t1_t4, HIGH);
digitalWrite(t3_t6, LOW);
digitalWrite(t5_t2, LOW);
delayMicroseconds(steg*1000);
//120-180GRADER
digitalWrite(t1_t4, HIGH);
digitalWrite(t3_t6, HIGH);
digitalWrite(t5_t2, LOW);
delayMicroseconds(steg*1000);
//180-240GRADER
digitalWrite(t1_t4, LOW);
digitalWrite(t3_t6, HIGH);
digitalWrite(t5_t2, LOW);
delayMicroseconds(steg*1000);
//240-300GRADER
digitalWrite(t1_t4, LOW);
digitalWrite(t3_t6, HIGH);
digitalWrite(t5_t2, HIGH);
delayMicroseconds(steg*1000);
//300-360GRADER
digitalWrite(t1_t4, LOW);
digitalWrite(t3_t6, LOW);
digitalWrite(t5_t2, HIGH);
delayMicroseconds(steg*1000);
}
```

Figur 34 – Utklipp av koden for styring av transistorpar

²⁰ Komponent med justerbar motstand ved bruk av et skruhjul

4.1.2 Frekvensstyring

For justering av frekvensen motoren kjører på, brukes et analogt signal fra et potmeter som varierer signalet inn på en analog inngang på Arduino Mega fra 0 (minst utslag på hjulet) til 1023 (maks utslag på hjulet). Tallet som oppfattes i programmet fra inngangen blir videre skalert fra 0 (minst utslag) til 50 (maks utslag) ved hjelp av en funksjon kalt *map*. *Map* skalerer verdispenn fra inngangen til ønskelig verdispenn på utgangen. I dette tilfellet tar funksjonen verdispennet 0-1010 fra potmeteret og gjør det om til spennet 20-50.

```
sensorverdi = analogRead(potmeter);  
hz = map(sensorverdi, 0, 1010, 20, 50);
```

Figur 35 – Koden for hertzjustering

Variabelen *sensorverdi* lagrer verdien fra potmeteret. Variabelen *hz* får sin verdi mellom 20 og 50 avhengig av verdien til *sensorverdi*, som videre blir brukt til six step-koden sin tidsforsinkelse.

4.1.3 Motortilstander

Programmet for hovedtilstandene til motoren baserer seg på enkle digitale argumenter for start/stopp og endring av dreieretningen til motoren. Dette gjøres ved å bruke to fysiske trykknapper som konstant endrer signalet gjennom bryterne ved hvert trykk. I den ene posisjonen vil signalet være 5V (HIGH), mens i det andre vil det være 0V (LOW).

Når trykknappen for start/stopp ligger inne og passerer 5V til digital inngang 31, vil loopen for signalgiving til transistorene bli aktivert. Det er implisitt at den ellers, når knappen ikke sender gjennom 5 V, er deaktivert slik at signalene til transistorene ikke sendes.

```
const int start = 31;  
const int dreieretning = 33;  
int les_start;  
int les_dreieretning;  
les_start = digitalRead(start);  
les_dreieretning = digitalRead(dreieretning);
```

Figur 36 – Deklarering av knappene

```
if (les_start == 1){  
  
    digitalWrite(led, HIGH);  
  
    if (les_dreieretning == 1){  
        //0-60GRADER  
        digitalWrite(t1_t4, HIGH);  
        digitalWrite(t3_t6, LOW);  
        digitalWrite(t5_t2, HIGH);  
        delayMicroseconds(steg*1000);  
        //60-120GRADER
```

Figur 37 – Øverste del av six step-koden med argument for å aktivere den

Når trykknappen for dreieretning ligger inne og lar 5V passere, vil signalene til transistorene sendes i den rekkefølgen koden ble satt opp etter tidligere vist tabell 2. Er knappen i posisjon for 0 V til digital inngang 33, vil koden for signalene til transistorene gå i loop baklengs i forhold til figur 28. Dette får to av signalene til å bytte rekkefølge, og motorakslingen vil dreie den andre retningen.

```
if(les_dreieretning == 0){  
    //300-360GRADER  
    digitalWrite(t1_t4, LOW);  
    digitalWrite(t3_t6, LOW);  
    digitalWrite(t5_t2, HIGH);  
    delayMicroseconds(steg*1000);  
    //240-300GRADER
```

Figur 38 – Kodeloop for baklengs signalgivning for motsatt rotasjon



Figur 39 – Bilde av start/stopp-knapp, (til venstre), dreiretningsknapp og LED

4.2 Måling og overvåking

Delkapittelet vil ta for seg hva som blir målt, regnet ut, og videre fremvist for at brukerne av modellen kan overvåke de ulike parameterne. Alt dette styres fra Arduino Uno.

4.2.1 LCD-skjerm

En LCD-skjerm er koblet til Arduino Uno for å kunne vise de tre parametrene; turtall, frekvens og spenning på DC-bussen.

Turtallet og frekvensen vises slik at det er mulig å se sammenhengen mellom frekvensen på spenningen og motorakslingens rotasjon. Hensikten med visning av DC-buss-spenning er å tilby oversikt over om modellen har riktig spenning til å kjøres, i tillegg til å vise at DC-spenningen er om lag 1.35 ganger tilførselens RMS-verdi, og at den sakte synker etter tilførselen er frakoblet på grunn av kondensatorenes energilagringsevne.



Figur 40 – Bilde av LCD-skjermen som blir brukt til å vise resultater

4.2.2 Turtall på aksling (RPM)

Turtallet på motorakslingen blir målt ved å bruke en sirkulær skive, IR-sensor, og en utregningskode i Arduino Uno. Skiven har fire punkter symmetrisk plassert og er festet på akslingen. Dens hensikt er å sette opp en forutsigbar skala som representerer hvor langt akslingen har dreid. IR-sensoren vil gi et signal til en inngang på kortet hver gang sensoren registrerer et gjennomiktig punkt, siden lyset da blir registrert. For hvert fjerde signal vet vi at akslingen har gått rundt en hel gang. Signalene vil deretter bli registrert i programmet, hvor antall signaler blir addert og lagret. Hvert sekund vil antall signaler bli

registrert, regnet ut og sendt ut til LCD-skjermen som viser resultatet for hvor fort akslingen spinner.

Koden er bygd opp slik at *void isr()* er en *interrupt*-funksjon som blir aktivert hver gang IR-sensorens signal har en *falling edge*, dvs. går fra høy til lav. Hver gang funksjonen blir aktivert vil variabelen *teller* økes med én. Interrupt-funksjonen vil kjøre uten å påvirke kjøringen av andre deler av koden.

```
void isr()
{
    teller++;
}
```

Figur 42 – Utklipp av isr-funksjonen

```
if( b_sek)
{
    b_sek=false;
    detachInterrupt(0);
    rpm = teller*60/4;

    teller = 0;
    attachInterrupt(0, isr, FALLING);
}
```

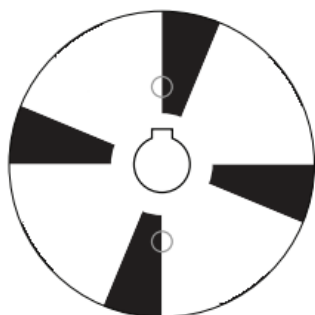
Figur 41 – Utklipp av koden for flagg

Her er koden som sier at for hver gang digital inngang 2 (IR-sensor) registrerer en fallende flanke, skal interrupt-funksjonen bli aktivert:

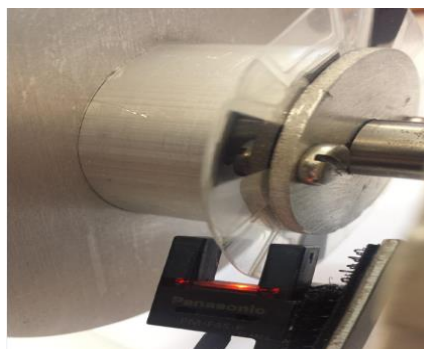
```
attachInterrupt(digitalPinToInterrupt(2), isr, FALLING);
```

Figur 43 – Utklipp av koden som initierer interrupt

Variabelen *b_sek* er et flagg som aktiveres hvert sekund. En gang i sekundet vil *if-loopen* aktiveres og gjennomføre koden i loopen. Det første som skjer er at flagget settes lavt igjen, slik at det kan aktiveres neste sekund. Etter dette avbrytes interrupt-sekvensen, slik at tellingen stopper opp. Deretter regnes rpm ut, hvor *teller* er variabelen som har summert opp IR-sensorens pulser. *Teller* multipliseres så med 60 for å få rotasjoner per minutt. Tallet divideres så med 4 for pga. telleskivens oppløsning – en rotasjon tilsvarer fire pulser. Teller nullstilles slik at den kan begynne å summere tall fra null. Til slutt restartes interrupt-prosesseringen, slik at *void isr()* kan aktiveres og teller kan begynne å summere pulser igjen.



Figur 44 – Bilde av telleskivemal



Figur 45 – Bilde av IR-sensor og telleskive

4.2.3 Innstilt frekvens

Det som representeres på LCD-skjermen er den innstilte frekvensen som justeres med potmeteret. Dette er gjort ved å bruke det samme signalet og den samme type koden fra potmeteret som er brukt på Arduino Mega, men istedenfor å regne ut tidsforsinkelse som i Mega, brukes verdien i variabelen *hz* rett ut på LCD-skjermen for å vise hvilken frekvens som er innstilt.



Figur 46 – Bilde av potmeteret som justerer innstilt frekvens

4.2.4 Spenning på DC-bussen

Måling av denne spenningen blir utført ved å koble til en utgang fra motorkontrollkortet som gir en spenning på mellom 0 og 5V inn på en Arduino-inngang. Spenningen representerer et forholdstall på hva spenningen på DC-bussen faktisk er. Forholdstallet ble beregnet ved å måle den faktiske DC-spenning og utgangssignalet ved ulike spenninger inn på modellen. Dette er en liste på resultatene og regnestykket.

Tabell 3: Spenningsverdier og forholdstall

Spenning Variac	Spenning fra MC-kort	Tilsvarende Digital Verdi	Spenning DC-Link	Forhold Digital verdi/DC link
230	002.6	532	323.5	1,65
160	001.81	370	225.5	1,64
100	001.125	230	140.3	1,64

```
les_dc_link = map(analogRead(dc_link), 0, 541, 0, 324);
```

Figur 47 – Utklipp av koden for omgjøring til DC-buss-spenning

Funksjonen *analogRead()* leser av en verdi fra 0 til 5V og gjør den om til en digital verdi fra 0 til 1023. Dette blir gjort av en krets inne i mikrokontrolleren som kalles *analog-to-digital converter* (ADC) (Arduino, 2015). Når vi har 230V på inngangen til motorkontrollerkortet så vil det gi ut en verdi på 2,6V, som igjen blir oppfattet av Arduino som en digital verdi på 532. Den faktiske målte spenningen på DC-bussen vil i det tilfellet være 325,5V.

Det blir brukt en *map()*-funksjon som omdanner verdiene 0-541 til en lineær verdi mellom 0-324, som i dette tilfellet er DC-busspenningen som ønskes skrevet ut. Vi velger en høyere digital verdi i tilfelle vi har en høyere spenning enn 230V inn på motorkontrolleren. Energien som er lagret vil synke med tiden. Det er ikke nøyaktige mål, men tilnærmet og det virker bra nok til å illustrere det vi ønsker.

5 Måleresultater

I dette kapitlet vil de mest interessante måleresultater av ulike spenning- og strømsignaler, motorhastigheter, og temperaturer finne sted. I den hensikt å få en formening om hvordan modellen presterer

Måleinstrumentene som blir brukt i disse forsøkene er:

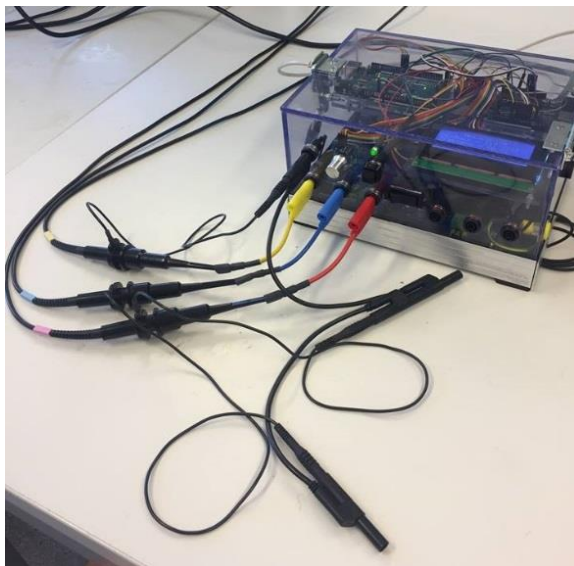
- Fluke 43B (Power Quality Analyzer)
- Amprobe 38XR-A (Multimeter)
- Tektronix TPS 2014 (Oscilloskop)
- Tektronix A622 (Strømtang/Probe)

(Spesifikasjoner for instrumentene finnes som vedlegg)

MERK: Når det står hvilken frekvens det er målt med (for eksempel 50Hz) så er dette innstilt frekvens på modellen gjennom Arduino. Instrumenter viser noe avvik fra hva som måles av frekvens i forhold til det som er innstilt. For eksempel blir 50Hz innstilt registrert som 48,73Hz på oscilloskopet.

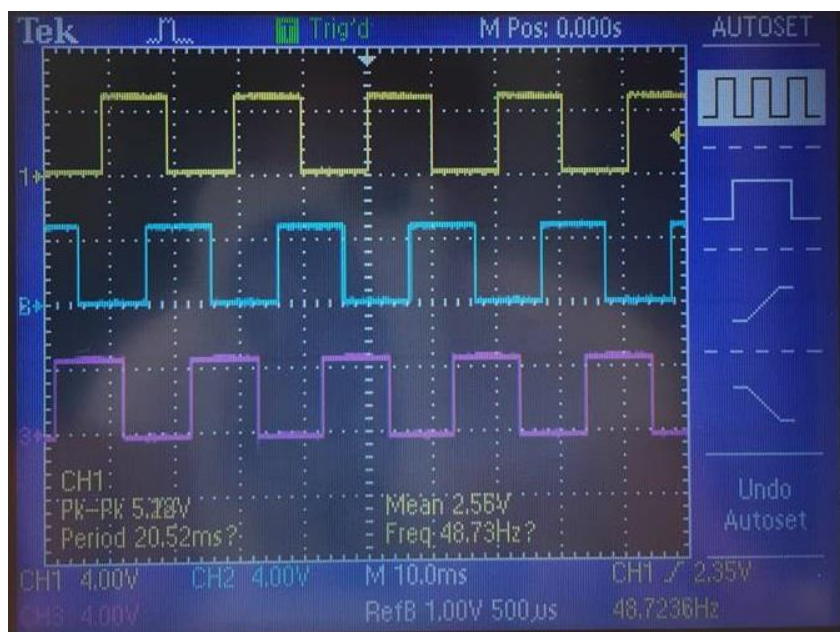
5.1 Gate-signaler

Her vil signaler for begge dreieretningene fra Arduino Mega gjennom L6390 til transistorparene bli presentert. Målingene er utført for både 20Hz og 50Hz. Instrumentet som ble brukt: Tektronix TPS 2014 (Oscilloskop).



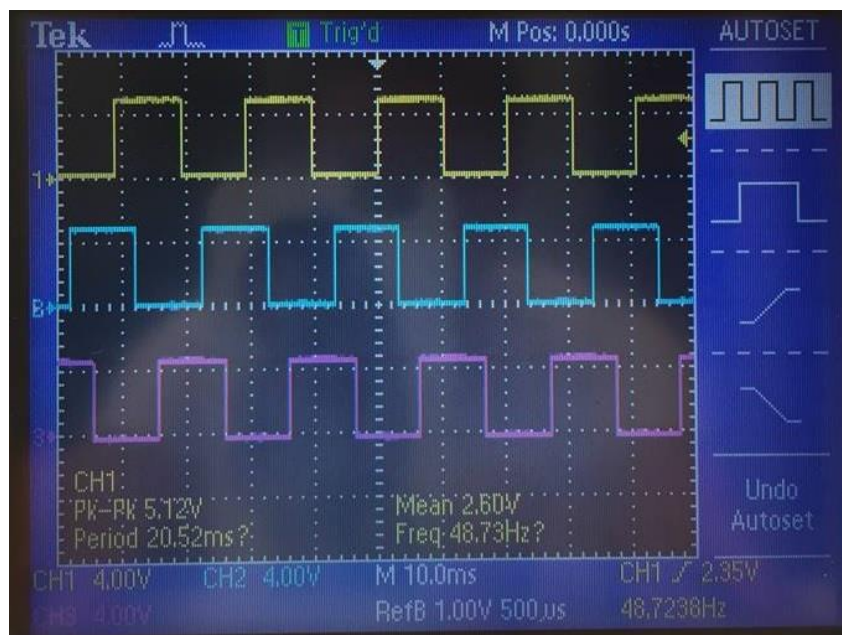
Figur 48 – Oversikt over oppkobling for måling av gate-signal.

5.1.1 Måling for 50Hz, retning A



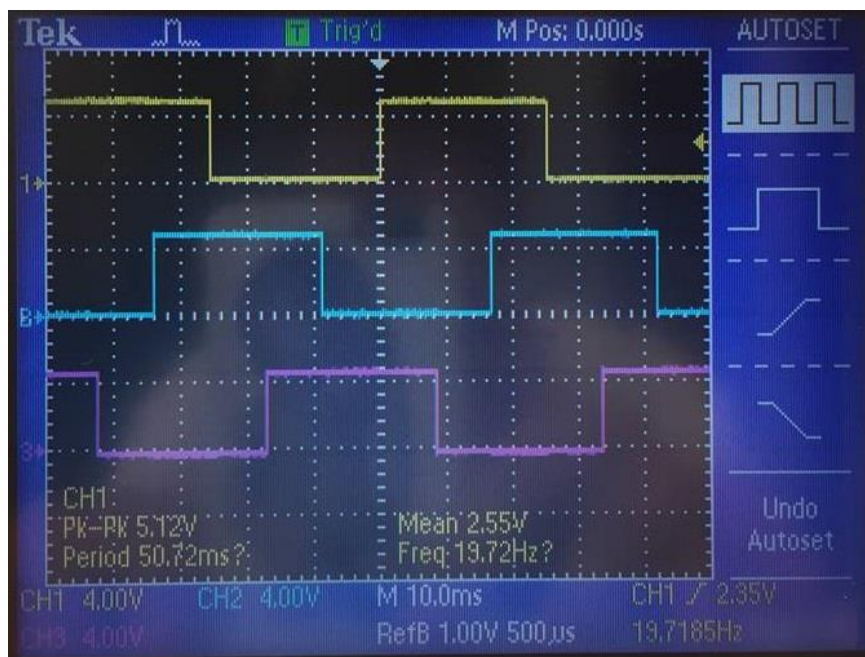
Figur 49 – Måling av gate-signaler på oscilloskop (50Hz, retning A)

5.1.2 Måling for 50Hz, retning B



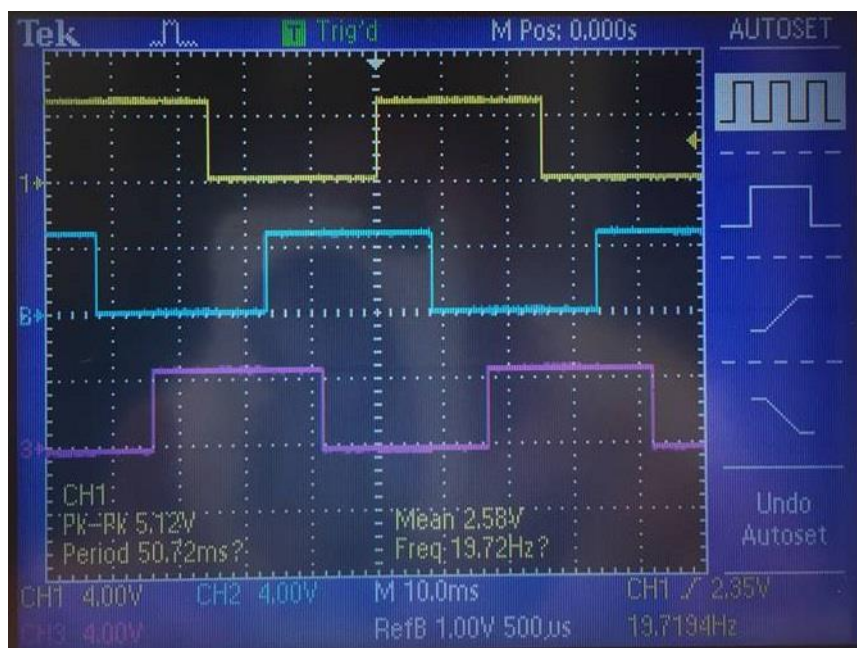
Figur 50 - Måling av gate-signaler på oscilloskop (50Hz, retning B)

5.1.3 Måling for 20Hz, retning A



Figur 51 - Måling av gate-signaler på oscilloskop (20Hz, retning A)

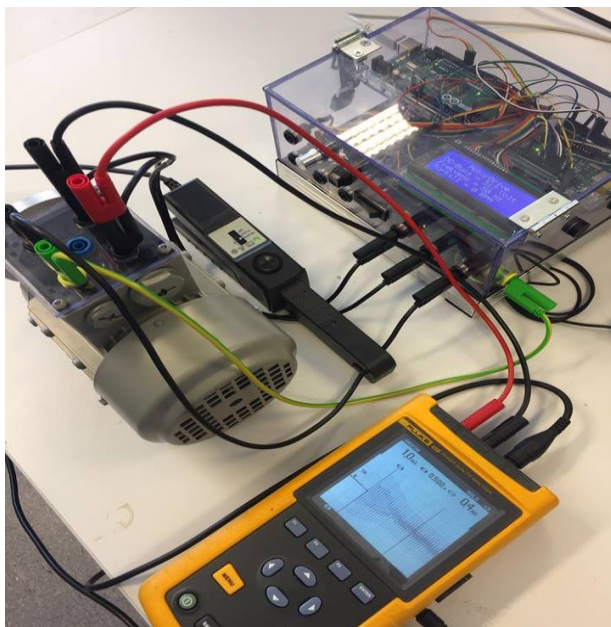
5.1.4 Måling for 20Hz, retning B



Figur 52 - Måling av gate-signaler på oscilloskop (20Hz, retning B)

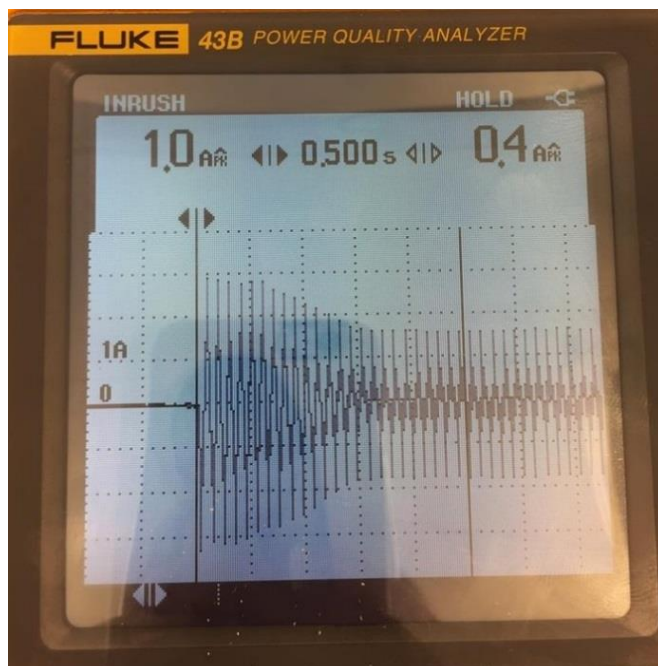
5.2 Strøm

Her vil malinger av ulike strømtrekk vises; startstrøm (inrush), strøm ved endring av dreieretning i fart, strøm ved stopp, og kontinuerlig strøm. Instrument som ble brukt: Fluke 43B (Power Quality Analyzer) og Tektronix A622 (Strømtang/Probe).



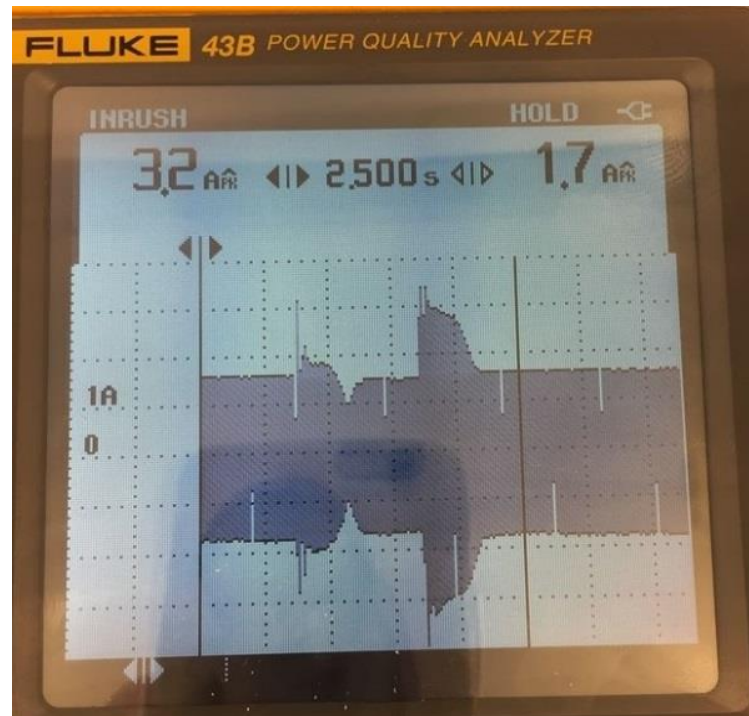
Figur 53 – Oversikt over oppkobling for måling av strømmer

5.2.1 Startstrøm (inrush)



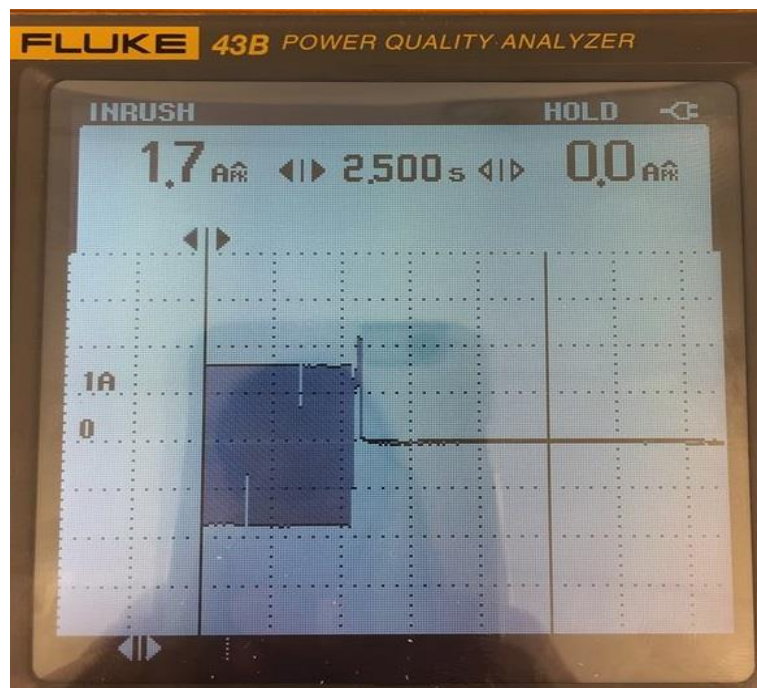
Figur 54 – Resultat av inrush ved 50Hz (Skalering: 1A, 1s)

5.2.2 Strømmåling ved endring av dreieretning



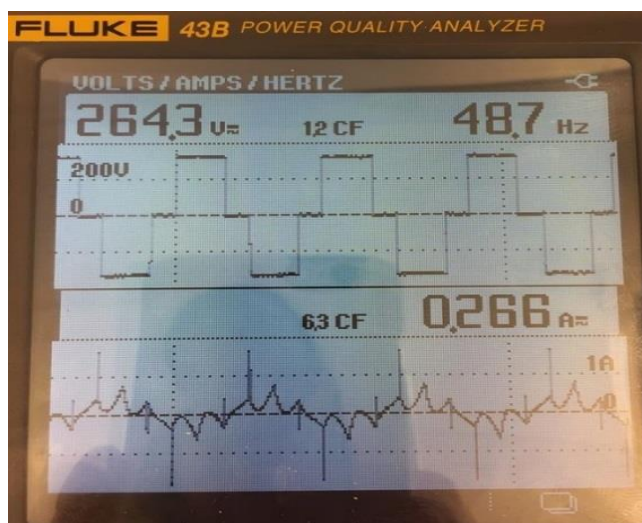
Figur 55 - Resultat ved endring av dreieretning to ganger ved 50Hz (Skalering: 1A, 1s)

5.2.3 Strømmåling ved stopp

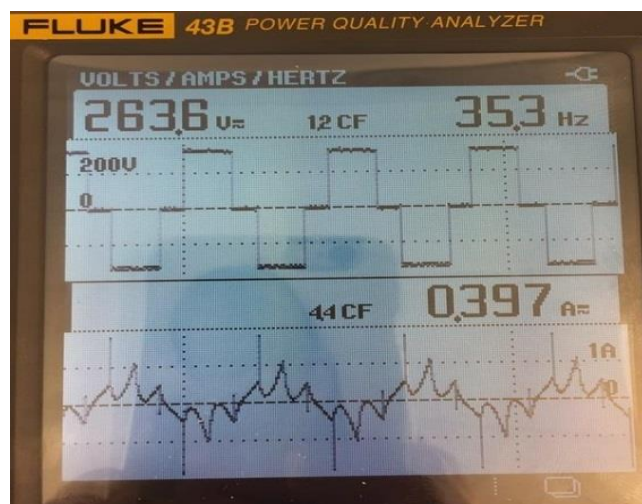


Figur 56 – Resultat ved stopp, 50Hz (Skalering: 1A, 5s)

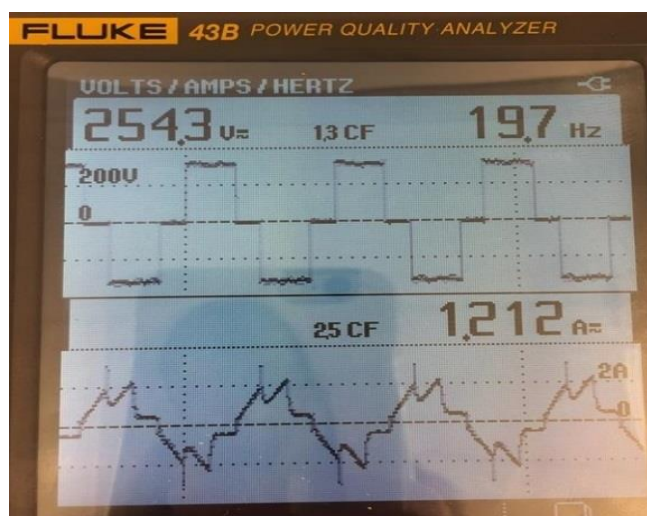
5.2.4 Kontinuerlig strømtrekk



Figur 57 – Kontinuerlig strømtrekk ved 50Hz



Figur 58 - Kontinuerlig strømtrekk ved 35Hz

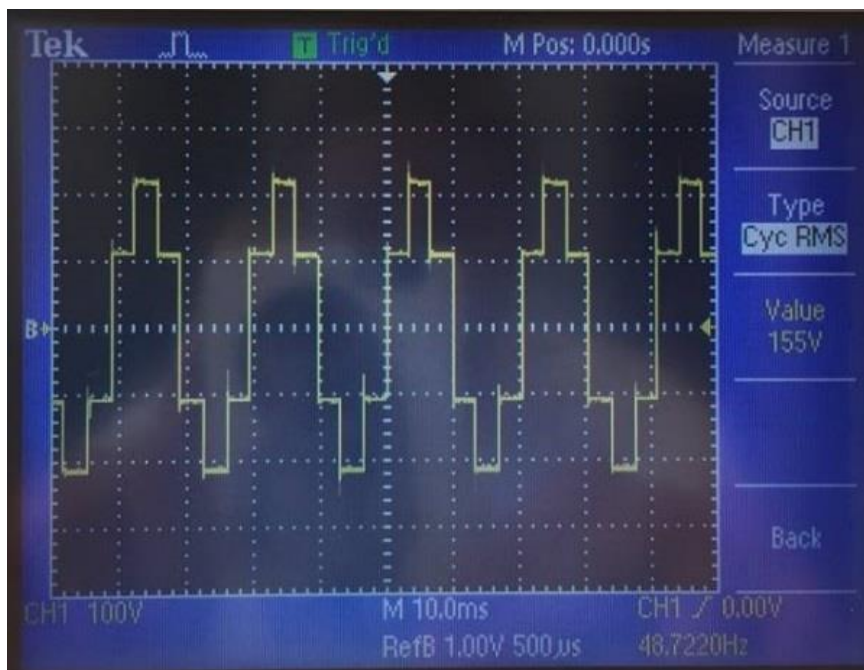


Figur 59 - Kontinuerlig strømtrekk ved 20Hz

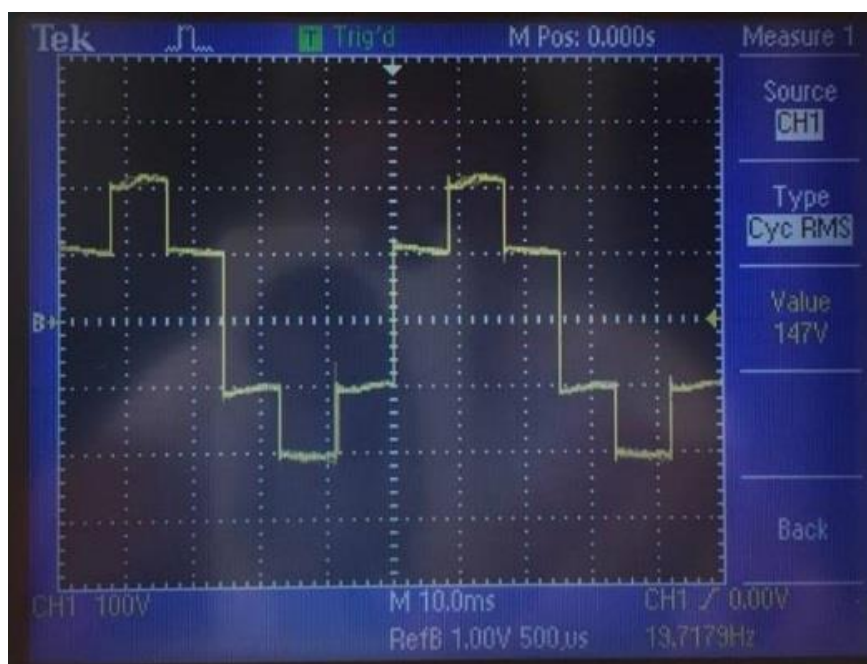
5.3 Spenning

Her vil målinger av ulike spenninger vises; fasespenning, linjespenning, og common-mode-spenning. Instrument som ble brukt: Tektronix TPS 2014 (Oscilliskop).

5.3.1 Fasespenning

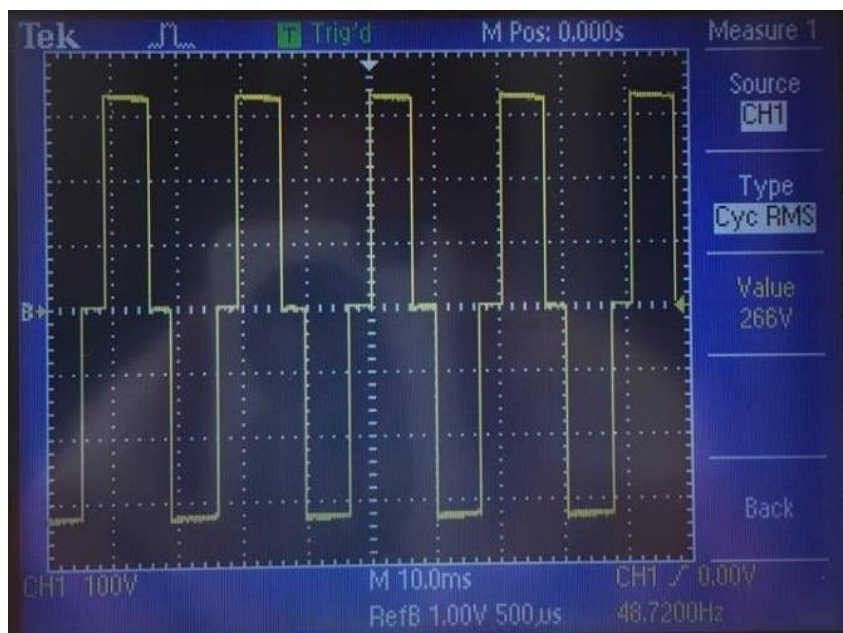


Figur 60 – Fasespenning ved 50Hz

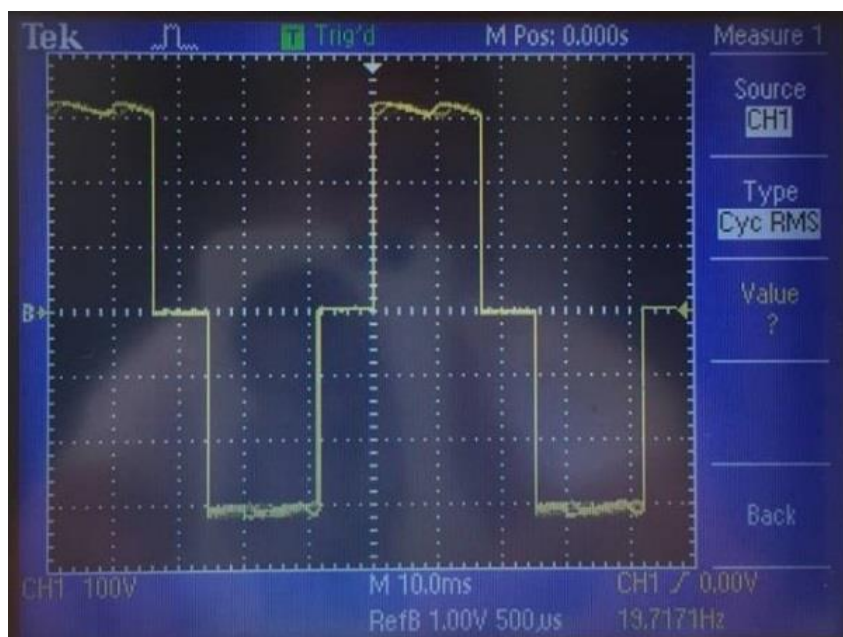


Figur 61 – Fasespenning ved 20Hz

5.3.2 Linjespenning



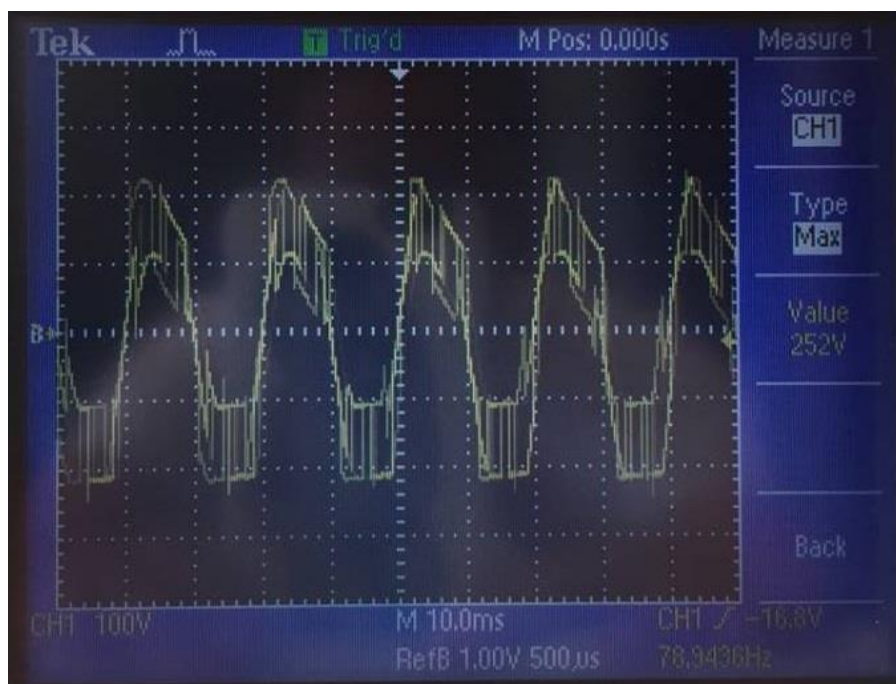
Figur 62 - Linjespenning ved 50Hz



Figur 63 - Linjespenning ved 20Hz

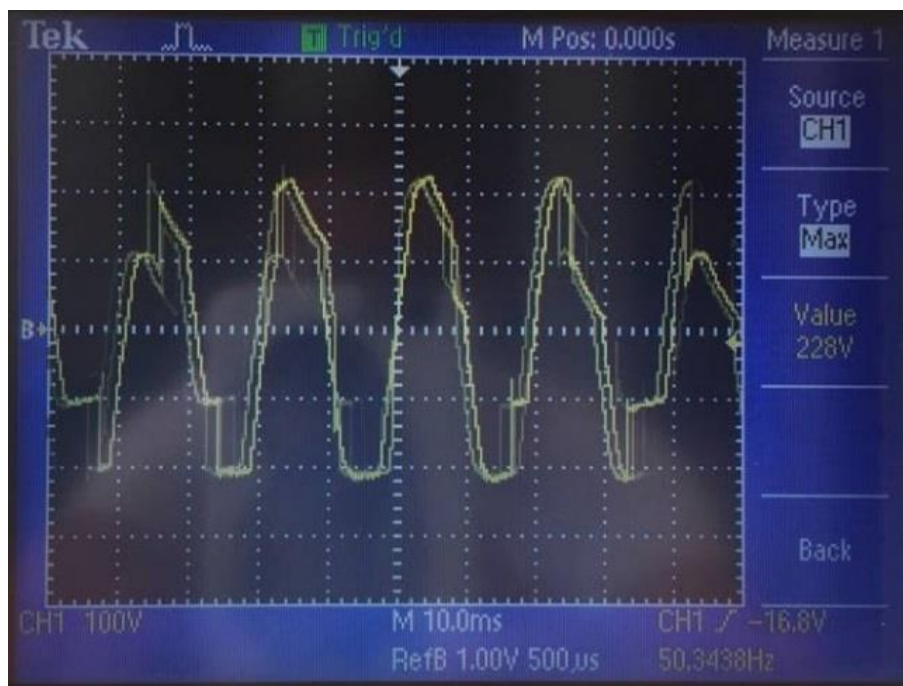
MERK: Oscilloskopet klarte ikke å måle RMS i dette tilfellet. 250V RMS målt med Amprobe 38XR-A (Multimeter).

5.3.3 Common-mode-spenning



Figur 64 – Common-mode-spenning ved 50Hz

MERK: 138V RMS målt med Amprobe 38XR-A (Multimeter).

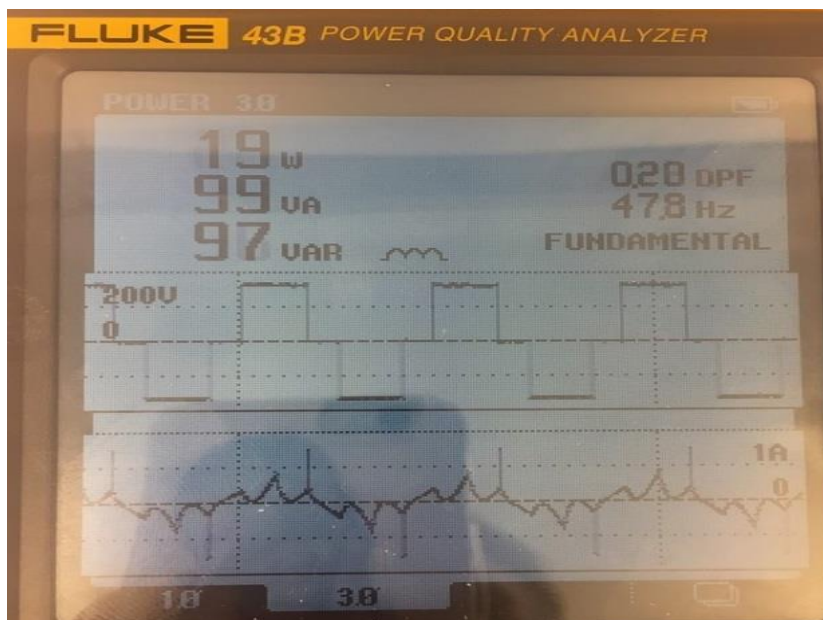


Figur 65 - Common-mode-spenning ved 20Hz

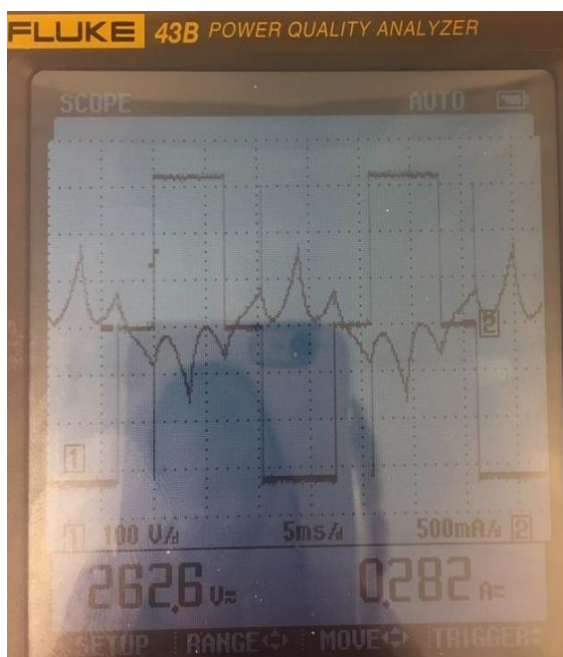
MERK: 138V RMS målt med Amprobe 38XR-A (Multimeter) her også.

5.4 Effekter

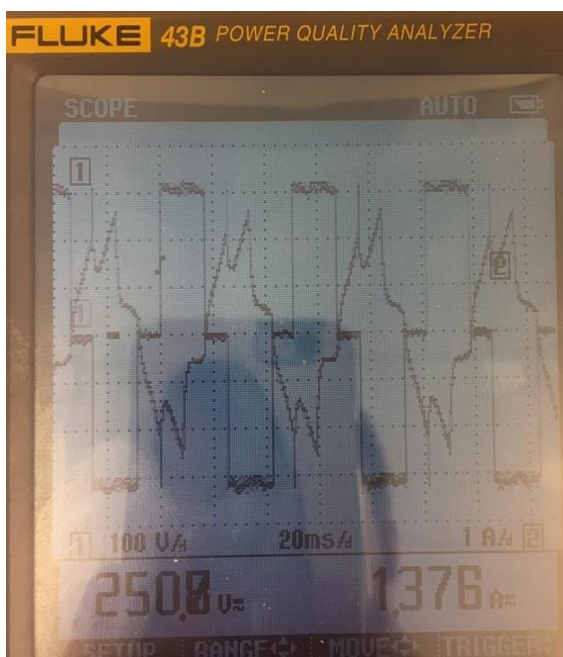
Her vil malinger av ulike effekter vises; spenning og strøm ved ulik frekvens. Instrument som ble brukt: Fluke 43B (Power Quality Analyzer) og Tektronix A622 (Strømtang/Probe).



Figur 66 - Effektmåling av motor ved 50Hz



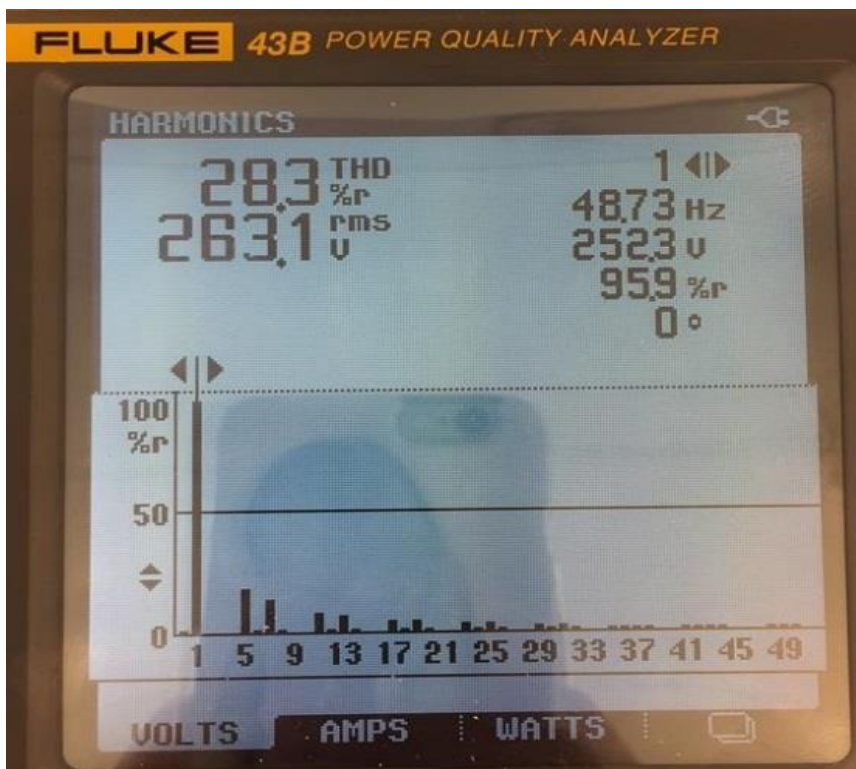
Figur 68 – Strøm og spenning i forhold til hverandre ved 50Hz



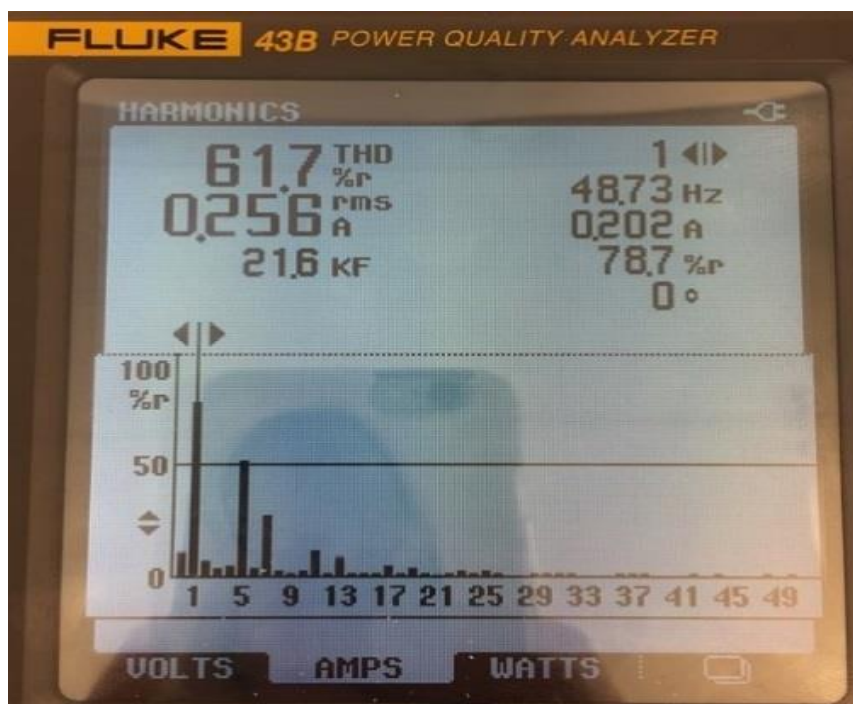
Figur 67 - Strøm og spenning i forhold til hverandre ved 20Hz

5.5 Harmonisk støy

Her vil målinger av harmonisk støy vises; spenning og strøm ved ulik frekvens. Instrument som ble brukt: Fluke 43B (Power Quality Analyzer) og Tektronix A622 (Strømtang)



Figur 70 – Harmonisk spenning ved 50Hz



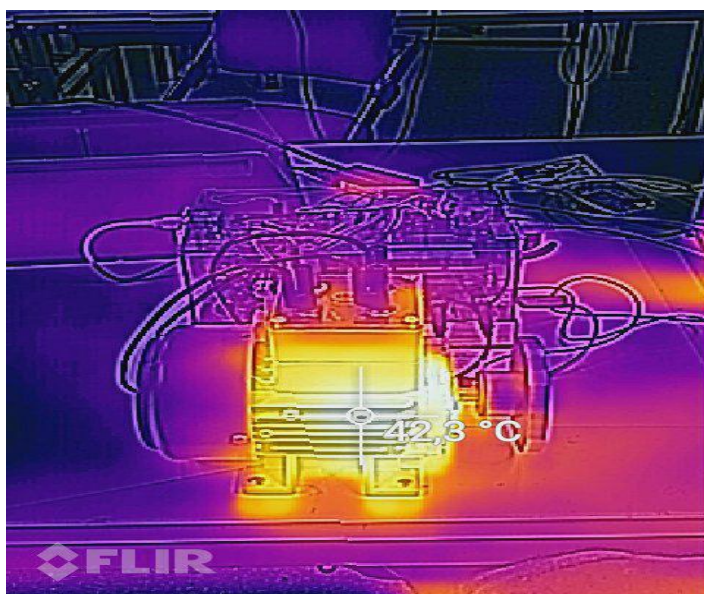
Figur 69 – Harmonisk strøm ved 50Hz.

5.6 Temperatur

For å undersøke hvor varm motoren kunne bli ble det utført ulike driftstester og tatt bilder med FLIR-kamera som resultatgiver. Motoren ble kjørt i 5 minutter med 20Hz, og oppnådde en temperatur rett under 50°C. Rett etter dette ble motoren kjørt i 50Hz i 5 minutter, temperaturen sank da ned mot 40°C på grunn av mer effektiv selvventilering.



**Figur 71 – Temperatur på motor
(20Hz motordrift 5 min)**



**Figur 72 - Temperatur på motor
(50Hz motordrift 5 min fra 48,7°C)**

5.7 Hastighet på aksling

Her vil resultat av rpm-måling vises for innstilt 20, 30, 40 og 50Hz. Dette er målt både med egenprodusert rpm-måler som tilhører modellen og et kommersielt foto-tachometer. Instrumenter brukt for dette: IR-sensor med telleskive og program og ELMA DT-2236



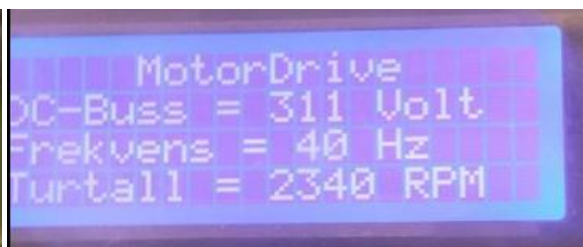
**Figur 74 - Måling av RPM ved 50Hz
(Tachometer)**



Figur 73 - Måling av RPM ved 50Hz (IR-sensor)



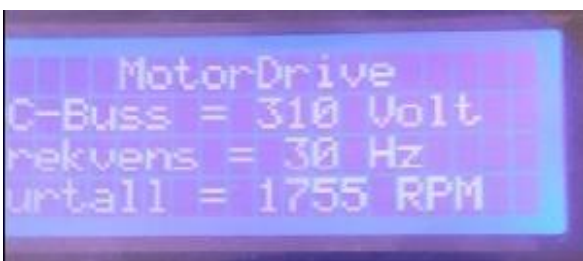
**Figur 77 - Måling av RPM ved 40Hz
(Tachometer)**



**Figur 78 - Måling av RPM ved 50Hz
(IR-sensor)**



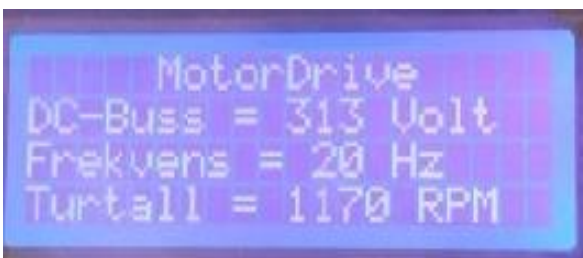
**Figur 79 - Måling av RPM ved 30Hz
(Tachometer)**



**Figur 80 - Måling av RPM ved 30Hz
(IR-sensor)**



**Figur 76 - Måling av RPM ved 20Hz
(Tachometer)**

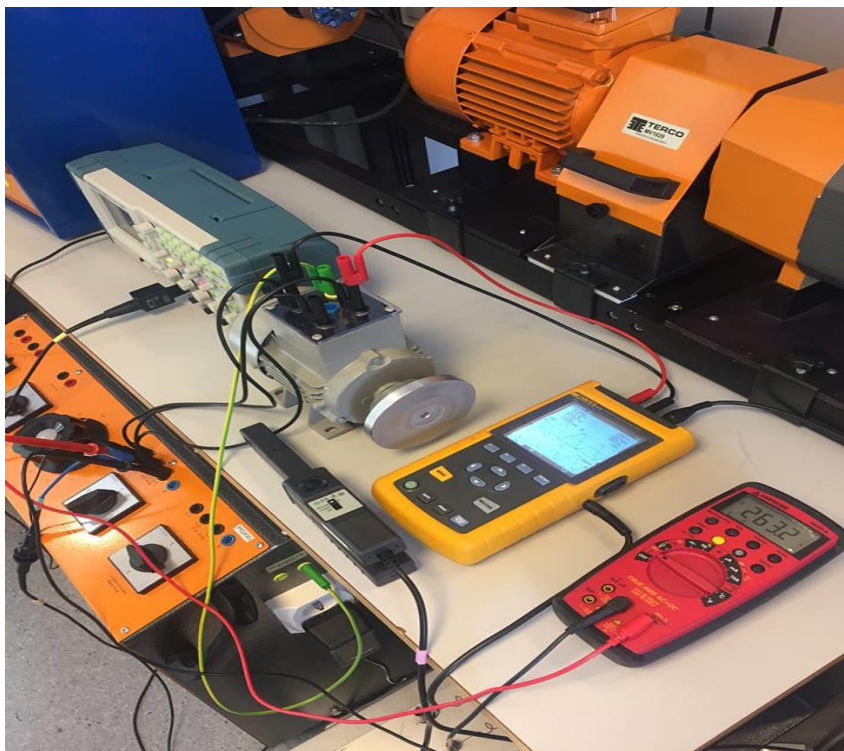


**Figur 75 - Måling av RPM ved 20Hz
(IR-sensor)**

5.8 DOL-kjøring av motoren

Her vil ulike målinger med *direct on line*-kjøring av motoren (DOL-kjøring²¹) vises slik at det er mulig å sammenligne hvordan modellen virker på motoren i forhold til fin sinus fra nettleverandør. En variac²² ble brukt til å oppnå samme RMS linjespenning (263V) som modellen gir ut til motoren slik at målingene blir mest mulig sammenlignbare. Instrument som ble brukt til følgende målinger:

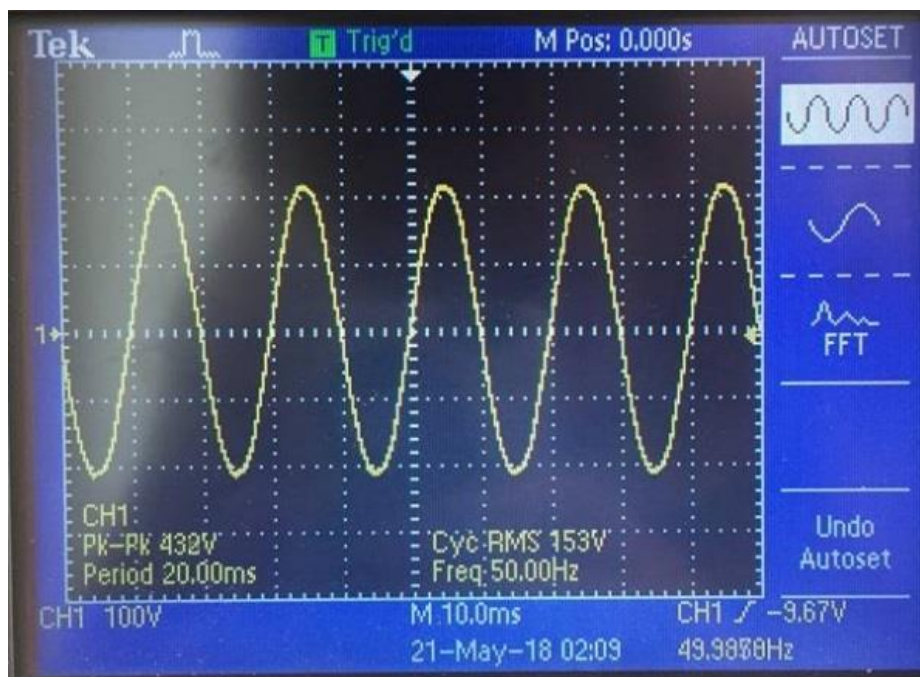
- Tektronix TPS 2014 (Oscilliskop)
- Amprobe 38XR-A (Multimeter)
- Tektronix TPS 2014 (Oscilliskop)
- TERCO MV1300 (VARIAC)
- Tektronix A622 (Strømtang/Probe)



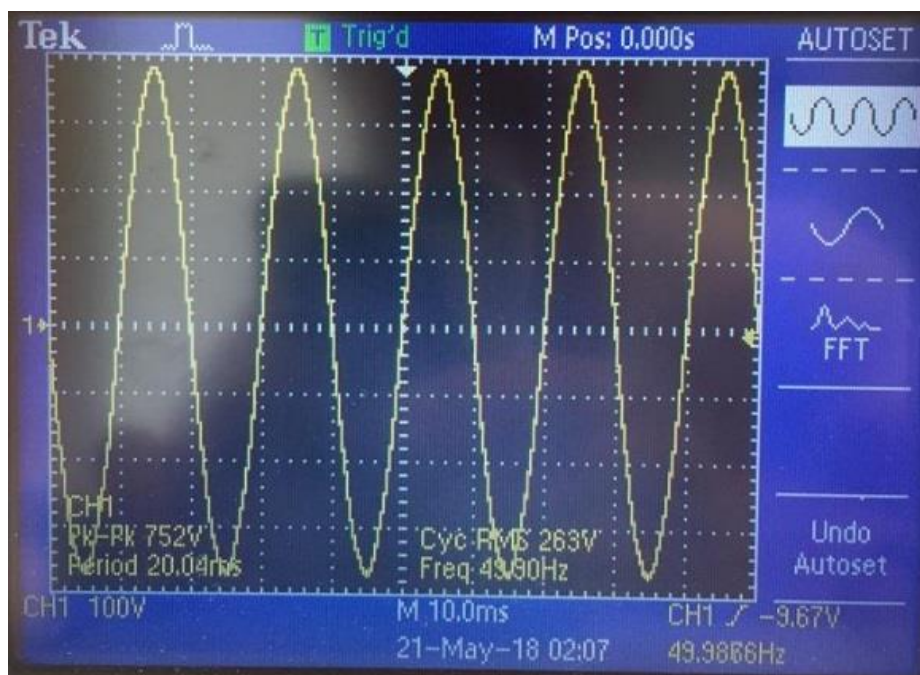
Figur 81 - Oversikt over oppkobling for måling av DOL-kjøring

²¹ Kjøre motoren rett fra nettet uten frekvensomformer

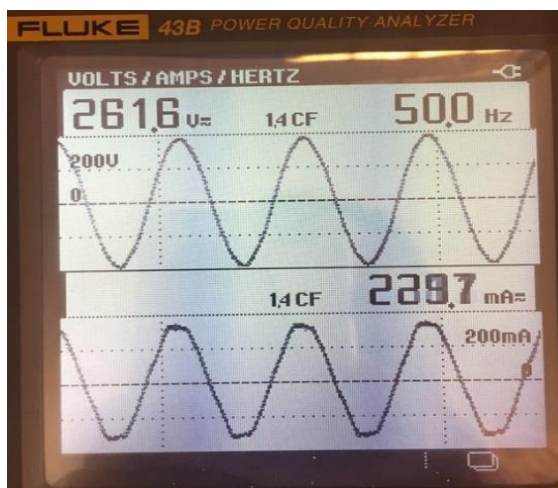
²² Variabel AC spenningstilførsel



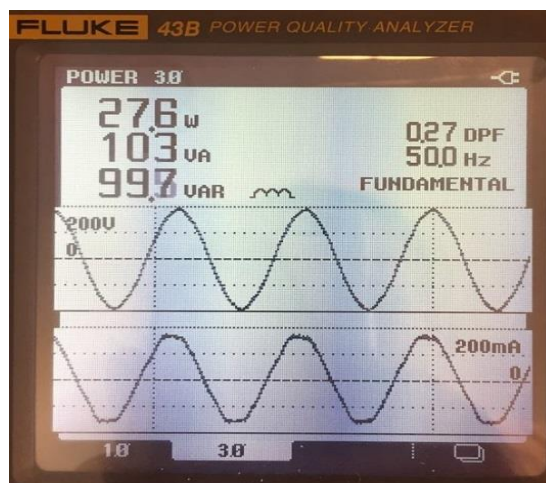
Figur 83 – Fasepenning på motoren DOL



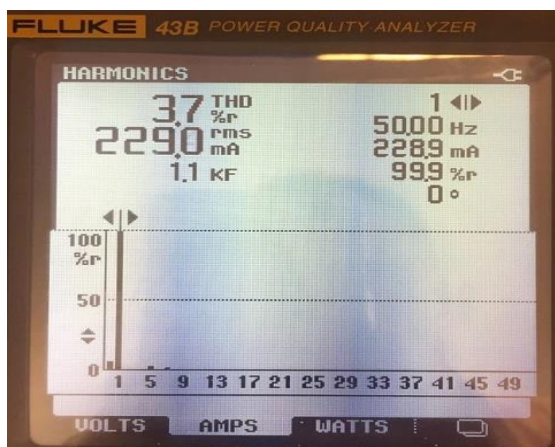
Figur 82 – Linjespenning på motoren DOL



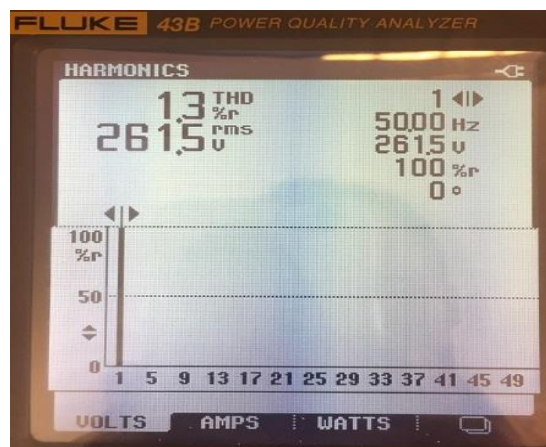
Figur 88 – Volt/Amper/Hertz i DOL



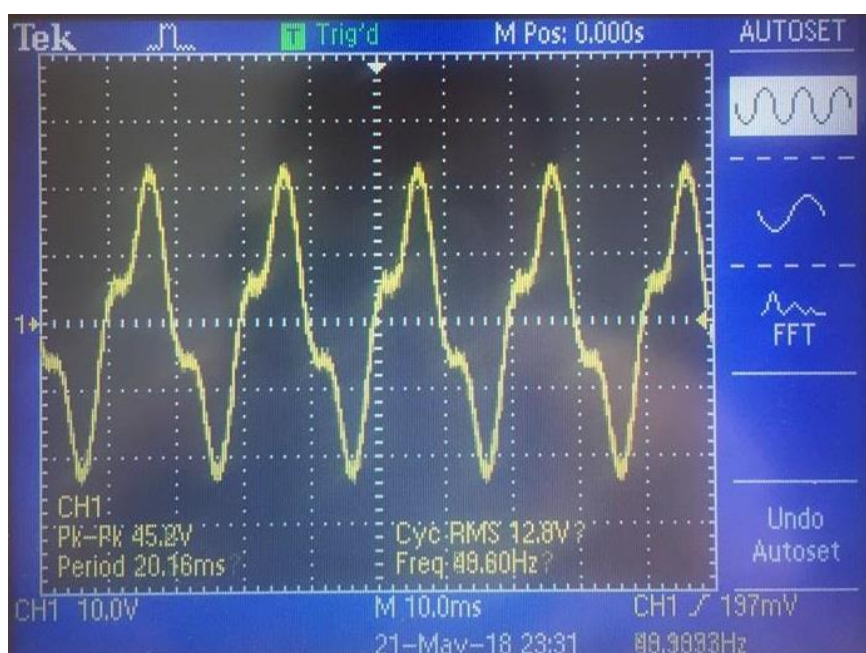
Figur 87 – Effekter i DOL



Figur 86 – Harmonisk strøm i DOL



Figur 85 – Harmonisk spenning i DOL



Figur 84 – Common-mode-spenning i DOL

MERK: Det ble målt 5,8V RMS på common-mode med multimeteret

6 Drøfting

I dette kapitlet drøftes ulike ideer, tanker, beslutninger og resultater som førte til det totale sluttproduktet som tidligere vist i kapittel 3.

6.1 Konsept

I konseptfasen vurderte vi ulike måter å utvikle en undervisningsmodell på. Delkapitlet tar for seg de forskjellige ideene for løsning som ble vurdert.

6.1.1 Komersiell frekvensomformer

Det ble vurdert å bruke en komplett kommersiell frekvensomformer til utvikling av en undervisningsmodell ved å åpne den og spre delene utover et brett. Videre var tanken med dette å overvåke ulike parametere som; temperatur, strømbruk og spenning, samt styring av frekvensomformeren på vanlig vis gjennom et *human machine interface*-program (HMI²³) og/eller med omformerens brukerpanel. En gammel frekvensomformer ble brukt i startfasen for tilegnelse av mer erfaring og kunnskap om funksjon, komponenter og styring av frekvensomformere. Denne ble kjørt med en asynkronmotor via et dataprogram, samt åpnet opp så mye som mulig slik at motorkontrollerkortet og styringskortet kunne bli skilt fra hverandre. Tanken var å sende styringssignal til driver-kortet, men det ble aldri utprøvd. Etter litt testing av omformeren ble dette ansett som en veldig enkel og gjennomførbar løsning på en potensiell modell. På den andre siden ville denne løsningen være for rigid når det gjaldt mulighet for gode visuelle forhold, målinger, og overvåking. Til tross for dette var det ønskelig å ha den som en reserveløsning om fremtidige løsninger ikke ville fungere.

6.1.2 Sammensetting av enkeltkomponenter

Neste idé var å kjøpe inn enkeltkomponenter for å bygge en helt enkel type frekvensomformer bestående av et vekselstrøminntak, likeretter diodebro, et filter, og seks transistorer for vekselretting med selvvalgt frekvens ut til en motor. Styringen var tenkt bestående av Arduino til å sende signaler til transistorene. Det skulle utføres et forsøk på dette og heller ha planen om en kommersiell frekvensomformer som plan B, i tilfellet plan A skulle bli mislykket. Komponentene skulle bestilles fra eBay, og dimensjoneringen av

²³ Brukergrensesnitt på norsk.

komponentene og ledningene var tenkt å bli valgt på bakgrunn av den kommersielle frekvensomformerer som var tilgjengelig. Vurderingen var at hvis samme komponenter og størrelser, eller bedre, som i frekvensomformerer ble brukt, så ville det være bra nok. Før deler ble bestilt ble noen enkle signaler fra et Arduino-kort til noen *Metal-oxide-semiconductor field-effect transistor* (MOSFETs²⁴) utprøvd for å teste ut en enkel kode for transistorstyring. Ved signal-måling ut fra Arduino-kortet med et oscilloskop så koden så ut til å fungere, basert på hvordan signalene så ut. Både faseforskyvningen og frekvensen på signalene så riktige ut.

Under signal-målinger ut fra transistorene ble det oppdaget at testingen ikke ga de resultatene som var forventet, basert på utgangssignalene fra kortet. Problemet viste seg å være at transistorens gate ikke ville holde seg åpen etter den hadde blitt lukket én gang. Det viste seg at gaten ikke klarte å bli helt 0V etter den hadde blitt 5V. Dette betydde at signalet var flytende mellom 0 og 5V. For å løse dette måtte en pull-down-motstand brukes. Oppdagelsen av dette var veldig viktig for videre arbeid med Arduino.

Etter videre diskusjon om deler som var ønskelig å bestille kom det opp en viktig problemstilling som omhandlet et fremtidig perspektiv; nemlig utfordringen med å erstatte deler på modellen hvis noe skulle bli defekt i fremtiden. Mange ulike deler fra forskjellige steder kunne bli vanskelig å oppdrive for overtagere av modellen, i tillegg til at det ville krevd veldig mye dokumentering. Dette, sammen med vanskelighetsgraden av å kunne oppnå et fungerende resultat, førte til at denne fremgangsmåten mot en undervisningsmodell ble revurdert.

6.1.3 Modulbasert modell

Det neste og siste alternativet som ble overveid var å finne en middelvei for det som tidligere hadde vært vurdert. Under denne perioden av konseptfasen ble det brukt om lag en måned før det ble fastsatt hva som var ønskelig å bestille for utprøving. Først ble tiden brukt til å undersøke hva som fantes på markedet. Endel motorkontrollerkort fra ST Microelectronics, som kalles for evalueringskort (eval boards) eller utviklingskort (development boards), ble funnet. Kortene så ut til å være brukervennlige for oppkobling, måling og sikkerhet; utstyrt med koblingspunkter, test/målepunkter, temperaturfølere og sikringer.

²⁴ En type rask transistor, men tåler mindre enn IGBT.

For at anskaffelsesprosessen skulle bli effektivisert for å finne riktig og funksjonelt kort opp mot modellen, ble det sendt e-mail til ulike leverandører av ST-produkter; Farnell, Arrow og selveste ST. Tilbakemeldingene fra leverandørene var til liten hjelp. Enten svarte de ikke, eller så ga de beskjed om at alt av informasjon om kortene stod på ST sine hjemmesider. Etter videre undersøkelse ble kortet STEVAL-IHM023V3 funnet. Kortet ville fungere til AC-motor med en kapasitet på 1 kW. Hovedgrunnene til at valget havnet på dette kortet var fordi det hadde egendefinerte testpunkter som var klare til oppkobling for måling, nok effekt, god sikkerhet og kortslutningssikkerhet mellom transistorene.

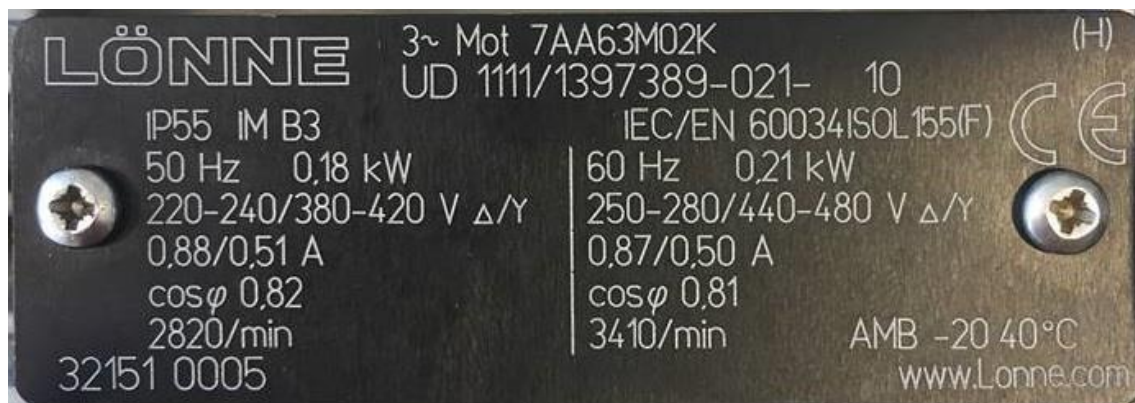
6.2 Materiell

Her vil det drøftes hva slags type utstyr som ble vurdert, prøvd ut, og valgt basert på oppgavens mål og hensikt.

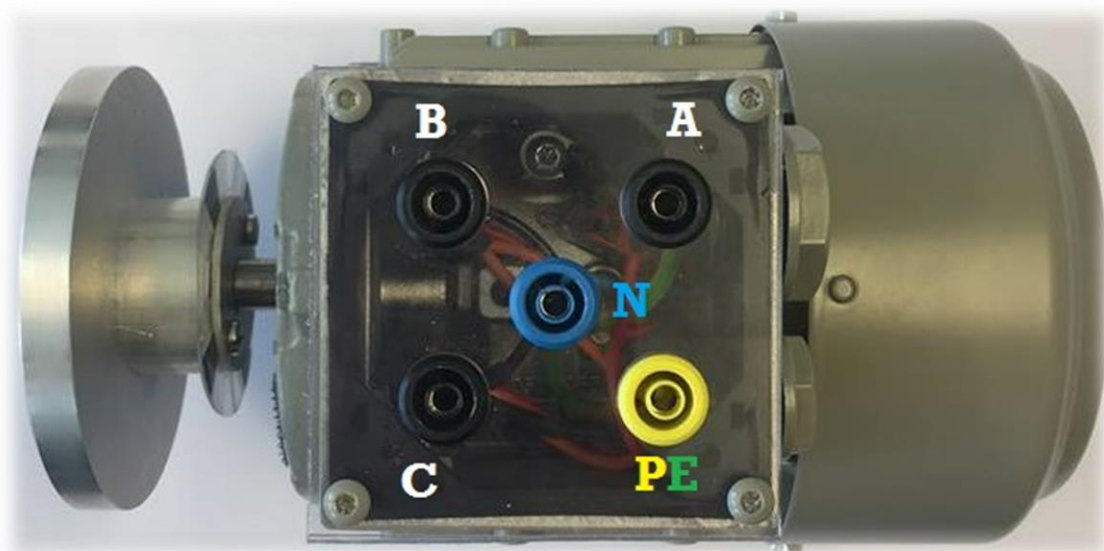
6.2.1 Valg av motor til modell

Motor som har blitt valgt til å følge modellen er en enkel AC induksjonsmotor fra Lönne på 180 watt. Vi har koblet motoren i stjerne for å redusere strømtrekket, og for å skape et nøytralt punkt. Tidligere var det montert et større svinghjul på motoren som gjorde det enkelt å se når vi endret dreieretning. Det var mulig å se at motoren først bremses ned, for så å begynne å spinne den andre veien. Etter noen tester der vi endret dreieretning og stoppet motoren flere ganger, utløste glassikringen inne i modellen slik at tilførselen ble brutt.

For å skape en robust modell, reduserte vi treghetsmomentet på skiven ved å redusere diameteren. Vi endte opp med en diameter på 10cm, slik at skiven ikke ville komme i kontakt med bordet. På denne måten løste vi problemet, og brukerne kan nå stoppe og snu dreieretningen uten å bekymre seg for at sikringen vil smelte. Man kan fortsatt tydelig se at motoren endrer dreieretning, men naturligvis noe raskere.



Figur 89 - Motorskilt



Figur 90 - Motorkobling

Nøytrallederen er ført frem slik at vi kan måle common-mode-spenning mellom N og PE. Motor lokket er byttet ut med plexiglass for å kunne vise mer av innsiden av motorekoblingene. Vi har også valgt å bruke berøringssikre bananplugger for å gjøre målingene enkle og sikre.

6.2.2 Motorkontrollerkort

Kortet STEVAL-IHM023V3 ble valgt til å kjøre hovedstrøm til motor.

I tillegg til 6.25A sikringen (i følge dokumentasjonen, men som faktisk er på 6.3A) er det blitt lagt inn en egen smeltesikring på 6A før J1-tilkoblingen som er mye lettere å bytte ut om overstrøm skulle bli tilfellet.

6.2.3 Kontroll og styring

De ulike kontrollenhetene for kontroll og styring som ble vurdert til modellen

6.2.3.1 Wago

En produktserie som ble vurdert til styring, måling og overvåking var PLS fra det tyske firmaet Wago. Grunnen til at dette ble vurdert først var fordi dette var kjent fra tidligere undervisning i Wago-PLS året før, som resulterte i god innsikt i alt det grunnleggende. Noen lånte Wago-moduler på Sjøkrigsskolen ble brukt for testing; en effektmåler-modul (som kunne måle strøm, harmoniske signaler, og effekt), og selve PLS-modulen med CPU. Det oppstod oppstartsproblemer med effektmodulen. Hjelp med problemet ble løst av å tilkalle en kontaktperson fra Wago. Personen fra Wago løste problemet, og testing av utstyret kunne fortsette. Samtidig som dette utstyret var under testing sendte vi mail til en konsulent i Tyskland for å spørre om Wago hadde modul for PWM-signaler eller raske digitale signaler. Etter hjelp fra Wago får vi resultater med effekt-modulen, men svar på at de ikke har PWM-modul til vårt bruk. Dette resulterte i at vi gikk helt bort fra Wago som kontrollenhet.

6.2.3.2 STMicroelectronics

For å finne ut mer om ST sine mikrokontrollerkort ble det brukt en annen metode enn tidligere. I stedet for å sende e-mail, ble det opprettet en live chat med en konsulent fra Arrow. Ut ifra chat-samtalen med Arrow ble det gitt hjelp til å finne ut av hvilke mikrokontrollerkort som ville fungere sammen med motorkontrollerkortet, og at de ville være gode nok til ønskelig bruk opp mot modellen. ST sine mikrokontrollerkort for styring av motorkontrollerkortene hadde mulighet for bruk av forskjellig programmeringsplattformer²⁵.

²⁵ Dataprogram som brukes til å skrive koder for å programmere mikrokontrollere.

Det ble bestilt fem mikrokontrollerkort og lastet ned plattformer og så instruksjonsvideoer om hvordan de kunne bli brukt til å realisere koding og opplasting. Plattformene inneholdt flere programeksemples som kunne brukes opp mot forskjellige kort, men disse programmene var kun ment for DC-motorer, ikke AC-motorer. Dette resulterte i at eget program måtte utvikles. Det tok lang tid før vi fikk kortene, og i mellomtiden lånte Arduino-kort testet ut på skolen. Da kortene ankom hadde et velfungerende program på motorkontrollerkortet allerede blitt utviklet. Ett av kortene som ble anskaffet viste seg å være kompatibelt med Arduino IDE, i tillegg til at det var mye raskere enn selve Arduino-kortet.

For å optimalisere ble det da tenkt å bruke dette kortet, NUCLEO-F446RE Development Board. Etter undersøkelse om hvordan innstillingene til Arduino IDE for bruk av dette kortet skulle være, ble det testet det med et enkelt LED-blinkeprogram. Dette fungerte, men på den tiden ble valget å gå bort fra kortet. Årsaken til dette var at det ble ansett som mindre sjans for brukere å gjøre noe feil. Det viste seg lettere å få tak i Arduino-kortene, samt at internettmiljøet for programeksemples og hjelp var større når det gjaldt feilsøking, kobling og drift av Arduino i forhold til Nucleo.

6.2.3.3 Arduino

Veldig tidlig i oppstarten av bacheloren ble det lånt Arduino Uno, Arduino Mega, ekstra komponenter som knapper, lysdioder, skjerm og potmeter, samt instruksjonsbok. Dette ble benyttet til å bli kjent med produktet og noen av dets muligheter før det ble tatt et valg om å benytte det til konstruksjon av modellen. Etersom valget havnet på å bruke Arduino ble det kjøpt inn en Arduino Mega og en Arduino Uno på Kjell & Company i Bergen sentrum. Arduino er åpen kilde og et godt springbrett for elever å lære seg koding. Vi så også at ST sine kontrollkort fort ville bli for avansert å programmere. Vi ville hatt muligheten til å laste inn ferdig programvare med FOC, men læringsutbyttet til programmeringen ville blitt redusert.

6.3 Software

6.3.1 Generelt

Det var relativt enkelt å sette seg inn i Arduino sin plattform, selv om det var helt ukjent. Grunnen til dette var antakeligvis at C++ og PLS-programmeringen gjennom utdanningen la et godt grunnlag for å forstå programmering, samt at mye av språket var likt. Det var noen store forskjeller når det gjaldt enkelte funksjoner i språket. Dog, det mest nødvendige av funksjoner for programmeringen som er relevant i denne oppgaven, ble funnet ved instruksjonsvideoer og forum på internett.

6.3.2 Six step-kode

Den første koden som ble skrevet var rekkefølgen og timingen for tenning av de seks transistorene slik at spenningen ut kunne bli god nok til å drive motorens rotor rundt. Dette var hovedfokuset når det gjaldt programmeringen, da det var veldig viktig å få en motor til å gå rundt. Grunnen til at valget havnet på å bruke six step-metoden var fordi den ble ansett å være enkel å forstå, kode og feilsøke. Første utkast av koden ble skrevet ganske tidlig på Arduino, mens bestillingene av de andre kortene var underveis.

Det skal nevnes at en kode for PWM-styring også ble utforsket og prøvd, men det viste seg at den så meget komplisert ut og vanskelig å forstå. I tillegg var det ønskelig å produsere en kode selv slik at forståelsen for den kunne bli optimal. På den annen side så ville PWM åpnet opp for å teste du/dt filter. Det kunne vært interessant å implementere begge programmene slik at man kunne sammenligne de ulike styringene, og sett på fordeler og ulemper i forhold til harmoniske strømmer og common-mode-spenninger. Slik kunne vi også ha gitt brukerne en bedre forståelse for hvordan de ulike filtrene påvirker systemene og når de bør anvendes.

Målinger med oscilloskop var helt nødvendig for å kunne se om koden var god eller ikke. Erfaringen med bruk av oscilloskop var liten, slik at det ble en egen prosess på å lære seg dette godt nok til å utføre målingene. Måten dette ble gjort på var å prøve selv, og til slutt spørre lærer for å fylle inn kunnskapshullene. I starten ble det brukt oscilloskop med prober som ikke var isolert fra jord. Dette skapte feil resultat da vi skulle måle signalspenningen. Det som ble målt var mellom PE-jord og utgang på Arduino istedenfor mellom Arduino referansejord 0V og utgang. Etter byttet til isolert oscilloskop ble resultatet som forventet.

Da disse signalene ble koblet opp til noen MOSFETs for å teste spenningen produsert av koden, ble batteriet som ble brukt veldig varmt. Målingene på MOSFETene viste at de alltid lå inne, som om de alltid hadde 5 V på gaten. Etter kontinuitetsmålinger på flere MOSFETs for å se om de var defekte, uten at de var det, ble en undersøkelse av Arduino opp mot MOSFETs foretatt. Fenomenet med flytende signal og pull-down-motstand var på dette tidspunktet helt ukjent. Etter dette ble funnet ut av og koblet opp, fungerte signalene som de skulle – en svært viktig oppdagelse for evnen til å bygge en funksjonell frekvensomformer.

Måten motoren stopper på er veldig brått på grunn av kortslutning av dens viklinger. Dette kan unngås ved å bruke egne signaler til alle seks IGBTene, men siden svinghjulet har lite treghetsmoment vurderes det at fordelene ved en enklere kode med tanke på læring veier mer enn å skåne motoren for belastningen ved å fungere som bremsemotstand.

Vi vurderte om vi skulle bruke *timere* til koden istedenfor å bruke delay-funksjonen for hver 60°. Å bruke interrupt-funksjoner inne i loopen er egentlig en ganske dårlig løsning, og det er mest sannsynlig derfor frekvensen vår er lavere enn innstilt frekvens. Til tross for unøyaktig frekvens og dårlig kodeskikk, mener vi at læringsutbyttet ved å bruke delay-funksjoner vil være større. Både for å enkelt kunne forstå kildekoden, men også for å forstå helheten på teori, koding, simulering og praktiske målinger.

6.3.3 Frekvensstyring

Hele poenget med en frekvensomformer er å kunne styre en motor sin hastighet og moment, så denne delen av koden var også svært viktig å få til. Koden for dette ble skrevet like etter koden for six step. Det var ønskelig å kunne bruke et trimbart analogt hjul til å bestemme frekvensen, istedenfor å bruke et digitalt panel eller å måtte gå inn i koden for å forandre verdien hver gang frekvensen skulle endres. Hele poenget var å tilrettelegge muligheten for å kunne se endringen på motoren kontinuerlig, ved både raske og trege endringer.

Et potmeter som fulgte med Arduino Starter Kit ble først prøvd ut med den enkle forholdstallfunksjonen Map() hvor verdispennet inn gir et viss verdispenn ut. Måten denne prosessen artet seg på var ved å koble 5V fra Arduino ut til potmeteret og videre til en analog inngang slik at det var mulig å måle minste verdi (med mest motstand) og største verdi (med minst motstand). Verdiene Arduino leste av på den analoge inngangen var først 0 som minst, og 1023 som størst. Dog, etter undersøkelse om kortets begrensninger,

ble det oppdaget at hver inn-/utgang hadde en I_{max}^{26} på 60mA. Kalkulasjoner viste at hvis potmeteret hadde et punkt hvor det ga veldig lav, kunne det skape problemer. For å løse dette ble det valgt en motstand på 100Ω i kretsen med potmeteret for å ta hensyn til I_{max} . $I = U / R^{27} \rightarrow I = 5 / 100 \rightarrow I = 0,05A \rightarrow I = 50mA \rightarrow OK$

Dette gjorde at verdispennet inn endret seg til 0-1010, fra 0-1023. Da modellen skulle settes sammen ble det kjøpt inn et finere og større potmeter.

På utgangen av funksjonen ble det prøvd alt fra 10 (laveste) til 70 (høyeste). Grunnen til at det ikke ble prøvd høyere eller lavere frekvens var på grunn av motoren. Motoren som ble testet var selvventilerende og trenger høyere turtall for å kjøles. Noe særlig høyere enn 70Hz ble ikke prøvd på grunn av generell ivaretagelse av motoren, siden merketurtallet er med 50Hz.

6.3.4 Motortilstander

Det ble vurdert startknapp, stoppknapp, dreieretningsknapp og nødstopppknapp. Nødstopp ble vurdert på grunn av noe potensielt kan feste seg i motoren. Da materiell skulle kjøpes inn var det vanskelig å finne en nødstopppknapp som var liten nok til modellen samtidig som den hadde kvalitetene til hvordan en nødstopp skal være. Det finnes en knapp ved strøminntaket på modellen som på en måte fungerer som nødstopp siden den kutter hovedstrømmen til modellen. Ut ifra dette ble valgt å droppe en nødstopp som ville passet dårlig til modellen.

Stoppknappen ble kjøpt inn, men vurdert bort fra modellen ettersom både start- og stoppknappen hadde to stillinger; på eller av ved hvert trykk. Ettersom motoren ville stå av når startknappen var ute, og på når knappet var inne, var det ansett som lite nødvendig å ha en separat knapp til å skru av med. Dette kunne ha gjort det mer forvirrende.

Etter testing og tilbakemelding fra en elev som testet modellen ble det nødvendig å utvikle en indikasjon på om startknappen var inne eller ute, da det var umulig å se dette på knappen i seg selv. Resultatet ble en liten grønn LED rett over bryteren som lyser når knappen er inne og motoren kan få spenning.

²⁶ Maks strøm den tåler kontinuerlig.

²⁷ Ohms lov.

Med samme grunn har også disse knappene en motstand 100Ω i serie med seg for å redusere strømmen i kretsen når de ligger inne. Motstandene er loddet rett på ledningene med krympestrømpe over.

6.3.5 Måling og overvåking

I utgangspunktet var det mange typer faste målinger som var ønsket å implementere i modellen. Målinger og beregninger som strøm, lagerstrøm, spenning, akustisk støy, vibrasjon/resonans, temperatur, harmoniske signaler, rpm, frekvens, slipp og moment var ønsket på modellen. Mye av dette ble mer eller mindre utforsket og/eller prøvd. Det som endte opp implementert i modellen var rpm, frekvens og DC-buss-spenning. Hovedgrunnene for at de andre målingene ble utelatt var på grunn av tid, kompleksitet, tilgjengelighet, og relevans.

6.3.5.1 LCD-skjerm

Til å begynne med var det usikkert om en skjerm i det heletatt skulle benyttes. Vanlige oscilloskop var det første som var tenkt, etter hvert også digitale skop. Samtidig som skop ble undersøkt, ble også tester for å prosjektere verdier på skjerm også prøvd ut. Først ble det utprøvd på en to-linjers LCD-skjerm fra Arduino Starter Kit.

På Arduino sine nettsider ble flere koblinger og koder for LCD-skjerm funnet. Noen av dem mer vanskeligere å forstå enn andre. Det viktigste var å finne ut av hva slags innganger skjermen hadde – hva som var til tilførsel, og hva som var til signaler som dikterte tegn for å kunne skrive. Etter dette var det enkelt å eliminere innganger til de som måtte bli fokusert på for å få til ønskelig resultat, altså å skrive ut verdier. Etter litt prøving med hjelp fra nettkilder av andre som hadde koblet opp dette før, ble målet om fremvisning av tekst en suksess.

Digitale oscilloskop sammen med Raspberry Pi²⁸ var veldig nære på å bli kjøpt inn, men på grunn av tid og kompleksitet med tanke på forståelse, dokumentasjon og brukervennlighet, havnet valget på LCD-skjerm til Arduino. For å kunne vise de tre målingene; rpm, frekvens og DC-buss-spenning ble det kjøpt inn en fire-linjers LCD-skjerm istedenfor den med to linjer. Skjermen ble nedfelt inne i pleksiglasset som innrammer modellen.

²⁸ En slags mini-computer på et lite kretskort.

6.3.5.2 Turtall (RPM)

Primærløsningen var IR-sensor og en sirkulær skive med en viss oppløsning (så høy som mulig) som stod som primærløsning da ideen om å måle turtall var ute. Kunnskap om å få til dette var allerede hentet fra et tidligere gjennomført kurs i PLS. Det var kjent at denne måten å gjøre det på ville være grei å utføre med suksess. Databladet for typen IR-sensor Sjøkrigsskolen allerede hadde ble søkt opp for å finne koblingsskjema, og den sirkulære skiven var det allerede en mal på.

Koden for dette derimot, var litt mer komplisert å få til i Arduino enn antatt på grunn av mangel på kunnskap om hvordan få til en funksjon. Istedenfor å bruke funksjonen som var ønskelig å bruke, ble det heller en mer komplisert kode med klokker.

Oppløsningen på telleskiven var ønskelig å ha så høy som mulig for å få rpm-resultatet mest mulig riktig til enhver tid. På grunn av koden som regner ut rpm ble prosessen for treg til å få riktig rpm ved 50Hz og oppløsning på 10 (10 hull. Det prøvd med 6, hvor 43Hz og oppover var den hastigheten det da ble feil på. For å få rpm til å bli riktig ved minst 50Hz måtte oppløsningen ned til 4. Med 4 var det godt nok innenfor, og det var det skiven endte opp med. Det skal nevnes at en 100Ω motstand også ble implementert gjennom IR-sensoren for strømreduksjon.

6.3.5.3 Innstilt frekvens

Det kan være interessant å se den innstilte frekvensen opp mot turtallet på akslingen til motoren. Hvis motoren driver ulike laster kan det være mulig å se turtallet forandre seg i forhold til frekvensen på tilførselen.

En test hvor innstilt frekvens på modellen opp mot den frekvensen motoren får, viste at det er en liten forskjell. Forskjellen er en reduksjon fra innstilt til opplevd frekvens på gjennomsnittlig 3.1%. Dette ble regnet ut ved å multiplisere alle forholdstallene av forskjellen for hver femte endring av frekvens fra 20 til og med 50Hz og delt med antall målepunkt (7). Dette mener vi er en ganske liten og relativt ubetydelig verdi opp mot hensikten til vår oppgave, men at det er greit å bemerke seg det.

Tabell 4: Resultater for innstilt Hz mot opplevd Hz

Innstilt Hz	Opplevd Hz	Forhold
20	19.72	0.986
25	24.60	0.984
30	29.46	0.982
35	33.34	0.952
40	38.17	0.954
45	43.46	0.966
50	47.77	0.955
Gjennomsnitt forhold:		0.969
Gjennomsnitt i prosent:		96.9 %
Reduksjon i prosent:		3.1 %

6.3.5.4 Spenning på DC-buss

Det å vise spenningen på DC-bussen til enhver tid var noe som enkelt kunne bli gjort ved å bruke en av utgangene på motorkontrollerkortet inn på Arduino. Utgangen ga en spenning mellom 0-5V alt ettersom hva spenningen på DC-bussen var. Til å starte med var det uvisst hvordan skalering det var på utgangen i forhold til DC-bussen. I databladet for motorkontrollerkortet ble det ikke funnet noe informasjon om det. På grunn av dette ble det foretatt en rekke målinger med en variac.

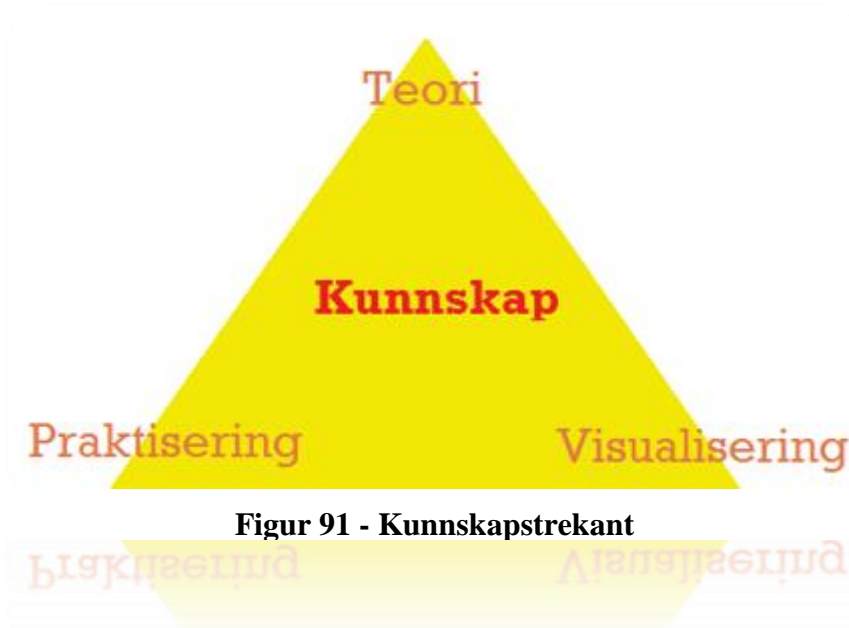
Resultatet av spenningen som vises på skjermen er ikke nøyaktig, men det er en relativt god verdi som en indikasjon. Det at man kan se spenningen fra kondensatorene i DC-bussen prøve å holde på energien når tilførselen til modellen kuttet er en interessant funksjon å ha med i modellen. Med tanke på sikkerhet er det fordelaktig å ha en indikasjon på om det fortsatt er spenning lagret på den eller ikke før berøring finner sted. Vurdering av egensikkerhet vil være enklere. I den tekniske manualen for motorkontrollerkortet står det ingen spesifikk tid for når det er greit å være borti motorkontrollerkortet etter frakobling. Det står kun at man ikke skal ta borti kortet etter frakobling på grunn av kondensatorene som lagrer energi (ST 2014, 7).

6.4 Ferdigstilling

Ved å få en bedre formening om vår modell oppfylte kravene vi stilte innledningsvis, valgte vi å bruke en testperson, som vi anså som en kunde som skulle validere det ferdige produktet. Vi var selv veldig fornøyd med hva vi hadde konstruert og anså modellen som god nok. Til tross for dette så ble vi oppmerksomme på noen mangler av testpersonen. Som følge av det måtte vi utbedre modellen for å gjøre den mer brukervennlig. Det ble montert inn en LED som indikerte om startbryteren var aktivert. Med bakgrunn i dette så ser vi at selv om vi var veldig godt fornøyd, så er det alltid lurt å få noen utenforstående til å teste det ferdige produktet. Det er tross alt ikke vi som konstruerer modellen som skal bruke den.

Den andre testperson hadde som oppgave å evaluere læringsutbyttet av oppgaveheftet. På bakgrunn av vurderinger rundt tilbakemeldinger fra oppgavetesting var det ønskelig å utføre tiltak som kunne forbedre oppgaveheftets potensiale for å tilrettelegge for læring. Ut ifra tilbakemeldingene ble oppgaveheftet utbedret med flere bilder og mer eksplisitt koordinering i oppgavenes gjennomgang.

6.5 Læringsutbytte



Vi hevder at ved å kombinere teori, praksis og visualisering så vil målgruppen få en bredere forståelse og et mer varig læringsutbytte.

Læring- og kunnskapsteori sier noe om at ulike type mennesker best lærer og tilegner seg kunnskap gjennom ulike metoder. I vår bacheloroppgave har målet vært å få med flest mulige metoder og dimensjoner som kan bidra til læring med det vi har sammenfattet og konstruert av modell, simuleringsprogram, og oppgavehefte. I forsøk på å tilfredsstille teoretikeren inneholder oppgaveheftet en del teori om kraftelektronikk som er relevant for modellen. For praktikeren er det de praktiske oppgavene opp mot kjøring og måling av modellen med motor som vil være mest attraktivt. Det nærmeste en kommer en ren visualisering rundt dette er å benytte seg av simuleringsprogrammet OrCAD og det som er lagt ved av oppgaver for dette.

Det skal nevnes at vi ikke har prøvd å skille ut disse tre typer elever, men at fokuset har vært å bruke disse tre dimensjonene til å skape en form for synergi mellom dem for alle elever. Noen vil antageligvis være mer motivert og mottakelig for det ene eller andre typen, noe som vi selv vet av egen erfaring etter flere år med skolegang. Dog, jo flere områder av faget som blir utforsket, jo mer forståelse vil kunne bli tilegnet den enkelte.

Ofte vil det være svært nyttig å ha en viss form for intuitiv forståelse av noe. Det å ikke måtte sette seg ned med penn og papir for å finne ut hvorfor noe er feil, eller hva årsaken

til en feil er, kan føre til at situasjoner som er avhengig av raske løsninger suksessfulle. Et eksempel på dette i maritim (forsvars)sammenheng kan være at fartøyet man jobber på er i en (krig-)/krisesituasjon har problemer, og at man ved intuitiv kunnskap raskt nok klarer å peke ut årsaker til feil eller defekte deler før et dramatisk utfall skjer.

Med denne modellen mener vi at elever har svært gode muligheter til å få et godt læringsutbytte, men at mye står på eleven selv slik det gjør med alt annet.

6.6 Målte resultater

6.6.1 Generelt

Målingene som ble utført ble gjort under normale forhold og omgivelser i en elektrolab på Sjøkrigsskolen. Med dette menes at luftfuktighet, temperatur og evt. andre faktorer ikke ble tatt særlig hensyn til. Det tas forbehold om helt nøyaktige resultat da forskjeller på motorvikling-temperatur vil forårsake ubalanserte målinger opp mot hverandre. Det ble forsøkt å holde motoren på om lag samme temperatur som rommet ved å ikke kjøre den for varm før målingene, bortsett fra temperaturtesten som ble utført.

Med tanke på omstendighetene instrumentene målte i er det viktig å være noe kritisk til måleresultatene. Spenning- og strømresultatet av en six step er nokså forskjellig fra en ren sinus fra nettleverandør. Det bemerkes at Fluke 43B sitt måleområde for **Power display** (watt, VA²⁹, VAR³⁰, og PF³¹), samt **Harmonics display** (volt, strøm, frekvens) gjelder for 40-70Hz. Vi var usikker på om måleinstrument i det hele tatt var kompatibelt med six step-spenning og om vi kunne stole på resultatene vi fikk med frekvenser under 40Hz. Etter studering av databladet for måleinstrumentet fant vi ut at det var bedre å kontakte produktutvikler direkte, for å være på den sikre siden. Etter en mailkorrespondanse med Fluke fikk ble det bekreftet at instrumentet kunne brukes. Strøm-, spenning- og skopfunksjonen vil gi gode måleresultater fra 10Hz til 15kHz.

Vi må også ta høyde for at instrumentene har et visst spenn når det gjelder nøyaktighet. En av de mest sannsynlige årsakene til feil i fremtidige målinger anser vi at vil være

²⁹ VA – Voltamper (Tilsynelatende effekt)

³⁰ VAR – Voltamper reaktiv (Reaktiv effekt)

³¹ PF – Power Factor (Effektfaktor)

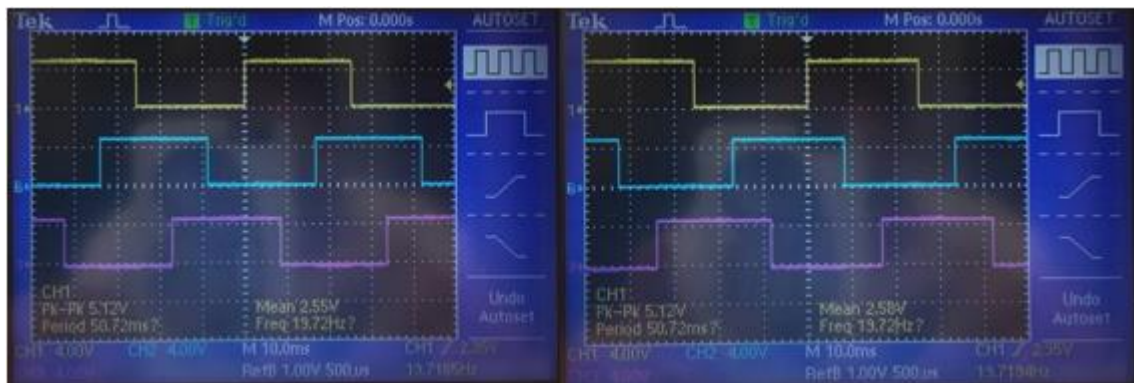
kalibreringen av strømtangen. Det er lett å glemme samtidig som at det er lett at utilsiktede berøringer stiller på kalibreringshjulet.

Om ikke resultatene isolert sett skulle være helt riktige, så mener vi at det ikke er så nøye opp mot vår oppgave, så lenge måleresultatene bidrar til interessante funn. Oppdagelser som verdiforskjeller ved ulikt spennings- og driftsmønster, forventede/uforventede verdier, og grenser på opphetning/nedkjøling anser vi som interessante.

6.6.2 Gate-signaler

Resultatet av målinger for gate-signaler viser at rekkefølgen til to av signalene endrer posisjon i tidsplanet på grafen når dreieretningen endres mellom dreieretning A og B. I elektromotorer vil rotoren endre dreieretning ved bytting av to vilkårlige faser til motorviklingene på grunn av magnetiseringsfeltets dreieendring.

Det er interessant å se at bytting av to signaler til styringen av vekselrettingen også fungerer som bytting av to faser til motoren for å oppnå dreieending som vi tidligere er kjent med.



Figur 92 – Sammenligning av gate-signaler ved dreieretning

6.6.3 Strøm

Når det gjelder strømtrekk er det interessante å merke seg hvor stor forskjell det er ved forskjellige frekvenser og tilstander motoren drives i. Ved 50Hz registrerer instrumentet startstrøm til å ligge på 1A peak-verdi, dreieendring til 3,2A peak-verdi, og stopp til 1,7A peak-verdi. Kontinuerlig strømtrekk varierer ved ulike frekvenser motoren kjører på. Noen resultater endrer seg drastisk ved endring av frekvensen, mens andre ikke. For eksempel fra 50Hz til 35Hz endrer strømmen seg kun fra 0,266A til 0,397A, mens fra 35Hz til 20Hz endrer den seg fra 0,397A til 1,212A.



Figur 93 – Sammenligning av høyest og lavest kontinuerlig strømtrekk (motordrift fra modell)

Vi tror grunnen til at vi måler så høye strømmer på lavere turtall er fordi vi ikke har konstant U/f forhold. Spenningen holdes konstant uansett hvilken innstilt frekvens vi kjører motoren på. Ved å redusere frekvens vil reaktansen synke, siden motorens viklinger regnes som spoler, og dermed vil impedansen i motoren synke og strømmen øke. Vi tror at ved å endre styringen til PWM med konstant U/f forhold kan problemet med økt strøm på lavere turtall løses. Det er litt uheldig at motoren trekker så mye strøm over nominell verdi på lavere turtall, siden den er selvventilerende, og lettere vil bli varm.

6.6.4 Spenning

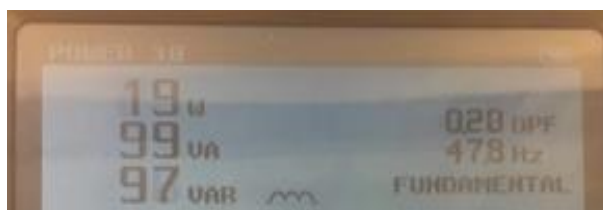
Common-mode-spenningen var vanskelig å få fin på oscilloskopet, men det var fortsatt mulig å få lesbare resultater.

Motordrift fra modell viste 252V peakverdi og 138V RMS ved 50Hz, mens resultatene fra DOL viste 22,6V peakverdi og 12,8V RMS. Disse resultatene viser et veldig stort

sprik mellom common-mode-spenning ved motordrift fra modell og DOL. Det er interessant å se at en type kilde for spenning kan skape så store forskjeller fra en annen kilde, og vite at den ene kan forårsake store skader som resultat av lagerstrømmer.

6.6.5 Effekter

Resultatene av effektmålingen viser en stor ubalanse mellom aktiv effekt (W) og reaktiv effekt (VAR). Vi vurderer dette til å ha årsak med at det ikke er noe særlig last på motoren, kun en lett sirkulær skive.



Figur 94 – Effekter ved 50Hz

6.6.6 Harmonisk støy

Siden Fluke 43B bare kan måle harmonisk på frekvenser mellom 40-70Hz, har vi kun foretatt måling ved 50Hz. Ved six step-metode er det først og fremst 5. og 7. harmoniske strømmer som vil utgjøre den største kilden til harmonisk støy. Ved bruk av nettanalysatoren kommer dette veldig godt frem, der vi har ca. 50% 5. harmonisk. Det hadde vært interessant å satt på et sinusfilter for å se hva som ville skjedd med de harmoniske forstyrrelsene.



Figur 95 – Harmonisk strøm ved 50Hz

6.6.7 Temperatur

Ut ifra de foretatte temperaturtestene på motoren fikk vi belyst hvor viktig det er med tilstrekkelig turtall på selvventilerende motorer.

Ved drift av motoren i fem minutter på 20Hz steg temperaturen til ca. 50°C på utsiden av motorens kapsling, men det er uvisst hvilke temperaturer som var inne i motoren. På grunn av dette ble det valgt å kjøre motoren opp i 50Hz igjen for å unngå skader på viklinger og isolasjon. Dette tiltaket gjorde at temperaturen sank ned mot 40°C etter kun 5 minutter. Om kobberisolasjonen svekkes og blir sprø kan det raskere føre til overslag og kortslutning. Testene ved lavt turtall viser da at temperaturen øker på grunn av dårligere selvventilasjon og høyere strøm, og at det er hensiktsmessig å kjøre modellen i høyere turtall over lenger tid.

6.6.8 Hastighet på aksling

Resultatene viser på det meste 9 rpm avvik, noe som ikke er så veldig mye siden det er rotasjoner over et helt minutt. Dette tilsvarer en feilmargin på $(360^\circ \cdot 9) / 60 = 54^\circ$ per sekund.

Det skal merkes at turtallet ved målingen under 50Hz ligger over nominelt turtall som er oppgitt på merkeskiltet til motoren (2820/min). ELMA-instrumentet målte 2844/min, mens IR-sensoren med program målte 2850/min. Vi vurderer at årsaken til dette ligger i at motoren kjører utenfor nominell drift, altså ingen betydelig last.

Ut ifra disse resultatene mener vi at nøyaktigheten til modellens rpm-måling er god nok til dens bruk og demonstrasjoner.

6.6.9 Sammenligning mellom DOL og modell

Som vist gjennom delkapittel 6.6 var det interessante resultater mellom DOL og modell-drift. Vi vurderte også å sammenligne med en annen kommersiell frekvensomformer. Dette ville trolig vært en mer interessant sammenligning for hvor god modellen er opp mot et profesjonelt produkt. Da kunne vi for eksempel ha sett på forskjellen mellom de harmoniske signalene og common-mode-spenningene. Ut ifra dette kunne hovedtrekk i forskjeller mellom six step og PWM blitt kartlagt.

7 Konklusjon med anbefaling

I denne oppgaven har vi tatt for oss konstruksjon av en undervisningsmodell for motorstyring i form av en frekvensomformer med tilhørende simuleringsprogram og oppgavehefte. Hovedfokuset har vært å gjøre modellen best mulig egnet til undervisningsbruk på bakgrunn av konkrete krav som; berøringssikkerhet, gode visuelle forhold, lett å frakte, frekvensstyring og turtallsmåling.

Gjennom planlegging, målsetting, undersøkelse, utprøving og testing har det oppstått mange konseptuelle tankeprosesser og praktiske utfordringer. Mulighetene for valg av materiell, program og praktiske løsninger har vi sett det finnes svært mange av. Dette har både hemmet og fremmet prosessen av oppgaven. Det har blitt brukt mye tid på å finne ut hva som kan fungere, men til tross for dette har det ved gode valg gjort at implementering og konstruksjon har gått raskere. Innspill fra testpersoner og egne verifiseringer opp mot modell og oppgaver, både underveis og til slutt, har bidratt til et bedre sluttprodukt.

Alle målinger, tester og sammenligninger rundt modellen viser svært lovende resultater opp mot kravene som i utgangspunktet var satt. Etter test av sluttproduktet ser vi at den har evnen til å starte, stoppe og kontinuerlig kjøre en trefase AC-motor med varierende selvinnstilt frekvens, måle turtall og spenning på DC-buss, samt endre dreieretningen til motoren. I tillegg til dette tilbyr konstruksjonen visuelle, gode, og sikre forhold for signalmålinger. Medfølgende oppgavehefte inneholder blant annet teori som er relevant for kraftelektronikken i modellen og oppgaver av teoretisk, visuell og praktisk karakter. Simuleringsprogrammet tilbyr kunnskap om hvordan en slik drive kan fungere; hvilke elementære komponenter de inneholder, dens egenskaper, fordeler og ulemper. For å tilrettelegge for trygg og enkel transport av modellen og dens tilleggsutstyr valgte vi å plassere dette i en plastkoffert med isolasjonsskum.

Med dette mener vi at den konstruerte modellen egner seg godt til demonstrasjoner og laboratorieundervisning med begrenset antall brukere på samme tid.

Som videre anbefaling oppfordres det til at modellen med alt som følger med benyttes til undervisning og laboratoriestudier, og at neste generasjon som ønsker å utvikle modellen implementerer PWM-styring med konstant U/f forhold, ulike filter og flere typer sensorer for overvåking.

Referanser

BØKER

G.Maarten Bonnema, Karel Th. Veenvliet, Jan F. Broenink

2016. *System design and engineering, Facilitating Multidisciplinary Development Projects.*

New York, London: CRC Press

NEK 410-1

2008. *Norsk elektroteknisk norm – Elektriske installasjoner om bord i skip og fartøyer, Del 1.*

Lysaker, Norge: Norsk elektroteknisk komite

Scott Fitzgerald, Michael Shiloh

2013. *Arduino Projects Book*

Torino, Italy: Arduino LCC

Olav Vaag Thorsen

2000. *Kraftelektronikk*

Oslo, Norge: Gyldendal Norsk Forlag AS

INTERNETT

ABB

2017a. Technical guide No.1 – Direct torque control. *http://www.abb.com/abblibrary/DownloadCenter*

<http://search-ext.abb.com/library/Download.aspx?DocumentID=3AFE64514482&LanguageCode=en&DocumentPartId=1&Action=Launch> 10.05.15

2017b. Technical guide No. 6 – Guide to harmonics with AC drives, *http://www.abb.com/abblibrary/DownloadCenter/*

https://library.e.abb.com/public/6866afe6c69df93dc1257888004521e1/ABB_Technical_guide_No_8_REVB.pdf 09.05.18

2011. Technical guide No. 8 - Electrical Braking. *http://www.abb.com/abblibrary/DownloadCenter/*
https://library.e.abb.com/public/6866afe6c69df93dc1257888004521e1/ABB_Technical_guide_No_8_REVB.pdf 14.05.18

2017c. Hardware manual – Sine filters. *search-ext.abb.com*
<http://search-ext.abb.com/library/Download.aspx?DocumentID=3AXD50000016814&LanguageCode=en&DocumentPartId=1&Action=Launch>
15.05.18

ARDUINO

2018. Pull up and down resistors. *arduino.cc*
<https://playground.arduino.cc/CommonTopics/PullUpDownResistor> 02.05.18

2015. Analog In, Out Serial. *arduino.cc*
<https://www.arduino.cc/en/Tutorial/AnalogInOutSerial> 01.05.18

DANFOSS

2015. Output filters design guide. *danfoss.com*
http://files.danfoss.com/download/Drives/doc_MG90N402.pdf 20.04.18

2010. Common-mode core kit. *electronicpowersolutions.com*
<http://electronicpowersolutions.com/wp-content/uploads/2014/07/MP-02-4A-02-HF-CM-cores1.pdf> 18.04.18

ELECTRICAL ACADEMIA

2018. Motor phase voltage and current. *electricalacademia.com*
<http://electricalacademia.com/dc-machines/voltage-source-inverters-vsi-operation-vsi-working-principle/attachment/motor-phase-voltage-and-current-when-connected-to-six-step-vsi/> 17.04.18

ELECTRONICS

2011. Electrical engineering. *electronics.stackexchange.com*
<https://electronics.stackexchange.com/posts/11786/revisions> 18.04.18

LÆRINGSPSYKOLOGI

1998. Læringspsykologi (e-bok). *psyk.cc*

<http://www.psyk.cc/lbok/> 15.05.18

NCBI

2009. Learning styles: A review of theory, application, and best practices.

ncbi.nlm.nih.gov

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2690881/> 15.05.18

NTNU

2017. Hva er Arduino? *ntnu.no*

<https://www.ntnu.no/wiki/pages/viewpage.action?pageId=77138637> 14.05.18

NVE

2016. Elektriske motorer. *nve.no*

<https://www.nve.no/energibruk-og-effektivisering/okodesign-og-energimerking-av-produkter/elektriske-motorer/> 13.05.18

2017. Elektriske motorer – spesifikke krav. *nve.no*, 09.02.

<https://www.nve.no/energibruk-og-effektivisering/okodesign-og-energimerking-av-produkter/elektriske-motorer-spesifikke-krav/>

24.05.2018

OrCAD

2008. Introduction to OrCAD Capture and PSpice. *userweb.eng.gla.ac.uk*

<http://userweb.eng.gla.ac.uk/john.davies/orcad/spiceintro160.pdf> 18.05.18

PROPEL AS

2015. Utvikling i skipsstørrelser, motor- og drivstoffteknologi fra 2006 til 2060. *kystverket.no*

<http://www.kystverket.no/contentassets/d4599516369749f5a984231048025d71/utvikling-i-skipsstorrelser-motor--og-drivstoffteknologi-fra-2006-til-2060-propel-2015.pdf>

10.05.18

QUORA

2016. Why do we use capacitor in rectifier circuits? *quora.com*

<https://www.quora.com/Why-do-we-use-capacitor-in-rectifier-circuits> 09.05.18

REFLEKSJONSFILOSOFI

2015. Kunnskapsteori. *docs.refleksjonsfilosofi.org*

<http://docs.refleksjonsfilosofi.org/Kunnskap%20overordnet.htm> 15.05.18

ST

2014. UM1823 – User manual. *st.com*

http://www.st.com/content/ccc/resource/technical/document/user_manual/bf/7a/b0/48/42/d1/48/43/DM00135179.pdf/files/DM00135179.pdf/jcr:content/translations/en.DM00135179.pdf 20.03.18

2018. L6390. *st.com*

<http://www.st.com/content/ccc/resource/technical/document/datasheet/ba/af/1f/e7/1d/33/44/a6/CD00186965.pdf/files/CD00186965.pdf/jcr:content/translations/en.CD00186965.pdf> 20.03.2018

TECHFLO

2014. Variable frequency drive. *tech-flo.net*

<http://www.tech-flo.net/variable-frequency-drive.html> 14.05.18

VVSAKTUELT

2013. IP20 frekvensomformer. *vvsaktuel.no*

<https://www.vvsaktuel.no/markedets-minste-690v-ip20-frekvensomformer-65241/ny-het.html> 14.05.18

WIRED

2014. How do you charge your DC phone with an AC source? *wired.com*

<https://www.wired.com/2014/11/charge-dc-phone-ac-source/> 17.05.18

WOLFRAM

2018. Arduino. *wolfram.com*

<http://reference.wolfram.com/language/ref/device/Arduino.html> 16.05.18

XPLANATOR

2017. Three phase inverter under 180 degree operation. *youtube.com*

<https://www.youtube.com/watch?v=J18PHbv4KQs> 02.03.18

Vedlegg

INNHOLDSFORTEGNELSE

Vedlegg A – Spesifikasjoner på instrumenter brukt i oppgaven.....	73
Vedlegg B – Tekniske data- og brukermanualer for utstyr.....	74
Vedlegg C – BOM (Bill of materials).....	75-76
Vedlegg D – Tidslinjal for oppgaven.....	77
Vedlegg E – AC motor test fra produsent.....	78
Vedlegg F – Koblingskjema for modell.....	79
Vedlegg G – Arduino Mega-kode.....	80-83
Vedlegg H – Arduino Uno-kode.....	84-87
Vedlegg I – Oppgavehefte (Ligger som eget dokument vedlagt innlevering)	

Vedlegg A – Spesifikasjoner på instrumenter brukt i oppgaven



Fluke 43B Power Quality Analyzer

<http://www.testequipmenthq.com/datasheets/FLUKE-43B-Datasheet.pdf> 05.2018



Amprobe 38XR-A

<http://www.amprobe.com/amprobe/usen/digital-multimeters/full-size-multimeters/amp-38xr-a.htm?pid=73039> 05.2018



Tektronix TPS 2014

<https://www.allaboutcircuits.com/test-measurement/oscilloscopes/tps2000-series-tps2014/datasheet/> 05.2018



Tektronix A622

<http://www.testequipmenthq.com/datasheets/TEKTRONIX-A622-Datasheet.pdf>
05.2018



ELMA DT-2236

https://www.mrclab.com/data/products/DT-2236_OPR.pdf 05.2018

Vedlegg B – Tekniske data- og brukermanualer for utstyr

STEVAL-IHM023V3

http://www.st.com/content/ccc/resource/technical/document/user_manual/bf/7a/b0/48/42/d1/48/43/DM00135179.pdf/files/DM00135179.pdf/jcr:content/translations/en.DM00135179.pdf 05.2018

L6390

<http://www.st.com/content/ccc/resource/technical/document/datasheet/ba/af/1f/e7/1d/33/44/a6/CD00186965.pdf/files/CD00186965.pdf/jcr:content/translations/en.CD00186965.pdf> 05.2018

Arduino Mega

<http://www.mantech.co.za/datasheets/products/A000047.pdf> 05.2018

Arduino Uno

<https://www.farnell.com/datasheets/1682209.pdf> 05.2018

IR-sensor (Panasonic PM-F45-P)

<https://datasheet.octopart.com/PMF45P-Panasonic-datasheet-66325599.pdf> 05.2018

Vedlegg C – BOM (Bill of materials)

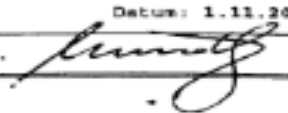
Indeks	Type	Antall	Produsent	Leverandør	Artikkel nr.	Pris NOK
1	Lenovo V110-15ISK	1	Lenovo	ATEA	80TL00A4MX	4036,0
2	Asynkronmotor	1	Lönne	Brammer	106473	1266,7
3	Motorstyringskort IHM023V3	1	STMicroelec- tronics	Farnell	2475463	1800,0
4	Mega 2560 Utviklingskort	1	Arduino	Kjell & company	87861	499,9
5	Genuino Uno Utviklingskort	1	Arduino	Kjell & company	87860	229,9
6	LCD Display 4x20	1	Luxorparts	Kjell & company	90216	129,9
7	Potensiometermodul for Arduino	1	Luxorparts	Kjell & company	90470	59,90
8	USB Kabel USB-B	2		Kjell & company	68151	79,9
9	Breadboard Koblingsbrett	1	Luxorparts	Kjell & company	36283	69,9
10	Trykkbryter	1		Biltema	43029	24,9
11	Strømbryter Rektangulær	1		Biltema	43030	24,9
12	Jumperwire Male-Male	1	Bud Industries Inc	Farnell	2762506	148,0
13	Jumperwire Female-Female	1	Bud Industries Inc	Farnell	2762507	130,0

14	Fotoelektrisk sensor PM-F45-P-C3	1	Panasonic	Elfa Dis- trelec	300-75- 634	231,2
15	Sikkerhetsuttak Sort SB 4mm schwarz	7	Schnepp	Elpro.org		22,0
16	Sikkerhetsuttak Gul SB 4mm gelb	1	Schnepp	Elpro.org		22,0
17	Isolerte bananplugger sort	10	Fikk de av Arild Sæbø			
18	Isolerte bananplugger gul	1	Fikk de av Arild Sæbø			
19	Strømforsyning med bryter og glassikring	1	Fikk de av Arild Sæbø			
20	PN 1,5 mm sort	1		Clas Ohlson	49-439-5	49,9
21	PN 1,5 mm gul/grønn	1		Clas Ohlson	49-438-5	49,9
22	Niter 1,5 mm	1	Vogt	Elfa Dis- trelec	148-23- 508	16,0
23	Kabelsko 1,5 mm	1	RND Connect	Elfa Dis- trelec	300-66- 617	140,0
24	Krympeslanger	1		Clas Ohlson	36-1368	149,0
25	Plexiglass	1		Biltema	262112	169,0
26	Hurtigepoksy	1		Biltema	364016	39,9
27	Møbelknotter	1		Biltema	86315	19,9
28	Kanthsler	1		Clas Ohlson	40-7067	59,9
29	Trappnese/Hjørnelist	1		Clas Ohlson	40-7886	99,9
30	Låsehengsel	1	Hard Head	Jula	34-34-11	49,9
31	Hengelås	1	Asaklitt TSA	Clas Ohlson	40-7470-1	49,9
32	Plastkoffert	1		Clas Ohlson	40-8470	749,0

Vedlegg D – Tidslinjal for oppgaven

Måned	Jan	Jan/Feb	Feb/Mar	Mar/Apr	Apr/Mai
Uke nr.	1 2 3 4	5 6	7 8	9 10 11 12	13 14 15 16 17
Uke nr.	18 19 20	25			
Aktivitet:					
Konsept	Blue				
Bli kjent med prog. og utstyr	Red	Red	Red		
Materiellanskaffelse		Purple			
Sammensetting			Yellow	Yellow	
Utvikle software			Cyan		
Prøving			Green	Green	
Utvikling av oppgaver			Light Green	Light Green	
Målgruppe tester modell				Black	
Utbedring/Utvildelse					Orange
Ferdigstilling Modell					Red
Oppgaveskriving		Dark Green	Dark Green	Dark Green	Dark Green
Ferdigstilling Bachelor					Cyan
					Imlevering

Vedlegg E – AC-motor-test fra produsent

SIEMENS Elektromotory s.r.o. MOHELNICE		PROTOKOLL UEBER TYPENPRUEFUNG			PROTOKOLLNUMMER T0063-770/2K		
Drehstrommotor mit Kaefiglaeufer		Typ: TAA63M02K		Herstellnummer: 0833-0010			
C a r a n t	Leistung: 0.180 kW		Belastungsart: S1		Frequenz: 50.00 Hz		
	Drehzahl: 2820 1/min		Leistungsfaktor: 0.82		Wirkungsgrad: 63.0 %		
	Spannung: 400 V		Schaltung: Y		Strom: 0.51 A		
	Form: IM 3641		Schutz: IP55		Umgebung: -		
W e r t e	Mk/Mn: 2.00		Ik/In: 3.70		Waermeschutz: -		
	Mmax/Mn: 2.20		Schall Kl.: 1		Techn. Beding.: -		
	Isol.Kl.: F		Umgb.T.: 40 °C		Ben.: Norma: EN 60034, DIN VDE 0530		
TV Nummer: 02.40300		Nulleiterszahl 1x163 Draht 1x0.265		Zweig 1		Schritt 1-11	
1. Mechan. Grundpruefungen:							
2. Kaltwiderstand d. Isol.:		100.0 MO		Warmwiderstand d. Isol.:		50.0 MO	
3. Kalt-Kleinwiderstand:		123.50 / 125.50 / 127.00 Ω		bei Wicklungstemp. 20.2 °C		-	
		- warm: 146.39 / 146.39 / 146.39 Ω		bei Umgebungstemp. 20.6 °C		-	
4. Windungsisolationspruef. 1.3Un/180s		U 520 V		5. Pruefung der el. Festigkeit		U 1800 V / 60 s	
6. Erwaerung: a/ bei Un und Mn :		Wicklung 46.9 K		b/ bei 420 V & Mn		Wicklung 48.7 K	
		Oberfl. 17.4 K				Oberfl. 18.0 K	
7. Pruef. der mechan. Festigkeit 1.2 nm / 120 s						3385 min.-1	
8. Schwingungsmessung: vefmax		0.8 mm/s		9. Schallmes.: Schallleistung		Lw 54.2 dB(A)	
10. Messung der Kuehlluftmenge: Q		m3/min		11. Schutzartpruefung			
12. Leerlaufmessung:		Rk0 [Ω]		U10 [V]		I10 [A]	
Anlage Nr. 1		137.00		400		0.395	
				P10 [W]		cosφ10 [-]	
				68.6		0.251	
13. Pruefung d. Belastung bei M = konst.: Charakteristiken: Belastung - Anlage Nr. 3, Regelung - Anlage Nr. 4							
U1 [%]		U1 [V]		I1 [A]		P1 [W]	
95.0		380		0.508		285.1	
100.0		400		0.512		290.9	
105.0		420		0.516		291.1	
n[min-1]		M [Nm]		cosφ [-]		P [W]	
2759		0.62		0.853		176.5	
2785		0.62		0.819		180.0	
2812		0.62		0.776		178.1	
eta [%]		f [Hz]		s [%]			
61.9		50.00		8.0			
61.9		50.00		7.2			
61.2		50.00		6.3			
14. Stromueberlastungspruef. 1.5In/120s		I1 0.765 A		15. Momentueberlastungspruef. 1.6Mn/15s		M 1.0 Nm	
16. Messung der Momentcharakteristik							
17. Minimal Anlaufmoment		Mmin 1.27 Nm (2.08 Nm)		18. Klippmoment			
				Mz 1.4 Nm (2.30 Nm)			
19. Kurzschlussmes.		U1k [V]		I1k [A]		P1k [W]	
Anlage Nr. 2		400		1.94		1244	
				cosφk [-]		Nk [Nm]	
				0.926		1.4	
				Mk/Mn		Ik/In	
				2.21		3.79	
				Raz		7.47	
20. Traegheitsmoment				J			
				kgm2			
21. Verluste bei Un und Pn		δPn [W]		δPPc [W]		δPj1 [W]	
		15.8		20.7		57.7	
				δPj2 [W]		δPd [W]	
				15.7		1.5	
				IδP [W]		eta [%]	
				111.3		61.9	
22. Weitere Messungen:				Erwaerung nach 30s Abkuehlung			
				39.1 K			
23. Zusammenfas.: Entspricht IEC 34-1, EN 60034, DIN VDE 0530							
Messungsplatz-Nr.:		11		Blaetteranzahl mit Anlagen: 6			
Geprueft von: KULISEK		Datum: 1.11.2000		Bearbeitet von: KULISEK		Datum: 1.11.2000	
Leiter der Pruefstelle: ..				Datum: ..			

Vedlegg F – Koblingskjema for modell

Koblingskjema:				
Komponent	Fra	Farge jumper	Til	Komponent
Fase L1	Forsyning	Blå	J1 - N (via hovedbryter)	Motorkontroller
Fase L2	Forsyning	Svart	J1 - L (via sikring og hovedbryter)	Motorkontroller
Jordleder	Forsyning	Gul/Grønn	J1 - PE	Motorkontroller
Potmeter	SIG	Rød splittes	A0	Arduino Uno
		til 2 sorte	A0	Arduino Mega
	VCC	Oransje (100ohm)	5V+	Breadboard
	GND	Gul	0V-	Breadboard
Start/Stopp Bryter	Inn	Lilla	5V+	Breadboard
	Ut	Blå (10Kohm)	0V-	Breadboard
	Ut	Grønn (100ohm)	D31	Arduino Mega
LED	Anode	Rød	D35	Arduino Mega
	Katode	Brun	GND	Arduino Mega
Dreieretning Bryter	Inn	Oransje (100ohm)	5V+	Breadboard
	Ut	Gul (10Kohm)	0V-	Breadboard
		Oransje (100ohm)	D33	Arduino Mega
IR sensor	Inn	Gul (100ohm)	5V+	Breadboard
	Ut	Brun	GND	Arduino Uno
	SIG	Lilla	D2	Arduino Uno
	Ut	Blå	0V-	Breadboard
Sort bananplugg	TP Ref	Sort	TP21	Motorkontroller
Sort bananplugg	TP T1_T4	Sort	TP5	Motorkontroller
Sort bananplugg	TP T3_T6	Sort	TP7	Motorkontroller
Sort bananplugg	TP T5_T2	Sort	TP9	Motorkontroller
Sort bananplugg	A	Sort	J2 - 3	Motorkontroller
Sort bananplugg	B	Sort	J2 - 2	Motorkontroller
Sort bananplugg	C	Sort	J2 - 1	Motorkontroller
Gul/Grønn bananplugg	PE Motor	Gul/Grønn	J1 - PE	Motorkontroller
LCD	VSS	Brun	0V-	Breadboard
LCD	VDD	Rød	5V+	Breadboard
LCD	V0	Oransje (100ohm)	0V-	Breadboard
LCD	RS	Gul	D7	Arduino Uno
LCD	RW	Grønn	0V-	Breadboard
LCD	E	Blå	D8	Arduino Uno
LCD	D4	Lilla	D9	Arduino Uno
LCD	D5	Grå	D10	Arduino Uno
LCD	D6	Hvit	D11	Arduino Uno
LCD	D7	Sort	D12	Arduino Uno
LCD	A	Brun	5V+	Breadboard
LCD	K	Rød	0V-	Breadboard
Arduino Mega	D6	Oransje splittes til	J5 - 3	Motorkontroller
		Rød og Oransje	J5 - 5	Motorkontroller
Arduino Mega	D5	Gul splittes til	J5 - 7	Motorkontroller
		Gul og Grønn	J5 - 9	Motorkontroller
Arduino Mega	D4	Grønn splittes til	J5 - 11	Motorkontroller
		Lilla og Blå	J5 - 13	Motorkontroller
Arduino Mega	GND	Grå	J5 - 2	Motorkontroller
Arduino Uno	A4	Grå til Brun	J5 - 14	Motorkontroller
Arduino Uno	5V	Hvit	5V+	Breadboard
Arduino Uno	GND	Sort	0V-	Breadboard
Arduino Uno	GND	Grå	0V-	Breadboard

Motorstyring_Mega

//180 GRADERS SIX STEP INVERTER

//Her deklarerer alle variabler

```

const int t1_t4 = 4; //deklarerer variabelen t1_t4 som constant integer med adresse 4. IGBT T1 og T4 blir styrt av denne variabelen
const int t3_t6 = 5; //deklarerer variabelen t3_t6 som constant integer med adresse 5. IGBT T3 og T6 blir styrt av denne variabelen
const int t5_t2 = 6; //deklarerer variabelen t5_t2 som constant integer med adresse 6. IGBT T5 og T2 blir styrt av denne variabelen
const int start = 31; //deklarerer variabelen start som constant integer med adresse 31
const int dreieretning = 33; //deklarerer variabelen dreieretning som constant integer med adresse 33
const int led = 35; //deklarerer variabelen led som constant integer med adresse 35
const int potmeter = A0; //deklarerer variabelen potmeter som en constant integer med adresse analog 0

```

```

float hz; //deklarerer variabelen hz som en float
float steg; //deklarerer variabelen steg som en float
int sensorverdi; //deklarerer variabelen sensorverdi som en integer
int les_start; //deklarerer variabelen les_start som en integer
int les_dreieretning; //deklarerer variabelen les_dreieretning som en integer

```

//setup kjører en gang i starten av programmet

```

void setup() {
  Serial.begin(9600); //starter seriell overvåking med baudrate 9600 bits per sekund
  pinMode(t1_t4,OUTPUT); //setter variabel t1_t4 som output
  pinMode(t3_t6,OUTPUT); //setter variabel t3_t6 som output
  pinMode(t5_t2,OUTPUT); //setter variabel t5_t2 som output
  pinMode(start,INPUT); //setter variabel start som input
  pinMode(dreieretning,INPUT); //setter variabel dreieretning som input
  pinMode(led,OUTPUT); //setter variabel led som output
}

```

//loop vil starte etter setup er ferdig, så vil programkoden kjøre om og om igjen inne i loop

void loop() {

```

  Serial.println(); //brukes til å lese av variabel verdier, skriv inn ønsket variabelnavn mellom parentesene og åpne seriell overvåker

```

```

  les_start = digitalRead(start); //digitalRead leser av den digitale inngangen start. og leger verdien inn i variabelen les_start

```

Vedlegg G – Arduino Mega-kode


```

les_start = digitalWrite(start);
leg_dreieretning = digitalWrite(dreieretning);
//digitalread leser av den digitale inngangen start, og legger verdien inn i variabelen les_start
//digitalread leser av den digitale inngangen dreieretning, og legger verdien inn i variabelen leg_dreieretning

sensorverdi = analogRead(potmeter);
hz = map(sensorverdi, 0, 1010, 20, 50);
//analogread leser av den analoge inngangen potmeter, og legger verdien inn i variabelen sensorverdi
//map fungerer som en lineær trafo som tar en analog verdi fra 0 til 1010 og skalerer den til å være lineær fra 20 til 50,
//en verdi mellom 20 til 50, avhengig av potmeterets stilling, blir så skrevet inn i variabelen hz
steg = (((1.0/hz)*1000.0)/6.0);
//steg er en variabel vi bruker til å regne ut hvor lenge vi skal forsinke hvert steg hver 60 grader av 360 grader
//1/hz for å gjøre om til sekunder, så ganger vi med 1000 for å få millisekunder, så deler vi på 6, for å få tiden hver 60 grader
digitalWrite(led, LOW);
//setter digital utgang led lav, deaktiverer en grønn led over startknappen som indikerer at start ikke er aktivert

if (les_start == 1){
    digitalWrite(led, HIGH);
    //hvis variabelen les_start er høy, vil koden inne i if sløyfen begynne å kjøre
    //setter digital utgang led høy, aktiverer grønn led over startbryteren for å vise at start er aktivert

    if (leg_dreieretning == 1){
        //0-60GRADER
        //hvis variabelen leg_dreieretning er høy, vil koden inne i if sløyfen kjøre
        digitalWrite(t1_t4, HIGH);
        digitalWrite(t3_t6, LOW);
        //setter digital utgang t1_t4 høy
        //setter digital utgang t3_t6 lav
        digitalWrite(t5_t2, HIGH);
        //setter digital utgang t5_t2 høy
        delayMicroseconds(steg*1000);
        //delayMicroseconds pauser koden i like mange microsekunder som er inne i parentes. Vi ganger med 1000 for å få det i millisekunder
        //avhengig av hva potmeteret er stillt inn på, vil programet pause slik at vi får riktig frekvens ut på motoren
        digitalWrite(t1_t4, HIGH);
        digitalWrite(t3_t6, LOW);
        digitalWrite(t5_t2, LOW);
        delayMicroseconds(steg*1000);
        //120-180GRADER
        digitalWrite(t1_t4, HIGH);
        digitalWrite(t3_t6, HIGH);
        digitalWrite(t5_t2, LOW);
        delayMicroseconds(steg*1000);
        //180-240GRADER
        digitalWrite(t1_t4, LOW);
        digitalWrite(t3_t6, HIGH);
        digitalWrite(t5_t2, LOW);
        delayMicroseconds(steg*1000);
        //240-300GRADER
        digitalWrite(t1_t4, LOW);
        digitalWrite(t3_t6, LOW);
    }
}

```

```

digitalWrite(t1_t4, LOW);
digitalWrite(t3_t6, HIGH);
digitalWrite(t5_t2, HIGH);
delayMicroseconds (steg*1000);
//300-360GRADER
digitalWrite(t1_t4, LOW);
digitalWrite(t3_t6, LOW);
digitalWrite(t5_t2, HIGH);
delayMicroseconds (steg*1000);
}
if(les_dreieretning == 0){
//300-360GRADER
digitalWrite(t1_t4, LOW);
digitalWrite(t3_t6, LOW);
digitalWrite(t5_t2, HIGH);
delayMicroseconds (steg*1000);
//240-300GRADER
digitalWrite(t1_t4, LOW);
digitalWrite(t3_t6, HIGH);
digitalWrite(t5_t2, HIGH);
delayMicroseconds (steg*1000);
//180-240GRADER
digitalWrite(t1_t4, LOW);
digitalWrite(t3_t6, HIGH);
digitalWrite(t5_t2, LOW);
delayMicroseconds (steg*1000);
//120-180GRADER
digitalWrite(t1_t4, HIGH);
digitalWrite(t3_t6, HIGH);
digitalWrite(t5_t2, LOW);
delayMicroseconds (steg*1000);
//60-120GRADER
digitalWrite(t1_t4, HIGH);
digitalWrite(t3_t6, LOW);
digitalWrite(t5_t2, LOW);
delayMicroseconds (steg*1000);
}
}
//hvis variabelen les_dreieretning er lav, vil koden inne i if sløyefen kjøre
//For å kjøre motoren i motsatt dreieretning leses koden av motsatt vei enn tidligere

```

```
//0-60GRADER
digitalWrite(t1_t4, HIGH);
digitalWrite(t3_t6, LOW);
digitalWrite(t5_t2, HIGH);
delayMicroseconds(steg*1000);
}
}
```

Utskrift ferdig.



```

//overvåking av dc bus, frekvens og turtall som skrives ut på lcd display

#include <LiquidCrystal.h> //inkluderer bibliotek for lcd display

LiquidCrystal lcd(7, 8, 9, 10, 11, 12); //definerer lcd utganger

const int potmeter = A0; //deklarerer variabelen potmeter som en constant integer med adresse analog 0
const int dc_link = A4; //deklarerer variabelen dv_link som en constant integer med adresse analog 4

bool b_søk = false; //deklarerer variabelen b_søk som en boolsk variabel. brukes som flag til utregning av rpm. settes av timer1.
unsigned long lodoppdater; //deklarerer variabelen lodoppdater som en unsigned long integer. brukes til å lagre tiden for hvor ofte lodskjermen skal oppdatere deg
volatile byte teller; //deklarerer variabelen teller som volatile byte. volatile brukes fordi teller er i en interrupt utenfor loopen
unsigned int rpm; //deklarerer variabelen rpm som en unsigned integer. positive verdier fra 0 til 65535
int les_dc_link; //deklarerer variabelen les_dc_link som integer, og initierer variabelen med verdien null
int les_hz; //deklarerer variabelen hz som integer, og initierer variabelen med verdien null
int hz; //deklarerer variabelen hz som integer

void isr() //void isr er en interrupt som blir aktivert hver gang ir sensoren får en falling edge, dette skjer uavhengig av hvor programmet er i koden
{
    teller++; //variabelen teller plusses på 1 for hver gang void isr kjører
}
//setup kjører en gang i starten av programmet
void setup()
{
    Serial.begin(9600); //starter seriell overvåking med baudrate 9600 bits per sekund
    lcd.begin(20,4); //starter lcd display med oppløsning 20x4
    attachInterrupt(digitalPinToInterrupt(2), isr, FALLING); //digital inngang 2 (ir sensor) skal bli lest på fallende flanke, for hver gang det skjer vil interrupt isr bli aktivert
    pinMode (dc_link, INPUT); //setter dc_link som en input
    cli(); //disabler avbrudd
}

```

```

//setter timer1 interrupt til 1hz

TCCR1A = 0;
TCCR1B = 0;
TCNT1 = 0;
// setter compare match register til 1hz trinn

OCRA1A = 15624;
TCR1B |= (1 << WGM12);
TCR1B |= (1 << CS12) | (1 << CS10);
TIMSK1 |= (1 << OCIE1A);

sei();
//åpner for avbrudd

//skriver ut en tekst i starten etter vi har lastet inn programmet

    lcd.clear();
    lcd.setCursor(3,0);
    lcd.print("Bachelor 2018");
    lcd.setCursor(8,1);
    lcd.print("Av ");
    lcd.setCursor(3,2);
    lcd.print("Birger Andresen");
    lcd.setCursor(1,3);
    lcd.print("Alf Inge Fagerheim");
    delay(10000);
}

//timer1 callback rutine (1hz)

ISR(TIMER1_COMPB_vect)
{
    b_sek=true;
}

//variabel b_sek blir true hvert sekund

//loop vil starte etter setup er ferdig, så vil programkoden kjøre om og om igjen inne i loop

```

```

void loop()
{
  les_hz = analogRead(potmeter);

  //analogread leser av den analoge inngangen potmeter, og legger verdien inn i variabelen les_hz

  hz = map(analogRead(potmeter), 0, 1006, 20, 50);
  //map fungerer som en lineær trafo den tar en analog verdi fra 0 til 1010 og skalerer den til å være lineær mellom 20 og 50,
  //en verdi mellom 20 til 50, avhengig av potmeterets stilling, blir så skrevet inn i variabelen hz

  //her regner vi ut rpm hvert sekund

  if( b_seek)
  {
    b_seek=false;
    detachInterrupt(0);
    rpm = teller*60/6;

    //hvis b_seek er true, så vill vi gå inn i if sløyfen. noe som skjer hvert sekund

    teller = 0;
    attachInterrupt(0, isr, FALLING);
    les_dc_link = map(analogRead(dc_link), 0, 541, 0, 324);
    //les_dc_link maper en verdi fra 0 til 541 fra motorkontrollkortet, vi gjør så den verdien lineær mellom 0 og 324
  }

  if( ( millis()-loddopdater ) >= 100 )
  //hvis millis - loddopdater er større eller lik 100 millisekunder vil vi gå inn i if sløyfen igjen
  //det vil si at lcd skjermen oppdaterer seg hvert 100 millisekund
  {
    //samme som for oppstartsteksten, men her skriver vi også ut variabler

    lcd.clear();
    lcd.setCursor(5,0);
    lcd.print("MotorDrive");
    lcd.setCursor(0,1);
    lcd.print("DC-Buss = ");
    lcd.print(les_dc_link);
    lcd.print(" Volt");
    lcd.setCursor(0,2);
    lcd.print("Frekvens = ");
    lcd.print(hz);
    lcd.print(" Hz");
    lcd.setCursor(0,3);
    lcd.print("Turtall = ");
  }
}

```

```
    lod.print(rpm);  
    lod.print(" RPM");  
    lodoppdater = millis();  
  }  
}
```