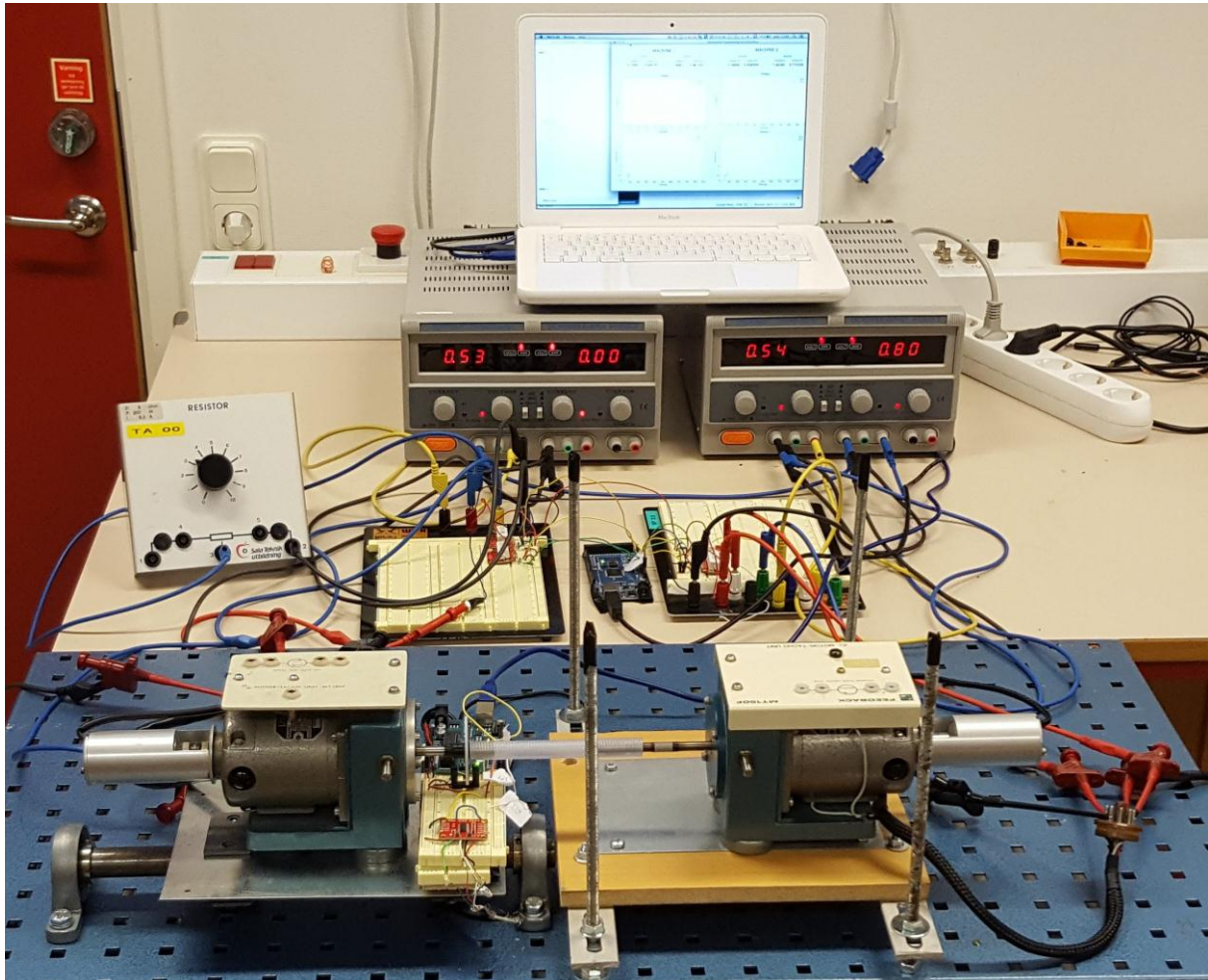




# CHALMERS



## Automatisk registrering av motordata

Examensarbete inom högskoleingenjörsprogrammet Elektroteknik

PATRIK ALERMAN  
INGRID-MARIE LINDQVIST

Institutionen för Signaler och system  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige 2016

## **Automatisk registrering av motordata**

Examensarbete inom högskoleingenjörsprogrammet Elektroteknik

Patrik Alerman, Ingrid-Marie Lindqvist

© PATRIK ALERMAN, INGRID-MARIE LINDQVIST, 2016

Institutionen för Signaler och System

Chalmers tekniska högskola

SE-412 96 Göteborg

Sverige

Telefon: +46 (0)31-772 1000

Institutionen för Signaler och System

Göteborg, 2016

## **SAMMANFATTNING**

Ny labbutrustning erfordras av Chalmers tekniska högskola för undersökning av elektriska maskiner. Nuvarande utrustning används för närvarande av flera studenter åt gången och har även stor energiförbrukning. Ny, mindre utrustning är därför att föredra, därav har en mätrigg utformats. Mätriggen skall kunna automatiskt registrera och digitalt presentera mätdata hos elmaskiner. Undersökning om huruvida en konstruktion kan digitalt övervaka alla parametrar hos elmaskiner samtidigt, om den nya mätriggen lämpar sig för utbildningsändamål samt om olika varianter av motorstyrning har genomförts. Mätriggen som konstruerades kan mäta strömmar och spänningar i både rotor och stator, även varvtal och moment kan mätas. Övervakningen av mätvärdena har god noggrannhet och sker digitalt i Arduino samt MATLAB. Eftersom mätriggen kan visuellt presentera elektriska maskiners karakteristik lämpar sig riggen för utbildningssyfte. Mätriggen kan endast övervaka likströmsmaskiner men ett utbyte av ström- och spänningssensorer medför möjligheter till att växelströmsmaskiner kan övervakas.

## **ABSTRACT**

Chalmers University of Technology requires new equipment for monitoring electrical machines. Existing equipment is currently used by several students at a time and also has high power consumption. New, smaller equipment is therefore preferable; hence a measuring rig has been constructed. The measuring rig needs to be able to automatically register and digitally display measurement data in electrical machines. Evaluations whether one construction can digitally monitor all parameters of electrical machines at the same time and whether the new measuring rig is suited for educational purposes have been conducted together with an investigation of different types of motor controls. The constructed measuring rig is able to measure currents and voltages in both the rotor and the stator. The rotational speed and torque of the machine can also be measured. Monitored results have good accuracy and are done digitally in Arduino and MATLAB. The measuring rig can visually present characteristics of electrical machines and is therefore suitable for educational purposes. The measuring rig can only monitor direct current machines, but a substitution of the current and voltage sensors would provide means for the rig to monitor alternating current machines as well.

# FÖRORD

Denna rapport är en del av examensarbetet som utförts på Chalmers tekniska högskola, institutionen för Signaler och system. Arbetet motsvarar 15 högskolepoäng och är det slutgiltiga momentet inom elektroingenjörsprogrammet, 180 högskolepoäng.

Vi vill tacka Sakib Sisteck för hans handledning, stöd och engagemang under arbetets gång. Ett stort tack går även till Lennart Widén för hans hjälp med vägledning om motorer samt Robert Svensson för hans hjälp med konstruktionen av mätriggen. Vi vill även tacka våra närmaste för deras stöd under arbetets gång.

Göteborg, juni 2016

Patrik Alerman

Ingrid-Marie Lindqvist

# INNEHÅLLSFÖRTECKNING

BEGREPP .....	1
1. INLEDNING .....	2
1.1 Bakgrund .....	2
1.2 Mål .....	3
1.3 Syfte .....	3
1.4 Avgränsningar .....	3
1.5 Preciseringar av frågeställningar .....	3
2 TEKNISK BAKGRUND .....	4
2.1 Elektrisk laddning .....	4
2.2 Maxwells ekvationer .....	5
2.3 Elektriska maskinen .....	6
2.3.1 Elektromekanisk omvandling .....	6
2.4 Likströmsmaskinen .....	7
2.4.1 Uppbyggnad .....	7
2.4.2 Funktion vid motordrift .....	8
2.5 Magnetisering av likströmsmotorer .....	10
2.5.1 Separatmagnetisering .....	10
2.5.2 Shuntmagnetisering .....	12
2.5.3 Seriemagnetisering .....	12
2.5.4 Kompoundmagnetiserad likströmsmotor .....	14
2.5.5 Funktion vid generatordrift .....	15
2.6 Växelströmsmaskinen .....	15
2.7 Asynkronmaskinen .....	16
2.7.1 Uppbyggnad .....	16
2.7.2 Funktion vid motordrift .....	16
2.7.3 Momentkurva för asynkronmotor .....	17
2.8 Synkronmaskinen .....	18
2.8.1 Uppbyggnad .....	18
2.8.2 Funktion vid motordrift .....	19
2.9 Wheatstonebrygga .....	19
2.10 A/D-omvandlare .....	20
2.11 Elsäkerhet .....	20

2.12	Arduino .....	21
2.12.1	Olika typer och deras skillnader .....	21
2.12.2	Arduino Shields .....	21
2.12.3	A/D-Omvandlaren i Arduino .....	21
2.13	LabVIEW .....	22
2.13.1	Vad är LabVIEW? .....	23
2.14	MATLAB .....	24
2.14.1	Mer om MATLAB .....	24
3.	METOD .....	26
4	KOMPONENTER .....	27
4.1	Transistor - IRF520 .....	27
4.2	H-brygga - L293DNE .....	27
4.3	Tryckkänsligt motstånd 0.5" .....	27
4.4	Strömsensor - INA169 .....	27
4.5	Lågströmsensor - ACS712 .....	28
4.6	Reflexdetektorn - QRE1113 .....	28
4.7	Ljusbrytare - GP1A57HRJ00F .....	29
4.8	LCD display - HD44780 .....	29
4.9	Lastcell - TAS606 .....	30
4.10	Lastcellsförstärkare - HX711 .....	30
5	TESTER .....	31
5.1	Spännings- och strömmätning .....	31
5.1.1	Ström- och spänningsmätning med ACS712 .....	31
5.1.2	Spänningsmätning i Arduino med spänningsdelning .....	31
5.1.3	Ström- och spänningsmätning med INA169 .....	32
5.2	Mätning av moment .....	35
5.2.1	Tryckmätning med tryckkänsligt motstånd .....	35
5.2.2	Tryckmätning med lastcell TAS606 .....	35
5.3	Varvtalsmätning .....	36
5.3.1	Varvtalsmätning med reflexdetektor breakout QRE1113 .....	36
5.3.2	Varvtalsmätning med ljusbrytare - GP1A57HRJ00F .....	36
5.4	Motorstyrning .....	38
5.4.1	Motorstyrning med Arduino och transistor .....	38

5.4.2 Test av motorstyrning med Arduino och H-brygga.....	38
5.4.3 Motorstyrning med motor shield .....	39
6. SLUTGILTIG KONSTRUKTION .....	40
6.1 Uppbyggnad.....	40
6.2 Verifiering .....	40
7 RESULTAT .....	43
8 DISKUSSION .....	46
9 SLUTSATS .....	49
REFERENSER.....	50
BILAGOR .....	1
A – Arduinokod .....	1
B – Tester på lösningsalternativ .....	1
C – MATLABkod.....	1
D – MATLAB GUI .....	1
E – Lathund: LabVIEW + Arduino .....	1
F – Kopplingsschema för slutgiltig konstruktion .....	1
G – Gränssnitt hos slutgiltig konstruktion .....	1



## **BEGREPP**

*Utpräglade poler* – rotorlindningen är lindad kring samtliga poler i rotorn.

*Cylindrisk rotor* – rotorlindningen ligger i rotorns spår.

*Reluktans* – magnetiskt motstånd.

*Märkdata* – beskriver maximalt tillåtna värden på storheter som en maskin kan utsättas för.

*Tachometer* – mätindikator som används för att mätning av varvtal.

*AC/DC Clampmeter* – ett flexibelt mätinstrument som kan mäta stora spänningar och strömmar för likspänning- och växelspanning.

*Stator* – stillastående del hos en roterande, elektrisk maskin

*Rotor* – rörlig del hos en roterande, elektrisk maskin

*Halleffekt* – ett fenomen där ett magnetfält läggs tvärs över en tunn platta, oftast i metall eller halvledarmaterial, samtidigt som en ström passerar plattan. Strömmen skär fältet ortogonalt vilket ger upphov till att en spänning, en hallspänning, skapas över plattan.

# 1. INLEDNING

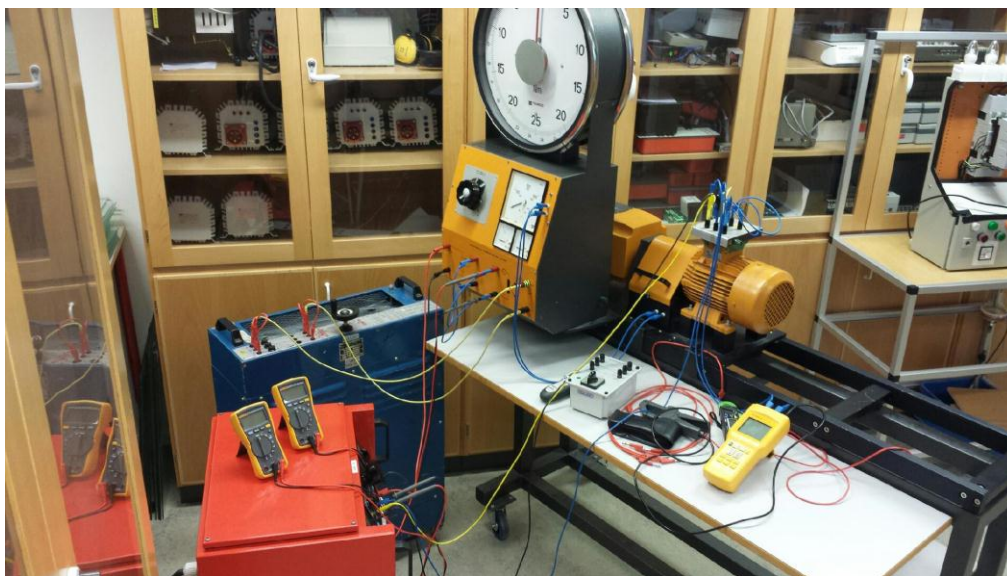
I kapitlet presenteras projektets bakgrund, syfte, mål, avgränsningar och frågeställningar.

## 1.1 Bakgrund

Chalmers tekniska högskola vill ha labbutrustning för studier och undersökning av elmaskiner. För tillfället finns maskiner som brukas vid höga effekter samt är stora och otympliga, se figur 1.1. Detta gör att det är svårt att förflytta utrustningen mellan olika lokaler. Det är även otympligt att försöka ställa upp ett flertal uppsättningar av den gamla utrustningen i laborationssalarna på grund av dess storlek. Därför finns det för nuvarande endast en uppsättning av utrustningen och studenterna tvingas genomföra mätningar på maskinerna i stora grupper. Detta kan försvåra inläringen för varje individuell student. Om en mindre mättrigg konstrueras kan varje laborationspar arbeta i sin egen takt och även kunna köra maskinerna på egen hand.

Den nuvarande mätutrustningen arbetar vid spänningsnivåer som ligger på ca 220 V. En fysiskt mindre mätkonstruktion kan istället använda sig av lägre spänningsnivåer och på så sätt minska risken för personskador. Dessutom innebär lägre spänningsnivåer lägre effektkonsumtion, vilket leder till en mindre miljöpåverkan. En fysiskt mindre konstruktion innebär även att problemet med förflyttning av konstruktionen underlättas och att fler uppsättningar av mättriggen kan ställas upp, vilket skulle leda till att studenterna kan arbeta i par.

Den gamla mättriggen avläser all mätdata analogt. Om datan kan digitaliseras går det att implementera mätningen i olika grafritande program och då kunna avläsa mätvärden i kurvor som ritas ut i realtid.



*Figur 1.1. Chalmers utrustning för att registrera större elmaskiners viktigaste data.*

## 1.2 Mål

Projektets mål är att skapa en demomättrigg som på ett tydligt sätt presenterar mätdata hos elektriska maskiner. Demomättriggen skall vara utformad för att säkerställa funktionalitet mellan maskiner och sensorer. Konstruktionen ska vara utformad så att studenterna kan erhålla en uppfattning om elektriska maskiner och deras karakteristik. Demomättriggen skall utnyttja lägre spänningar och svagare strömmar än nuvarande utrustning. På så sätt skall slutprodukten ha mindre risk för att orsaka personliga skador och ha mindre miljöpåverkan.

## 1.3 Syfte

Att utforma en konstruktion som automatiskt registrerar och digitalt presenterar mätdata hos en motor och generator.

## 1.4 Avgränsningar

Maskiner av fysiskt större storleksordning samt permanentmagnetiserade likströmsmaskiner behandlades ej. Endast elektriska maskiner berördes.

Olika varianter av övervakningsprogram har endast undersökts enskilt och har därför ej jämförts. En uppskattning av vilken mjukvara som krävdes genomfördes. På så sätt undersöktes inte alla lösningsalternativ.

Undersökning av styrning på annat vis än via spänningsreglering i spänningsaggregat utfördes, dock implementerades inte styrningen.

Mättriggen som konstrueras skall endast vara en funktionsmodell.

## 1.5 Preciseringar av frågeställningar

- Är det möjligt att samtidigt övervaka flera olika parametrar hos elektriska maskiner med en och samma konstruktion?
- Fungerar det att övervaka maskinernas storheter digitalt och i realtid?
- Kommer konstruktionen vara lämplig att användas vid laborationer för studenter?
- Kan en motor styras på annat sätt än vanlig spänningsreglering via spänningsaggregat?

## 2 TEKNISK BAKGRUND

Nedan presenteras fakta kring olika avsnitt inom fysik, elektriska maskiner och mjukvaruverktyg. Faktan ger stöd till kommande konstruktion och analys av mätdata.

### 2.1 Elektrisk laddning

När ett oväder bryter ut mörknar himlen och dess moln laddas upp. Ur detta sker en kort elektrisk urladdning som kan betraktas med både örat och ögat. Detta fenomen är kallat blixten och är en av de synliga effekterna av elektrisk laddning i naturen. Under människans tidsålder upptäcktes synliga blixtar i andra fall, exempelvis när bärnsten gnids mot en mjuk duk eller när håret är nykammat. Det var inte förrän under 1700-talet som det kunde bevisas att det fanns två typer av elektrisk laddning – en positiv och negativ laddning. Utvecklingen och användningen av denna formulering har de nästkommande hundra åren fram till idag blivit en fundamental företeelse av vad elektrisk laddning egentligen är för något [1].

För att erhålla förståelse om elektrisk laddning måste atomens uppbyggnad studeras. En atom består av en central atomkärna innehållande partiklar kallade protoner och neutroner. Där protonerna och neutronerna är positivt respektive neutralt laddade. Kring kärnan finns det lättare partiklar, elektroner, vilka är negativt laddade. Dessa är i antal lika många som protonerna vilket gör atomen neutralt laddad, vilket gör den i princip laddningslös. Elektronerna roterar kring kärnan i banor som bildar elektronskal, där varje skal har en viss mängd energi. Med tillräcklig tillförsel av energi kan elektroner flytta sig från ett skal till ett annat med en större energinivå. När denna förflyttning sker får atomen ett överskott eller ett underskott av elektroner och blir även därmed kallad en jon. En negativ jon fås genom att atomen har flera elektroner än protoner, för en positiv jon är det motsatta. Teorin om atomer och partiklar säger att en elektrisk laddning bärs av så kallade laddningsbärare, vilka är joner eller elektroner [1].

Elektrisk laddning mäts i SI-enheten Coulomb, som är uppkallad efter franska fysikern Charles-Augustin de Coulomb (1736-1806). I formel (1) nedan framställs den elektriska laddningen som mängden ström per tidsenhet som passerar en koppartråds area [2].

$$Q = I \cdot t \quad (1)$$

Där:            Q = elektrisk laddning    [C]  
                  I = ström                        [A]  
                  t = tid                                [s]

Samma person namngav även Coulombs lag, vilket är den grundläggande beskrivningen för att kraften är proportionell mot avståndet och rörelsen mellan två laddade partiklar, se ekvation (2) [2].

$$F = \frac{Q_1 \cdot Q_2}{4\pi\epsilon_0 r^2} \quad (2)$$

Där:            F = kraft                                    [N]  
                   $\epsilon_0$  = Permittivitet för vakuum =  $8,853 \times 10^{-12}$     [As/Vm]

$r = \text{radie}$  [m]

Olika laddningar mellan två partiklar, eller föremål, gör att de attraheras och partiklar med lika laddningar gör att de repelleras [3].

## 2.2 Maxwells ekvationer

Elektromagnetism är sambandet mellan de två fysikaliska fenomenen elektricitet och magnetism [4]. Även fast det var länge känt att blixnar gav upphov till magnetisering antogs det att det inte fanns något samband mellan elektricitet och magnetism. Det var inte förrän år 1820, då dansken Hans Christian Ørsted (1777-1851) hade lektion för sina studenter, som ett samband upptäcktes. Under lektionen placerade Ørsted en elektriskt ledande tråd parallellt ovanför en magnetisk nål. Ørsted och hans studenter bevittnade hur nålen, istället för att röra sig åt samma håll som strömmen i tråden, rörde sig vinkelrätt mot strömmen. Han bytte även håll på strömmen och bevittnade hur nålen återigen rörde sig vinkelrätt mot strömmen, men den här gången åt motsatt håll [2].

För att bestämma den magnetiska och elektriska flödestätheten används ekvationerna (3) och (4).

Den elektriska flödestätheten i vakuum bestäms av:

$$\mathbf{D} = \epsilon_0 \cdot \mathbf{E} \quad (3)$$

Där:  $\mathbf{E}$  = Elektrisk fältstyrka [N/C]  
 $\mathbf{D}$  = Elektrisk flödestäthet [As/m<sup>2</sup>]

Den magnetiska flödestätheten i vakuum bestäms av:

$$\mathbf{B} = \mu_0 \cdot \mathbf{H} \quad (4)$$

Där:  $\mathbf{H}$  = Magnetisk fältstyrka [A/m]  
 $\mu_0$  = permeabilitet i vakuum [H/m]  
 $\mathbf{B}$  = Magnetisk flödestäthet [T]

James Clerk Maxwell (1831-1879) gjorde många upptäckter. Men det som anses vara hans största verk var när han på ett elegant, matematiskt vis uttryckte samband mellan elektricitet och magnetism. Med hjälp av de sambanden gick det att beskriva all kunskap som på den tiden fanns om elektromagnetism. Sambanden var: Gauss lagar, Ampères lag samt Faradays induktionslag. Nedan beskrivs de fyra elektromagnetiska fenomenen, även kallade ”Maxwell’s ekvationer”.

Enligt den första ekvationen, ekvation (5), ger en elektrisk laddning upphov till ett elektriskt fält.

$$\oint_S \mathbf{D} \cdot d\mathbf{S} = Q \quad (5)$$

Den andra ekvationen, ekvation (6), säger att det inte existerar några magnetiska monopoler.

$$\oint_S \mathbf{B} \cdot d\mathbf{S} = 0 \quad (6)$$

Enligt den tredje ekvationen, ekvation (7), ger tidsvariationer i ett magnetfält upphov till ett elektriskt fält.

$$\oint_C \mathbf{E} \cdot d\mathbf{l} = \int_S -\frac{\partial \mathbf{B}}{\partial t} \cdot d\mathbf{S} \quad (7)$$

Avslutningsvis säger den fjärde ekvationen, ekvation (8), att strömmar ger upphov till magnetiska fält [2].

$$\oint_C \mathbf{H} \cdot d\mathbf{l} = \int_S \mathbf{J} \cdot d\mathbf{S} + \int_S \frac{\partial \mathbf{D}}{\partial t} \cdot d\mathbf{S} \quad (8)$$

Där:  $\mathbf{J}$  = Strömtäthet  $[A/m^2]$

## 2.3 Elektriska maskinen

Efter Hans Christian Ørsteds upptäckt av sambandet mellan elektricitet och magnetism fascinerades den engelska forskaren Michael Faraday av Ørsteds forskning. År 1821 visade Faraday upp sina två konstruerade prototyper - en upphängd strömledande metalltråd som elektromagnetiskt roterade runt en magnet och en liknande konstruktion där magnetens magnetfält roterade kring en strömförande ledning. I detta ögonblick hade han skapat en apparat som kunde, med elektromagnetismens hjälp, omvandla en energiform till en annan. I och med detta föddes den första elektriska maskinen [5]-[6].

### 2.3.1 Elektromekanisk omvandling

En elektrisk maskin kan ses som en länk där energi omvandlas mellan två olika system – ett elektriskt och ett mekaniskt system. Energi kan överföras mellan dessa om systemen sammankopplas. När energi överförs från det elektriska till det mekaniska systemet är maskinen i ett tillstånd som kallas för motordrift. I en motsatt överföring av energin, när mekanisk energi blir elektrisk, befinner sig maskinen i ett tillstånd kallad generatordrift.

Energiöverföringen mellan de två systemen kallas för elektromekanisk omvandling. Systemen binds samman via elektriska laddningar som färdas i elektriska och magnetiska fält. Förutom fälten tillkommer dessutom rörelser och krafter i omvandlingen. Det finns flera anledningar till att en elektrisk maskin är reversibel. Den främsta och viktigaste anledningen är att ett

magnetiskt fält uppstår när två magneter med motsatt polaritet närmar sig varandra. Med ett oföränderligt avstånd är fältet homogent eftersom magneterna är fixerade i statorn. När minst en strömförande ledare placeras i magnetfältet påverkas ledaren av en kraft enligt ekvation (9).

$$F = B \cdot I \cdot l \quad (9)$$

Där:  $l =$  längd [m]

Kraften,  $F$ , enligt ekvation (9) balanseras av en motkraft som kan vara vilken kraft som helst. Om ledaren rör sig med konstant hastighet induceras spänningen,  $E$ , enligt ekvation (10).

$$E = B \cdot l \cdot v \quad (10)$$

Där:  $v =$  ledarhastighet [m/s]

Detta gäller vid motordrift. Om maskinen körs i generatordrift kommer rörelsen hos ledaren att ge upphov till en inducerad spänning i ledaren som sedan motverkas av en kraft. Alltså är maskinen reversibel [7]-[8].

## 2.4 Likströmsmaskinen

Nedan beskrivs likströmsmaskinens uppbyggnad samt funktion.

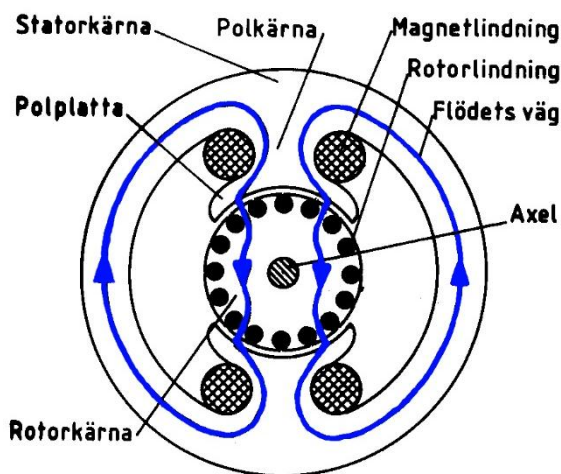
### 2.4.1 Uppbyggnad

En likströmsmaskin består av, liksom alla andra roterande elektriska maskiner, en stator och en rotor. Inuti statorkärnan finns ett antal poler, vilka är lindade med elektriskt ledande material, så kallade magnetlindningar. Magnetlindningarna magnetiseras med en likström och ett magnetiskt flöde uppkommer. Mellan statorn och rotorn finns det ett luftgap och längst ut i rotorn ligger rotorlindningar. Polplattorna är vända mot luftgapet och ser till att flödet breddas ut över rotorns yta. Antalet poler i maskinen kan variera. Oftast är det endast mycket små maskiner som har två poler och större maskiner har desto fler poler.

Det magnetiska flödet flödar genom statorn, rotorn och även över luftgapet [9]. Eftersom det magnetiska flödet vill ta den kortaste vägen genom luftgapet kommer flödet gå vinkelrätt mot rotorytan. Anledningen till att det magnetiska flödet väljer att gå kortast möjliga väg genom luften är för att luft har högre magnetisk reluktans än järn. Luftgapet är lika långt hela vägen runt rotorn, därför blir det samma reluktans kring hela rotorn. Detta leder till att den magnetiska flödestätheten är konstant överallt [10].

Rotorlindningarna består av ett antal spolrar som kan vara kopplade på olika sätt. Spolarna kallas för härvor, vilka är kopplade mot en kommutator. Mer om kommutatorns funktion beskrivs i nästa kapitel. Rotorlindningarna kan ibland kallas för ankarlindning. Mot kommutatorn ligger så kallade borstar, som består av kol eller kol med metalltillsatser. Via dessa ansluts likströmsmaskinen till yttre kretsar [9].

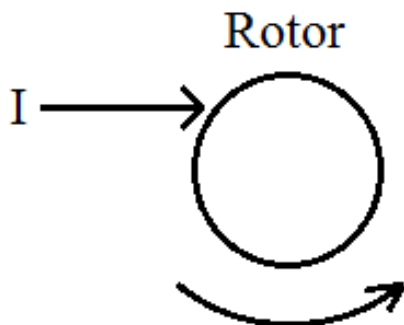
Mer detaljerat över hur uppbyggnaden av likströmsmaskinens järnkärna generellt ser ut går att se enligt figur 2.1.



Figur 2.1. Järnkärnans beståndsdelar hos en tvåpolig likströmsmaskin [9].

### 2.4.2 Funktion vid motordrift

När rotorlindningen matas av en spänning uppkommer det en ström i ankarkretsen. Eftersom strömmen flyter genom ett magnetfält ger den upphov till en kraft, se ekvation (9), som i sin tur ger upphov till ett moment. På så sätt försätts rotorn i rörelse. Kommutatorn matar kretsen med en ström som är oberoende av rotationen hos rotorn, då strömmen hela tiden matas in i samma riktning ifrån ett och samma läge. Figur 2.2 illustrerar en bild över detta. Hade strömmen varit beroende av rotationen hade en kraft åt motsatt håll erhållits vilket hade motverkat rotationen.



Figur 2.2. Principiell skiss över hur strömmatningen är oberoende av rotorns rotation.

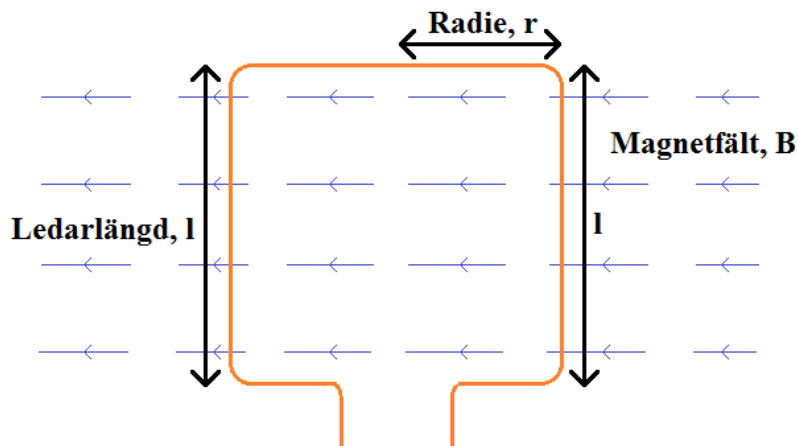
Vid rotation induceras en spänning,  $E$ , som ger upphov till en elektromotorisk kraft i motsatt riktning mot rotationens rörelse. Rotationshastigheten hos motorn mäts i antal rotationer per minut, även kallat rpm. Spänningen som induceras går att beskriva enligt ekvation (11).

$$E = 2 \cdot l \cdot B \cdot v = \frac{2 \cdot l \cdot B \cdot 2 \cdot \pi \cdot n \cdot r}{60} \quad (11)$$

Där:             $n$  = varvtal    [rpm]  
                    $2 \cdot l$  = hela ledarlängden i magnetfältet   [m]  
                    $r$  = radie hos rotor    [m]

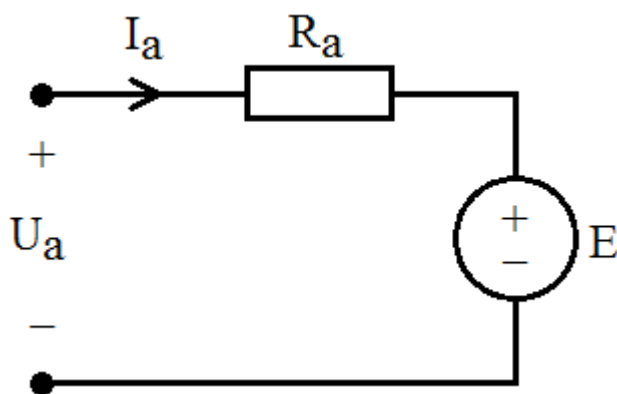


Eftersom båda sidorna hos rotorn korsar magnetfältet behöver rotorledarens längd multipliceras med två, se figur 2.3.



Figur 2.3. Principiell skiss på en rotors dimensioner i ett magnetfält.

En likströmsmotor kan beskrivas enligt figur 2.4 nedan.



Figur 2.4. Ekvivalent krettschema för likströmsmotor.

Ankarströmmen,  $I_a$ , kan vid rotation beräknas enligt ekvation 12 nedan.

$$I_a = \frac{U_a - E}{R_a} \quad (12)$$

Där:  $U_a$  = ankarspänningen [V]  
 $R_a$  = ankarlindningens resistans [Ω]

Om belastningen på maskinen ökar krävs det ett större moment ifrån den. Detta kan fås genom att minska varvtalet. Enligt sambandet i ekvation (11) minskar även då  $E$ , vilket leder till att  $I_a$  ökar, se ekvation (12). Detta leder sedan till att krafterna som verkar på ledarna, se ekvation (9), ökar och de ger då upphov till ett större moment.

Då kraften är proportionell mot strömmen och det magnetiska flödet går vridmomentet,  $M$ , att uttrycka som:

$$M = k_M \cdot \phi \cdot I_a \quad (13)$$

Där:  $k_M$  = maskinkonstant  
 $\phi$  = magnetiskt flöde [Wb]

Maskinkonstanten beror på antalet rotorlindningar, lindningarnas längd med mera. Därför maskinkonstanten unik för varje maskin.

Den inducerade spänningen,  $E$ , går att beskriva som:

$$E = k_E \cdot \phi \cdot n \quad (14)$$

Där  $k_E$  = annan maskinkonstant

Sambandet gäller eftersom den inducerade spänningen är direkt proportionell mot varvtalet och magnetflödet. Därutöver tillkommer även här en maskinkonstant,  $k_E$ , som beror på antalet seriekopplade härvor och lindningarnas längd.

Om den inducerade spänningen i ekvation (12) substitueras genom insättning i ekvation (14) fås följande samband:

$$I_a = \frac{U_a - k_E \cdot \phi \cdot n}{R_a} \quad (15)$$

Löses varvtalet ut ur ekvation (15) fås ekvationen nedan som kallas för varvtalsekvationen.

$$n = \frac{U_a - R_a \cdot I_a}{k_E \cdot \phi} \quad (16)$$

Varvtalsekvation kan dock behöva kompletteras med borstspänningsfallet,  $U_{borst}$ , vid lägre driftspänningar för att tillräcklig noggrannhet ska erhållas. Borstspänningsfallet beror på materialet hos borsten och kan typiskt ligga på ca 1V/borste för kolborstar eller ca 0,3V/borste för kol med metalltillsatser. Med borstspänningsfallet blir varvtalsekvationen [11]:

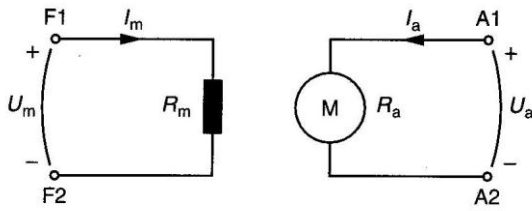
$$n = \frac{U_a - R_a \cdot I_a - U_{borst}}{k_E \cdot \phi} \quad (17)$$

## 2.5 Magnetisering av likströmsmotorer

Likströmsmotorer kan indelas i tre olika magnetiseringssätt: separatmagnetiserad, seriemagnetiserad och komppoundmagnetiserad. Skillnaderna mellan dessa är hur magnetlindningarna är kopplade i förhållande till ankarkretsen. Eftersom testriggen skall, teoretiskt sett, kunna använda sig av vilken likströmsmotortyp som helst behandlas alla magnetiseringssätt nedan.

### 2.5.1 Separatmagnetisering

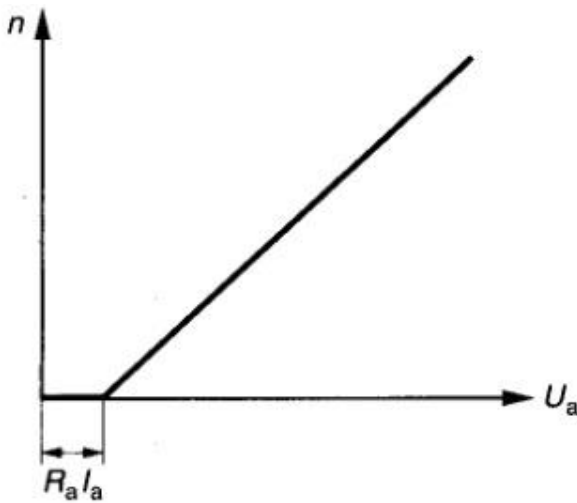
Den separatmagnetiserade motorn är den typ som används mest inom industriellt bruk. Den kallas separatmagnetiserad därför att magnetiseringskretsen och ankarkretsen ej är sammankopplade, se figur 2.5 nedan.



Figur 2.5. Ekvivalent schema för separatmagnetiserad likströmsmotor. Till vänster visas magnetiseringskretsen och till höger ankarkretsen [11].

Driftfallen som används med denna typ av motor är ankarspänningsreglering, fältreglering samt konstant ankarspänning och konstant magnetiseringsspänning.

Vid ankarspänningsreglering är det, som namnet antyder, ankarspänningen som varieras. Magnetiseringsspänningen,  $U_m$ , och belastningsmomentet hålls konstanta. Vid konstant magnetiseringsspänning blir magnetflödet också konstant. Detta leder i sin tur, enligt ekvation (13), till att ankarströmmen är konstant. Eftersom alla variabler i varvtalsekvationen förutom ankarspänningen och varvtalet hålls konstanta fås ett linjärt samband mellan dessa storheter. En avvikelse från linjäriteten uppkommer dock vid varierande belastningsmoment. Det beror på att spänningsfallet som fås från produkten av ankarresistansen,  $R_a$ , och ankarströmmen,  $I_a$ , ändras. Sambandet mellan varvtalet och ankarspänningen går att se i figur 2.6 nedan.

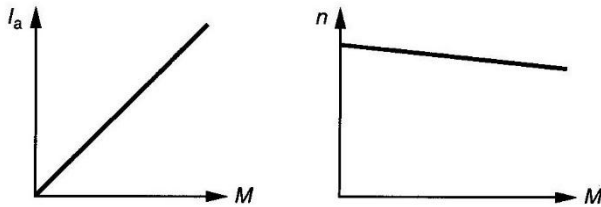


Figur 2.6. Sambandet mellan varvtal och ankarspänning [11].

Fältreglering kallas det när magnetiseringsspänningen varieras och ankarspänningen hålls konstant. Den kallas för fältreglering eftersom magnetfältets flödesstorlek ändras vid varierande  $U_m$ . Om magnetiseringsspänningen minskas kommer magnetiseringsströmmen,  $I_m$ , också minska, vilket i sin tur leder till att det magnetiska flödet minskar. Enligt varvtalsekvationen, ekvation 16, ökar varvtalet om det magnetiska flödet minskar. Momentekvationen, ekvation 13, säger att om en konstant ankarström önskas och det magnetiska flödet minskar måste belastningsmomentet minska. Då varvtalet är omvänt proportionellt och momentet är direkt proportionellt mot det magnetiska flödet blir varvtalet omvänt proportionellt mot momentet. Detta innebär att det är en olinjär karakteristik mellan

varvtal och moment.

Sista driftfallet är när ankarspänningen och magnetiseringsspänningen är konstanta. När belastningsmomentet ökar leder det till att magnetiseringsströmmen ökar linjärt och varvtalet minskar linjärt, se figur 2.7 [11].



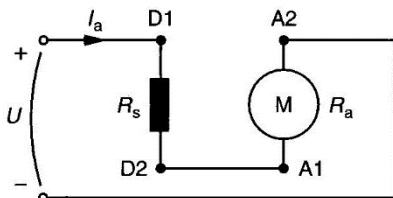
Figur 2.7. Sambandet mellan ankarström och moment samt varvtal och moment hos en separatmagnetiserad likströmsmotor [11].

### 2.5.2 Shuntmagnetisering

Det finns ytterligare ett magnetiseringssätt som kallas shuntmagnetisering, vilket är en variant av separatmagnetisering. Denna typ av magnetisering genomförs genom att  $U_m = U_a$ , realiseringen av detta görs genom att koppla magnetiseringslindningen över ankarret. Denna typ av magnetisering användes mycket under tiden då elnätet använde sig av likström, numera använder elnätet växelström [11].

### 2.5.3 Seriemagnetisering

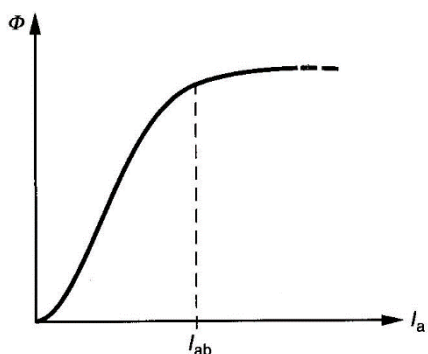
Vid seriemagnetisering är ankarret och magnetiseringskretsen seriekopplade enligt figur 2.8.



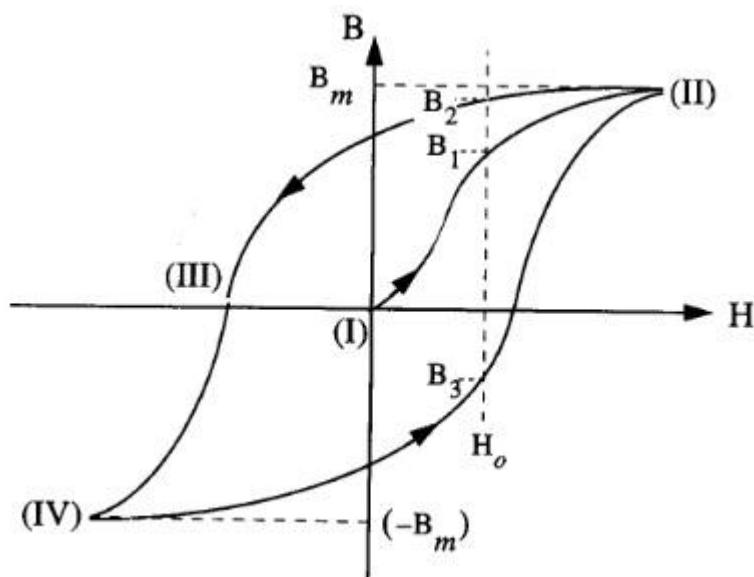
Figur 2.8. Ekvivalent schema för seriemagnetiserad likströmsmotor [11].

Seriekopplingen medför att magnetiseringsströmmen blir lika med ankarströmmen. Serieresistansen,  $R_s$ , och ankarresistansen är nära varandra i storleksmässigt, vilket innebär att de båda är relativt små [11].

Järnets karakteristik gör att förhållandet mellan ankarströmmen och det magnetiska flödet ej är linjärt, trots att storheterna är proportionella mot varandra, se figur 2.9. Det beror på en tröghet som existerar vid magnetisering av ferromagneter. Trögheten ger även upphov till hysteres, som illustreras i figur 2.10 [4].



Figur 2.9. Förhållandet mellan det magnetiska flödet och ankarströmmen i en seriemagnetiserad likströmsmotor [11].



Figur 2.10. Kurva över hysteres vid magnetisering av ferromagneter. Vid påbörjad matning startar kurvan vid (I) för att röra sig mot (II), där magnetisk mättnad uppstår. Därefter krävs ett negativt H för att B ska återgå till noll (III) för att sedan återigen uppnå magnetisk mättnad. Därefter rör sig kurvan från (IV) till (II) [4].

Om  $I_a < I_{ab}$  är flödet ungefär produkten av en konstant,  $k$ , och  $I_a$ . Om  $I_a > I_{ab}$  är flödet nästan konstant. Värdet på  $I_{ab}$  bestäms av motorns konstruktion men befinner sig vanligtvis inom  $0,8-0,9 I_{an}$ .

Detta leder till att varvtalsekvationen och momentekvationen får följande utseende om  $I_a < I_{ab}$ :

$$n \approx \frac{U_a - I_a \cdot (R_a + R_s) - U_{borst}}{k_E \cdot k \cdot I_a} \quad (18)$$

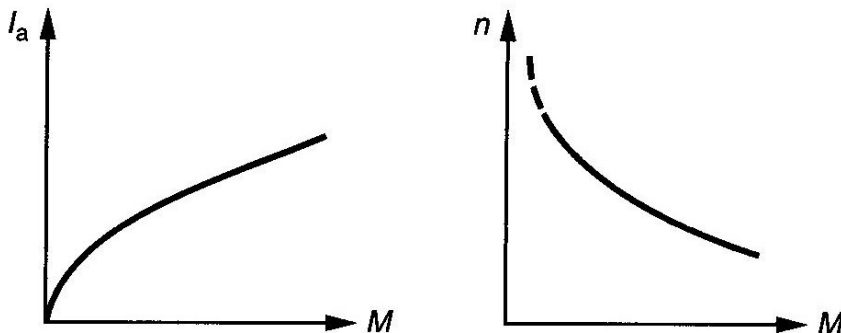
$$M \approx k_M \cdot k \cdot I_a^2 \quad (19)$$

Om  $I_a > I_{ab}$  blir det istället:

$$n \approx \frac{U_a - I_a \cdot (R_a + R_s) - U_{borst}}{k_E \cdot \phi} \quad (20)$$

Momentet kommer då bli ungefär lika med momentekvationen, se ekvation 13.

Ökas belastningen och spänningen hålls konstant kommer varvtalet på motorn att minska. Om  $I_a < I_{ab}$  leder en dubbling av belastningsmomentet till en ungefär 40 % ökning av ankarströmmen, se figur 2.11.



Figur 2.11. Momentets påverkan av ankarström (till vänster) och varvtal (till höger) i en seriemagnetiserad likströmsmaskin [11].

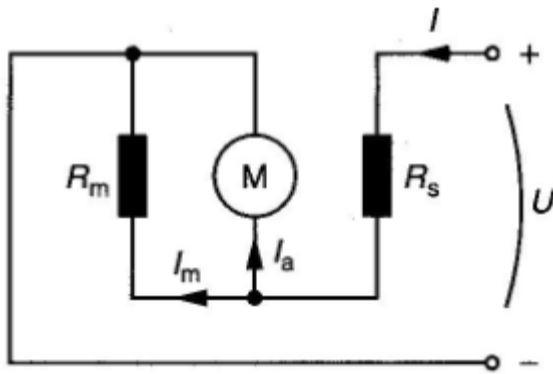
Jämförelsevis om samma förutsättningar hade rått hos en separatmagnetiserad motor hade ankarströmmen dubblerats, se figur 2.7. Det betyder att en ökning av belastningen hos en seriemagnetiserad likströmsmotor leder till en kraftig sänkning av varvtalet när ankarströmmen ökar endast något. Det betyder att lägre effekter erhålls än vad separatmagnetiserade likströmsmotorer ger.

På grund av detta regleras varvtalet bäst med hjälp av spänningsreglering.

Om motorns spänning hålls konstant samtidigt som motorn ej längre är tillkopplad mot belastningen kommer varvtalet att öka. Detta kan leda till att motorn kommer till skada. Därför är det av yttersta vikt att motorn alltid är direkt tillkopplad mot belastningen och ej med kilrem eller liknande [11].

#### 2.5.4 Kompoundmagnetiserad likströmsmotor

För att motverka problemet med ett ökande varvtal vid konstant spänning hos seriemagnetiserade likströmsmotorer kan en till magnetiseringslindning tillkopplas, se figur 2.12.



Figur 2.12. Ekvivalent schema för kompondmagnetiserad likströmsmotor [11].

De båda magnetiseringslindningarna kan antingen kopplas till att ha motverkande eller samverkande flöde, så kallad med- eller motkomponderingsmaskin.

Medkomponderingsmaskiner har en liknande karakteristik som en seriemagnetiserad likströmsmotor, dock med skillnaden att tomgångsvarvtalet begränsas.

Motkomponderingsmaskiner håller ett konstant varvtalet oavsett vad den belastas med [11].

### 2.5.5 Funktion vid generatordrift

För separatmagnetiserade likströmsgeneratorer gäller att:

$$U_a = E - R_a \cdot I_a - U_{borst} \quad (21)$$

Den inducerade spänningen bestäms likadant som för separatmagnetiserade likströmsmotorer, se ekvation (14).

Varvtalet bestäms enligt ekvation (22) nedan:

$$n = \frac{U_a + R_a \cdot I_a + U_{borst}}{k_E \cdot \phi} \quad (22)$$

Ankarspänningen kan hållas konstant vid både belastnings- och varvtalsändringar eftersom magnetiseringsspänningen är varierbar [11].

## 2.6 Växelströmsmaskinen

Maskinens flexibla omvandling av energi gör den väldigt användbar, både i industrin och i hemmet [12]. Det har bidragit till att maskinen har utvecklats för att anpassas efter elnätet. Anpassningen har gjort att den existerande likströmsmaskinen fick ett nytt sällskap av en annan maskin som kunde hantera växelspanning – växelströmsmaskinen. Denna maskin kan som likströmsmaskinen arbeta i motordrift såväl som generatordrift. Den delas in tre olika varianter beroende på användningsområde: asynkronmaskiner, synkronmaskiner, och kommutatormaskiner. Särskilt kommer asynkronmaskinen behandlas, eftersom en

noggrannare undersökning av den har genomförts i en laboration [8].

## 2.7 Asynkronmaskinen

En enkel, relativt billig, anpassningsbar, driftsäker och kraftig konstruktion har gjort asynkronmaskinen, eller induktionsmaskinen, till den vanligaste tillämpade roterande maskinen [7]. Ute i industrin, där större effekter förekommer, är asynkronmaskinen som motor att föredra. För stora delar av elnätet krävs en växelströmsmaskin med tre faser [8]-[9].

### 2.7.1 Uppbyggnad

Som tidigare nämnt existerar ett sammanhängande luftgap runt och mellan rotorn och statorn i maskinen [9].

Rotorn kan vara konstruerad på två olika sätt - kortsluten eller släpringad. Kortsluten, även kallad burlindad, rotor innebär att ledarna i spåren hos rotorn är kortslutna till varandra med en lindning. Till sitt utseende liknar den en bur, därav tillkom namnet. I det andra fallet är varje fas hos lindningen Y- eller D-kopplad, där varje lindning har en varsin släpring med placerade borstar emot sig [9], [11]. Vid motordrift är de tre faslindningarna hos rotorn kortslutna över varje släpring, vilket innebär att oavsett utförande på rotorn är funktionsprincipen densamma i asynkronmaskinen [11].

På ytan av statorkärnan sitter ankarlindningar monterade i spår, där lindningarna på statorn delas upp i tre faslindningar. Dessa utgörs av tre spolar som är monterade symmetriskt till varandra med ett 120 graders mellanrum. På så vis när lindningarna tillkopplas till elnätets huvudspänning respektive fasspänning, kommer spänningarna, såväl som strömmarna, vara fäskjutna i tiden. Detta skapar ett magnetiskt flöde som roterar längs luftgapet som så småningom orsakar induktion [9].

### 2.7.2 Funktion vid motordrift

Till skillnad från statorn förses inte rotorn med någon spänning. Istället induceras en elektromotorisk kraft genom det roterande magnetiska flödet. Flödet vrider sig med en hastighet motsvarande det synkrona varvtalet,  $n_s$ . Hur många varv som magnetfältet roterar avgörs av frekvensen hos nätet och antalet poler i maskinen, enligt formel (23) nedanför [11].

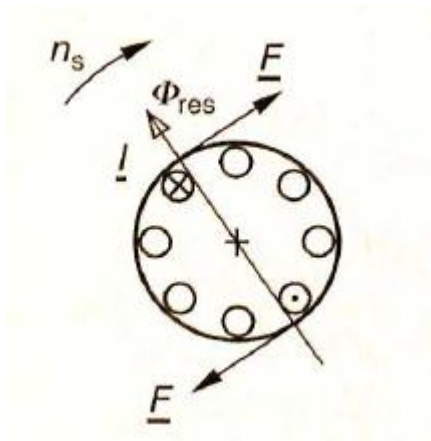
$$n_s = \frac{f}{p/2} \quad (23)$$

Där:  $n_s$  = synkrona varvtalet [r/s]  
 $f$  = frekvensen [Hz]  
 $p$  = poltalet

När rotorn förs mot flödet, bryts det, vilket ger upphov till en spänning som ger en ström i rotorlindningarna. I lindningarna, där strömmen flödar, skapas en kraft. Genom att kraften är riktad åt ett håll vill den motverka sig själv. Detta alstrar en induktion och rotorn börjar rotera



efter det roterande magnetiska flödet. Figur 2.13 illustrerar principen för detta, där en maskin i motordrift vars rotor vrider sig efter det roterande flödet när en kraft induceras.



Figur 2.13. En kraft skapas i samband med att en rotor förs mot ett roterande magnetiskt flöde. Detta ger upphov till att en ström alstras i rotorlindningarna. Rotorn kommer att vilja rotera efter flödet [11].

Friktionen i lager och ventilation i rotorn orsakar att en fördröjning uppstår mellan det magnetiska flödet och rotorns läge. Det innebär att hastigheterna hos dessa aldrig kan bli lika stora. Därmed inte heller varvtalen, där det synkrona varvtalet är alltid större än varvtalet hos motorn. Detta fenomen kallas för eftersläpning och kan beskrivas med ekvationen (24). Om rotorn hade följt flödet, det vill säga likadan rotationsriktning och hastighet som det roterande magnetiska flödet, skulle kraftverkan upphöra i och med att varken vridmoment eller induktion uppnås.

$$s = \frac{n_s - n}{n_s} \quad (24)$$

Där:  $s$  = eftersläpning [r/s]  
 $n_s$  = synkrona varvtalet [r/s]

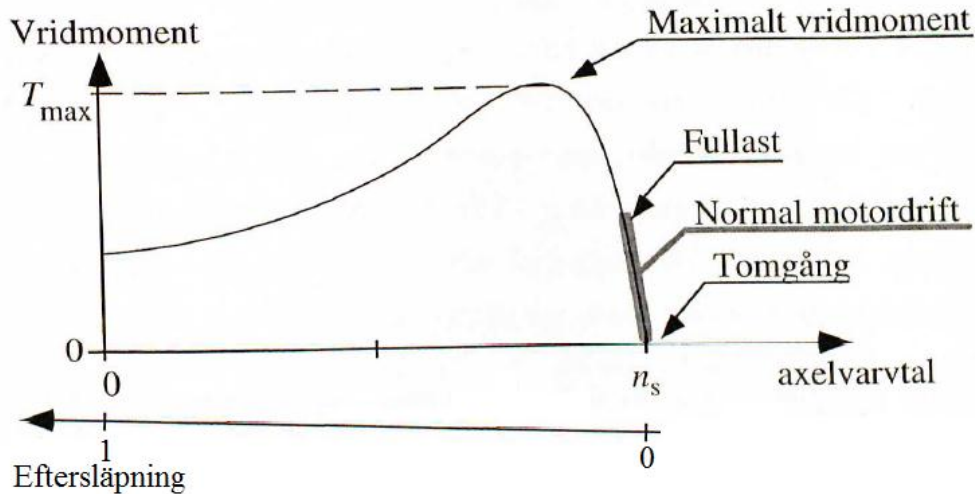
### 2.7.3 Momentkurva för asynkronmotor

När motorn befinner sig i tomgång är eftersläpningen liten, eftersom momentet som skapas ska övervinna friktionen som finns i luften och kullagret inuti i motorn. I samband med detta är det synkrona varvtalet nästintill lika stort som motorns varvtal enligt uttryck (24).

I det andra fallet, när motorn kopplas till en last, krävs det ett större vridmoment. Motorns varvtal skulle minska och eftersläpningen öka enligt formeln. Vidare skulle induktionen av spänningen, liksom strömmen i rotorns lindningar och kraften bli större. I och med det ges ett större vridmoment och jämvikt nås, där jämviktens varvtal är lägre än tomgångsvarvtalet. Eftersläpningen gör att motorn alltid arbetar i ett asynkront tillstånd, vilket gav den namnet asynkronmotor [9], [11].

En asynkronmotors moment kan beskrivas som en funktion av varvtalet. Hur förhållandet ser

ut beror på utförandet av rotorlindningen. Det finns två utföranden av rotorn vilket betyder att två kurvor skulle kunna illustreras, dock eftersom kurvorna för båda är snarlika har de antagits till att vara likadana. I figur 2.14 illustreras en kurva som beskriver sambandet mellan momentet och varvtalet hos en asynkronmotor [9].



Figur 2.14. Samband mellan moment och varvtal för en asynkronmaskin [13].

Figuren ovan visar vilka varvtal en motor kan ha. Den kan ha samtliga förutom det synkrona varvtalet, eftersom det skulle ge en eftersläpning som är lika med noll. När motorn startas är dess vridmoment relativt hög och ökar i samband med att varvtalet ökar. Då varvtalet ökar närmar sig motorn successivt det maximala vridmomentet. Vridmomentet börjar sedan minska när varvtalet hos motorn närmar sig det synkrona varvtalet och blir noll när eftersläpningen är lika med noll [9], [11].

## 2.8 Synkronmaskinen

För att kunna generera höga elektriska effekter till elkraftsnätet är synkronmaskinen den mest lämpliga kandidaten som kraftverksgenerator [7]. Maskinen kan arbeta i såväl generator- och motordrift men brukas mest som generator. Den är utförd på två olika sätt, antingen med trefas eller enfas. I detta arbete läggs fokus på den trefasiga maskinen eftersom den är den mest förekommande. Därutöver förekommer den enfasiga endast i vissa specialfall, exempelvis där den fungerar som en kraftgenerator till järnvägsnätet, vilket inte är relevant för detta projekt [9], [11].

### 2.8.1 Uppbyggnad

Precis som roterande maskiner i allmänhet, är synkronmaskinen uppbyggd av en rotor och en stator [7]. Den likströmsledande lindningen i rotorn kan vara uppbyggd på två sätt, antingen med utpräglade poler i rotorn eller att rotorn är cylindrisk [8]. Rotorn innehåller en lindning som matas med en likström. Utformningen av polerna i synkronmaskinen är nästan densamma som i en likströmsmaskin. Statorn påminner om asynkronmaskinen till sin uppbyggnad eftersom den också har tre faslindningar [9], [11].

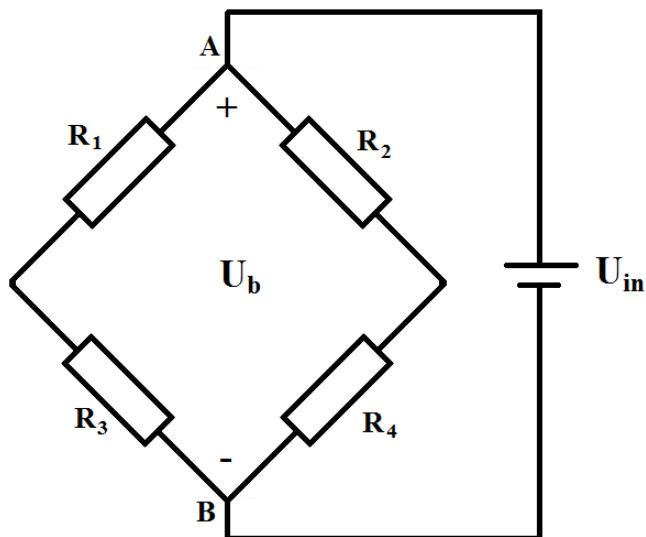
## 2.8.2 Funktion vid motordrift

Precis som asynkronmaskinen skapas ett roterande magnetisk flöde i synkronmaskinen då trefasnätet är kopplat till dess statorlindningar, vilka är fasförskjutna till varandra. Vid drift existerar ingen eftersläpning, det vill säga rotorn rör sig efter det roterande magnetiska flödet. Då varvtalet hos motorn är detsamma som det synkrona varvtalet beskrivs rörelsen mellan rotor och flödet som synkront, därav namnet till maskinen [11]. Varvtalet hos motorn är konstant och oberoende av lastens storlek, om magnetiseringsspänningens frekvens hålls konstant [8].

## 2.9 Wheatstonebrygga

En wheatstonebrygga är en resistiv givare, bestående av en formation av fyra tillkopplade motstånd i en elektrisk krets. Resistanserna i kopplingen kan vara konstanta eller variabla. Beroende fördelningen av konstanta respektive variabla resistorer kallas kopplingen antingen för kvarts-, halv- eller helbrygga.

Figur 2.15 visar ett krettschema över hur bryggkopplingen ser ut till sin uppbyggnad och ekvation (25) beskriver dess funktion [14]-[15].



Figur 2.15. Krettschema för en wheatstonebrygga.

$$U_b = \left( \frac{R_2}{R_2 + R_4} - \frac{R_3}{R_3 + R_1} \right) \cdot U_{in} \quad (25)$$

Där:  $U_b$  = Utspänning [V]  
 $U_{in}$  = Inspänning [V]  
 $R_1$ - $R_4$  = godtycklig resistans [ $\Omega$ ]

När utspänningen är 0 V är kretsen balanserad, det vill säga spänningsskillnaden mellan nod A och B tar ut varandra.

Det finns ett flertal användningsområden för wheatstonebryggor. Vanligast är att den brukas

som en lastcell, vilken kan vara utförd på olika vis [16]. I detta projekt används en knapputförd lastcell. Varje resistans i wheatstonebryggan har ersatts med trådtöjningsgivare. Med denna konfiguration i kretsen kan små variationer av motståndens resistanser, med hög noggrannhet, upptäckas när en kraft påverkar lasten. På så vis kan utspänningen, vars storlek påverkas av resistansändringen när en kraft tillförs mot givarna, bestämmas i enlighet med sambandet (26) nedan [15].

$$U_b = \frac{dr}{4 \cdot R + 2 \cdot dR} \cdot U_{in} \quad (26)$$

## 2.10 A/D-omvandlare

I dagens moderna elektronik har digitaltekniken fått allt större utrymme i flera användningsområden vid sidan om den analoga tekniken. Signaler behandlas numera som analoga eller digitala, därför är det viktigt att dessa signaler kan omvandlas däremellan. Med hjälp av den elektroniska komponenten A/D-omvandlare möjliggörs omvandlingen av signaler. ”AD” står för analog-till-digital [17].

När ett modernt digitalt instrument, exempelvis en multimeter eller en dator, mäter en analog signal med kontinuerliga värden samplar den samtidigt signalen. Det vill säga med jämna mellanrum tar den stickprov av signalen. Vid sampling av en signal blir den digital med diskreta värden. Med A/D-omvandlaren kan de samplade analoga värdena omvandlas till heltal, vilka sedermera görs om till binära tal (1 och 0) som digitala instrument kan tolka. A/D-omvandlaren matas med en referensspänning,  $U_{ref}$ , och antalet nivåindelningar av den uppmätta spänningen bestäms av  $2^n$ , där  $n$  är antalet bitar hos omvandlaren. Dessa parametrar påverkar avståndet mellan spänningsnivåerna eller vad upplösningen,  $\Delta U$ , blir hos omvandlaren, enligt ekvation (27).

$$\Delta U = \frac{U_{ref}}{2^n} \quad (27)$$

Sista nivån brukar ej räknas med då ett mindre spänningsfall, som uppstår av förluster, förhindrar spänningen till att nå till sitt maximala värde. Istället är nivåerna numrerade från 0 upp till den ”näst sista” nivån. Med detta i åtanke kan den analoga signalens motsvarande närmaste heltal  $D_{ut}$  uppskattas, efter formel (28). Där  $A_{in}$  representerar värdet av det sampel som kommer in till A/D-omvandlaren. Oavsett vilket sampelvärde det är i ett intervall blir heltalet alltid detsamma [14].

$$D_{ut} = \text{round} \left[ \frac{A_{in}}{\Delta U} \right] \quad (28)$$

## 2.11 Elsäkerhet

I samband med arbete med elektricitet finns det alltid en risk för att olyckor kan uppstå. Olyckorna kan vara så pass farliga att någon blir allvarligt skadad eller avlider. Faran hos sådana olyckor ligger i strömmen, vilken går genom kroppen när ett spänningsatt föremål berör kroppen. I detta avsnitt beskrivs vilka risker det finns med strömmen [18].

Parametrar som avgör hur pass allvarliga personskador blir vid elolyckor:

- Strömmens storlek genom kroppen
- Tiden som strömmen befinner sig i kroppen
- Kretsens totala resistans

Strömmens väg genom kroppen är ingen avgörande parameter eftersom hjärtat ändå blir kritisk påverkat oavsett var strömgenomgången sker i kroppen.

För att en ström ska kunna passera kroppen krävs att den och en spänningsförande del blir tillsammans en sluten krets. Detta gör att spänningen från spänningsförande del läggs parallellt över kroppen och en del av strömmen går genom kroppen, eftersom kroppen har en inre resistans [11].

## **2.12 Arduino**

Arduino är ett mikrokontrollerkort som har ett eget programmeringsspråk. Språket kallas ”Arduino Programming Language” och är baserat på C/C++. Arduino har en öppen källkod och öppen hårdvara. Med öppen källkod och programvara menas det att vem som helst kan tillverka, modifiera eller sälja just den designen eftersom den är tillgänglig för allmänheten.

En Arduino kan detektera olika inputs så som knapptryckningar eller indata från olika typer av sensorer och göra dessa till outputs. Exempel på outputs är stopp och start av motor eller tändning av en lampa [19].

### **2.12.1 Olika typer och deras skillnader**

Det finns ett flertal olika modeller av Arduino. Det som skiljer modellerna åt kan exempelvis vara hastigheten på processorn, antalet analoga och digitala in- och utportar, mängden minne eller inspänningsnivå [19].

I det här projektet används modellen Arduino Uno samt Arduino Mega eftersom de har tillräckligt med analoga inportar. Arduino Uno är även den mest använda modellen och har därför gott om dokumentation kring sig. Hur Arduino Uno ser ut kan beskådas i Figur 2.15.

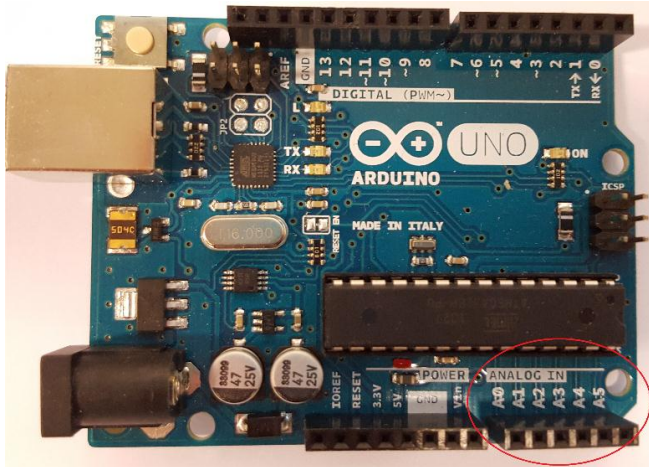
### **2.12.2 Arduino Shields**

Arduino Shields är ett kretskort som kan tillkopplas ovanpå en Arduinoenhet för att ge den ytterligare funktioner. Exempel på dessa funktioner är trådlös kommunikation mellan olika Arduinos, styrning av motorer eller WiFi [19].

### **2.12.3 A/D-Omvandlaren i Arduino**

För att ett inläst analogt värde, från exempelvis en sensor, ska kunna utnyttjas i en mikrokontroller är det angeläget att kunna omvandla detta värde till ett digitalt. I de flesta mikrokontroller, inkluderat Arduino, finns det en inbyggd A/D-omvandlare som gör denna omvandling.

Det finns specifika portar på en mikrokontroller som genomför A/D-omvandlingen. I Arduinos fall finns det sex portar med namnen A0-A5, se den röda cirkeln i figur 2.16, som kan göra omvandlingen [20].



Figur 2.16. ANALOG IN-portarna A0-A5 har förmågan att omvandla ett värde från analogt till digitalt.

Antalet analoga spänningsnivåer varierar med antalet bitar som kan läsas in till en mikrokontroller. Antalet bitar som Arduinos omvandlare har är 10-bitar, vilket innebär att det finns 1024 spänningsnivåer som kan mätas fram. Med känt antal nivåer och en referensspänning på 5 V till AD-omvandlaren kan upplösningen hos den beräknas, enligt formel (29) [14].

$$\Delta U = \frac{5 \text{ V}}{2^{10}} = 4,88 \text{ mV} \quad (29)$$

Genom omskrivning av ovanstående ekvation kan ett förhållande mellan det digitala värdet,  $D_{ut}$ , från omvandlingen och det uppmätta analoga spänningsvärdet, eller samplers värde, beskrivas med uttrycket (30) nedan.

$$\frac{U_{ref}}{2^n} = \frac{A_{in}}{D_{ut}} \quad (30)$$

Förhållandet gör att A/D-omvandlaren uppfattar att referensspänningens värde är proportionellt mot värdet hos det digitala värdet. Det betyder att Arduinos referensspänning mellan 0-5 V motsvarar de numeriska värdena 0-1023, eftersom det finns 1024 nivåer [14], [20]-[21].

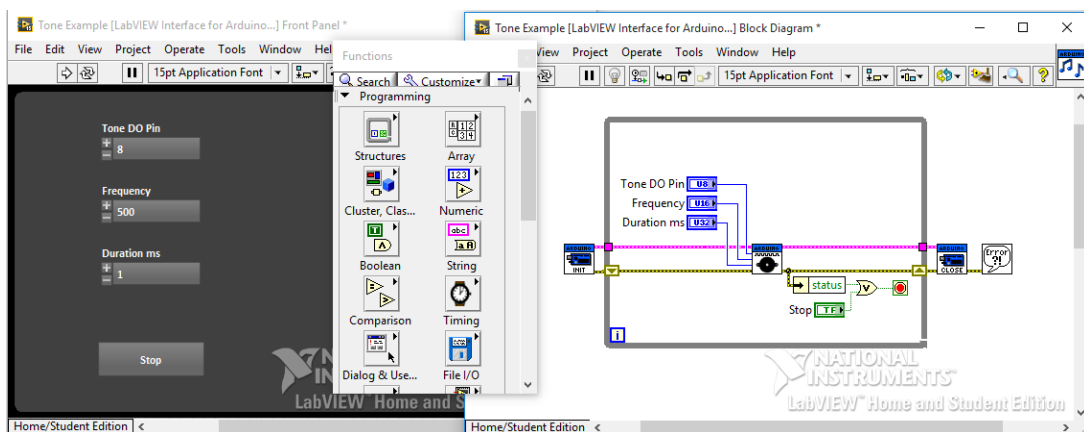
## 2.13 LabVIEW

År 1986 lanserade företaget National Instruments ett verktygsprogram där användaren kunde programmera i en miljö där kodsyntaxen bestod av grafiska objekt och animationer i alla dess varianter och färger. Det grafiska språket, kallat "G", skiljde från det sedvanliga textbaserade programmeringsspråket. Företaget namngav programmet Laboratory Virtual Instrument

Engineering Workbench eller förkortningen LabVIEW. Idag är programmet väletablerat; mestadels eftersom National Instruments inte enbart skapade programmet för Apples dator Machintosh utan även för Microsofts Windows några år efter lanseringen. Därför har programmet många användare inom både universitet och industrier [22].

### 2.13.1 Vad är LabVIEW?

När programmeringskod ska skrivas i LabVIEW görs det i två olika miljöfönster – front panel och block diagram, vilka illustreras i figur 2.17. Front panel är ”ytterskalet” av programmet där indikatorer, som exempelvis mätare och utritade grafer, samt kontroller, som knappar och switchar, kan programmeras i. Det andra fönstrets blockdiagram kan ses som koden ”under skalet”. Där sätts indikatorerna och kontrollerna ihop med ledningar på olika sätt beroende på programmets uppgift. För att programmera i båda fönstren har användaren en ”palett” innehållande alla tänkbara funktioner och verktyg, vilka anpassas efter frontpanelers eller blockdiagramms arbetsmiljö. För att använda en viss funktion ”dras den ut” från paletten in till det aktuella fönstret. Fönstren har sammankoppling till varandra, det vill säga vad som ändras i det ena fönstret påverkar det andra och vice versa [23].



Figur 2.17. Detta är en exempelkod som visar frontpanel-fönstret till vänster och blockdiagram-fönstret till höger. Med paletten, som syns i det lilla fönstret, kan kod programmeras.

I den senaste programversionen LabVIEW 2015 kan användaren programmera och visualisera på ett mer kreativt och enklare sätt än i sin föregångare, LabVIEW 2014. Det optimerade programmet gör den senaste versionen nästan tio gånger snabbare och har ett flertal nya smartare verktyg/funktioner. Där LabVIEW 2015 även är kompatibel med de tidigare funktionerna ur äldre versioner av LabVIEW. Exempelvis funktioner som att skapa egna snabbmenyer, öppna upp eller förbättra ytor i miljöfönstren, standardisera hyperlänksadresser med mera. Detta öppnar upp för att stora invecklade system kan bemästras effektivare och samtidigt som tid och pengar inte går förlorade [24]-[25].

Trots att LabVIEW inte har ett traditionellt programmeringsspråk känner många erfarna programmerare igen sig i programmet. Främst för att språket liknar de funktioner och strukturer som förekommer i andra textbaserade programmeringsspråk. Vad som skiljer dessa

språk med språket G är exekveringen. Istället för att listade kodrader utförs sekventiellt bestäms ordningen av ett dataflöde som går mellan uppbyggda noder. Vid simulering och felsökning kan dataflödet följas med noggrannhet som övriga språk saknar. Dessutom kan andra språk skrivas och integreras i LabVIEW, bland annat Arduino. Parallella processer kan nås av sektioner som byggs upp separat i programmet, vilket inte existerar i traditionell programmering [26].

Det är viktigt att ha i åtanke att andra verktygsprogram, exempelvis C, inte är bättre än LabVIEW och vice-versa. Vilket program som bäst lämpar sig för lösningen av en uppgift beror i synnerhet på hur uppgiften ser ut. Projektets ändamål handlar om skapandet av ett program som ska utföra mätningar vilket innebär att användandet av LabVIEW kan vara fördelaktigt [27]. Andra fördelar är att signaler kan mätas in från kretsar i kombination med att matematiska algoritmer beräknas i realtid [28].

## **2.14 MATLAB**

Matematikern James H. Wilkinson och hans forskarkollegor, George Forsythe och John Todd, hade under flera år utvecklat på ett program. Programmet som bland annat var baserat på Alan Turnings algoritmer, kunde utföra matematiska beräkningar. Linjära algebraiska problem, bland annat vektorer såväl matriser, kunde enkelt och effektivt lösas i en laborativ miljö. Därav fick programmet sitt namn – Matrix Laboratory eller förkortningen MATLAB.

Sedan första versionen av programmet släpptes av MathWorks, år 1984, har populariteten hos användarna växt världen över [29]. Under de tidiga åren var programmet endast lämpligt för forskare och ingenjörer och numera är den tillgänglig för studenter såväl som för privatpersoner. Eftersom programmet förekommer överallt finns även fakta kring programmet på Internet, där bland annat MathWorks har både support och forum för sina användare [30]-[31].

### **2.14.1 Mer om MATLAB**

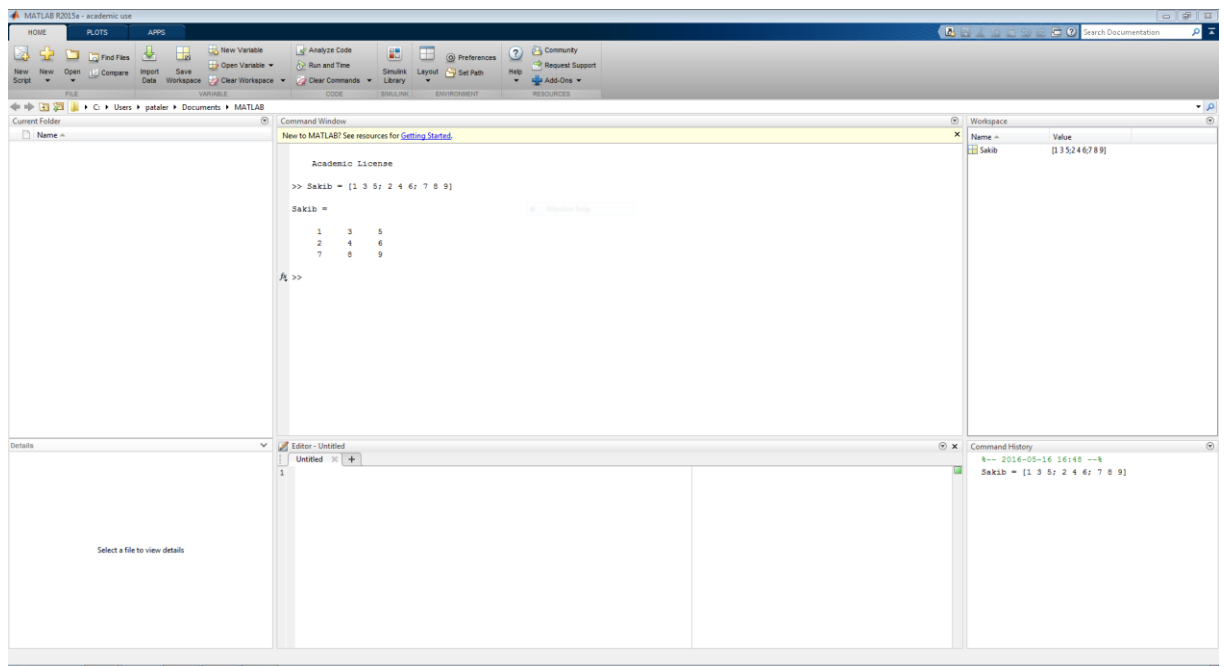
Programmet MATLAB är en kraftfull plattform där användaren får lösa problem och skapa lösningar med hjälp av matematiska verktyg och programmering. Programmets funktioner gör att beräkningar inom de matematiska grenarna matematisk analys och linjär algebra kan utföras.

Språken, Java, Fortran och C++ förekommer i MATLAB. Dessa kan användas för att skapa figurer, grafer och bilder vilket underlättar presentationen av insamlad data. Dessa kan illustreras i figurer som antingen är 2-dimensionella eller 3-dimensionella. Programmet lämpar sig för många områden och har särskilda funktioner som underlättar kopplingen mellan plattformen och områden som statistik, reglerteknik och signaler [31]. Den bidragande orsaken till att programmet dominerar marknaden är den enkla utvecklingsmiljön.

När användaren startar MATLAB har fönstret utseendet enligt figur 2.18. I fönstret finns flera mindre fönster där samtliga har sin funktion. Till vänster finns arbetskatalogen, ”Current



folder”, som visar i vilken mapp som den aktuella MATLAB-filen som bearbetas ligger i. Nedanför finns ”Details” som uppvisar mer information om en vald fil. I mitten är kommandofönstret, ”Command Window”, placerad där kommandon, syntaxer och algoritmer skrivs in. Där visas även händelserna i programmet. Under ”Command Window” finns fönstret ”Editor” där mindre program kan skapas. Alla kommandon i ”Editor” sparas och försvinner inte lika lätt som det gör i kommandofönstret. Till höger, i variabelbläddraren ”Workspace”, listas de skapta variablerna som har förekommit under programmets körning. Nedtill finns kommandohistorien, ”Command History”, där MATLAB visar tidigare inskrivna kommandon. Viktigt är att påpeka att de mindre av dessa fönster kan flyttas och några av dem kan döljas efter användarens behov.



Figur 2.18. Utvecklingsmiljön i MATLAB.

MATLAB används inom flera områden. I rymdindustrin används den för farkosters flygsystem. Inom fordonsindustrin optimeras bilars säkerhet. Allteftersom förlusterna ska minskas i elkraftsnät samt att styrning av system blir mer digitaliserad tillämpas MATLAB. Inom telekombranschen är programmet vanligt förekommande [30].

### 3. METOD

Arbetet inleddes med informationsinsamling om elektromagnetism, elektriska maskiner, givare och Arduino. Insamlingen kring elektromagnetism och elmaskiner erhöles via kurslitteratur, Chalmers biblioteks databas och tekniska rapporter. Informationssökningen om Arduino skedde via Arduinos gemenskap, då Arduino är välkänt för hobbyanvändare såväl som ingenjörer. Detta kompletterades med hjälp av högskolans personal som är kunniga inom området.

En undersökning om olika varianter av sensorer för strömmar, spänning, moment och varvtal utfördes. Givarna testades med simplifierade kopplingar på så vis att endast spänningsmatning, sensorer, Arduinoenheter och mätobjekt utnyttjades. Därutöver verifierades givarnas funktion med dessa kopplingar tillsammans med multimetrar, tachometrar, clampmetrar samt våger.

När alla storheter mättes på ett tillfredsställande vis, undersöktes det hur och med vilka program mätdatan kunde presenteras grafiskt. MATLAB användes för att övervaka ström och spänningar. För övervakning av moment och varvtal användes Arduino. LabVIEW och en Arduino Motor Shield användes för att undersöka möjligheter till motorstyrning.

Sedermera konstruerades mätriggen där alla givare kopplades till de elektriska maskinerna. Först undersöktes de elektriska maskinerna separat i tomgångsdrift. Därpå användes ena maskinen istället som generator och det undersöktes huruvida mätningen på de båda elmaskinerna utfördes korrekt.

## 4 KOMPONENTER

Följande kapitel beskriver funktionen hos olika komponenter vilka användes för test och konstruktion av mätutrustningen.

### 4.1 Transistor - IRF520

En transistor är, som bekant, en elektronisk switch med 3 ben som slås på eller av beroende på hur stor storleken är på styrströmmen. Första benet "gate" kopplas till komponenten som skall styras. Andra benet "drain" ansluts till jord och det tredje benet "source" tillkopplas en styrenhet. När styrenheten försörjer transistorn med en matningsspänning stänger den kretsen mellan jord och den styrda komponenten. Det innebär att ingen ström passerar komponenten [32]. När spänningen mellan gate och source är noll har denna transistor en maximal spänning på 100 V mellan dess drain och source. Transistorn har en maxkapacitet att operera under hög strömbelastning vid 40 A för drainanslutningen, vilket ger en maximal arbetstemperatur på 175 °C grader. Användbara tillämpningsområden är låg-och högspännings produkter, bland annat elektriska motorer, regulatorer och ljudförstärkare samt applikationer inom fordonsindustrin, såsom luftkuddar och ABS-teknik [32]-[33].

### 4.2 H-brygga - L293DNE

För att skifta spänningens polaritet används en integrerad krets kopplad till någon form av last, lämpligen en elektrisk motor [32]. Kretsen är designad att skicka dubbelriktade strömmar upp till 1 A vid 4,5 - 36 V spänning. Kretsen används mest fördelaktigt för att driva induktiva laster, såsom reläer och elektriska magneter samt stegmotorer [34]. Det finns totalt 16 ben varav fyra ben kopplas till jord, två ben till spänningskälla, två andra ben till "Enable", och åtta ben tillhör inbyggda transistorer [32].

### 4.3 Tryckkänsligt motstånd 0.5"

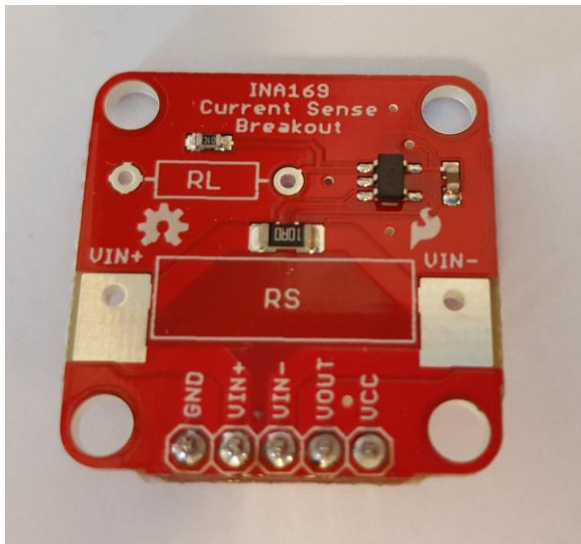
Speciell typ av potentiometer. Resistansen justeras efter trycket den utsätts för. Komponenten har längden 60 mm, där tryckytan är 12,7 mm i diameter. Lämpligen justeras resistans med tummen och pekfingeret. Trycks en hårdare kraft med fingertopparna minskar resistansen i motståndet. Standardvärdet på motståndet ligger vid 1 M $\Omega$  om ingen kraft appliceras mot komponenten. Motståndet kan maximalt och minimalt belastas med cirka 100 Nm (10 kg) respektive 1 Nm (100 g) kraft, vilka utgör mätintervallet. Hela baksidan av komponenten är täckt med klister för att underlätta monteringen på branta ytor. Utrustad med stiftanslutningar i nedre änden av det långa skaftet vilka underlättar anslutningen till experimentkortet, bland annat breadboard [35].

### 4.4 Strömsensor - INA169

Strömsensorgivaren, INA169, sitter på ett elektronikkort, tillverkat av Sparkfun. Kortet kan mäta strömmen i en krets som matas av en likspänning.

Vid inkoppling av ett shuntmotstånd,  $R_S$ , mellan  $V_{IN+}$  och  $V_{IN-}$  kan strömsensorn uppmäta resistorns spänningsfall mellan dessa. Därefter kan strömmen kalkyleras med Ohms lag. Mätintervallet av spänningen ändras beroende på vilken resistans  $R_S$  har. I INA169 sitter en

operationsförstärkare, vilken fungerar som en komparator genom att den tar differensen mellan spänningarna  $V_{IN+}$  och  $V_{IN-}$ . Differensens storlek påverkar hur stor strömmen blir ut från operationsförstärkare. Ström flyter vidare till en transistor och sedan genom en resistor  $R_L$  till jord. Spänningen över detta motstånd varierar efter spänningsskillnaden som uppstår mellan förstärkarens ingångar. Komponenten har en  $60 \mu\text{A}$  vilostrom, och dess matningsspänning får maximalt vara  $60 \text{ V}$  och minimalt  $2,7 \text{ V}$ . Totalt besitter sensorn 5 ben, där 2 standardmatningar till operationsförstärkaren ingår, en spänningsutgång, en jord och spänningsmatning. Figur 4.1 nedan visar hur sensorn ser ut [36]-[38].



Figur 4.1. Strömsensor av modell INA169.

Utspänningen,  $V_{OUT}$ , beräknas enligt nedan.

$$V_{OUT} = \frac{I_S \times R_S \times R_L}{1k\Omega} \quad (31)$$

Där strömmen genom resistansen  $R_L$  är känd och resistanserna är valda. Om det inte är önskvärt att justera mätintervallet behålls  $R_S$  till att vara  $10 \Omega$ . Den inre resistansen från transistorn uppskattas den till  $1k\Omega$  [38]-[39].

#### 4.5 Lågströmsensor - ACS712

Strömsensor med egenskapen att kunna mäta både växel- och likström. Sensorn mäter låga strömmar eftersom den mäter strömmar under  $5 \text{ A}$ . Sensorn utnyttjar halleffekten för att mäta strömmar. Internt är modulen utrustad med en förstärkare som förbättrar känsligheten hos strömmätning, framför allt då mätning av mycket små strömmar. Används lämpligen som en mätare av effektförbrukning i ett elektriskt system [40].

#### 4.6 Reflexdetektorn - QRE1113

Reflexdetektorn, QRE1113, är en mindre modul som registrerar reflektioner från ytor på ett  $0,5 - 5 \text{ mm}$  avstånd, där  $3 \text{ mm}$  ger noggrannast mätresultat [41]. Konstruktionen är uppbyggd av två delar, en infraröd LED-sändare och en fototransistor som mottagare [42]. LED-

sändaren är aktiv vid inmatningsspänningen 5 V [41]. Modulen är utrustad med en 100  $\Omega$  resistor kopplad i serie med LED-sändaren för att begränsa strömmen.

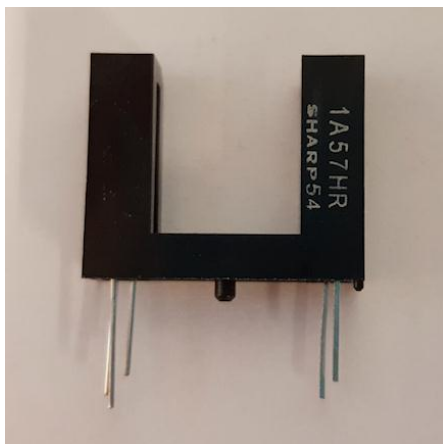
Optimal reflexion på utgången vid fototransistorn uppnås vid den tidpunkt när kondensatorn, på 10 nF, snabbt laddas ur. Detta inträffar när modulen träffar ljusa ytor, då mer reflexion uppstår jämfört med mörka ytor. Pinconfigurationen hos komponenten består av 3 ben –  $V_{CC}$ ,  $V_{OUT}$  och GND. ”QD” som finns markerad på komponenten symboliserar att dess uppbyggnad är digital, genom en inbyggd A/D-omvandlare [42].

Reflexdetektorer tillämpas bland annat för att bistå robotar med att hålla en viss kurs och position [41].

#### 4.7 Ljusbrytare - GP1A57HRJ00F

Kompakt infraröd sensor som detekterar olika föremåls rörelse. Givaren består av två stolpar där den ena är utrustad med en sändare och den andra med en mottagare. Infrarött ljus sänds från sändaren till mottagaren. Om denna stråle bryts indikeras att ett föremål har passerat mellan stolparna, vilket sensorn registrerar. Distansen mellan stolparna är 1 cm, vilket begränsar storleken hos det passerande föremålet.

Sensorn tillämpas i många användningsområden, bland annat som automatisk utmatning till pellets och identifiering av objekt i allmänhet [43]. Sensorn visas i figur 4.2.



Figur 4.2. Ljusbrytare med infraröd sändare och mottagare.

#### 4.8 LCD display - HD44780

Enheten ”HD44780” från Hitachi tillämpas för att presenterar tecken på en display. Bildskärmen initieras genom att ändra signalerna till hög eller låg. Utrustad med 16 ben kan skrivning och läsning, till och från, displayen genomföras. Två av dessa ben är anslutna mot 5 V spänning respektive jord. Inläsning sker via styrsignaler på tre andra ben. Utöver dessa ben används ytterligare åtta ben för utskrivning av tecken till displayen. Övriga ben försummas och används ej i pinconfigurationen. Totalt är skärmen begränsad till att maximalt visa upp 16 symboler på en rad. En symbol är 8x5 pixlar stor [44].

## 4.9 Lastcell - TAS606

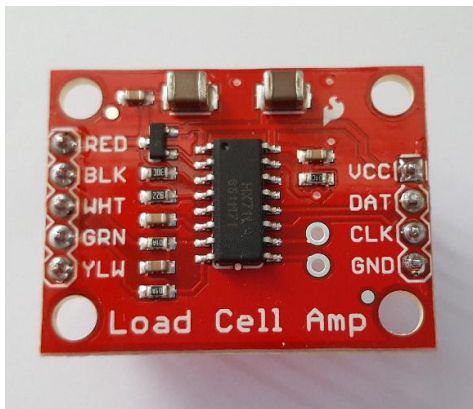
Töjningsgivare, eller lastcell, med en mobil stållegerad tryckskiva. Omvandlar en fysikalisk kraft till en elektrisk signal. Har en tryckkapacitet på 50kg. Den uppstickande, knappliknande delen av tryckskivan är området där kraften skall appliceras. Beroende på hur stort trycket är ändras resistansvärdet proportionellt, vilket resulterar i en indikation av storleken hos nedtryckningen. Fördelen med denna typ av töjningsgivare är att den enkelt kan monteras i olika konstruktioner. Givaren består av fyra töjningsgivare, vilka är kopplade som en Wheatstone-brygga [45]. Mer ingående om hur Wheatstonebryggor fungerar finns att läsa i 2.9 *Wheatstonebrygga*. I figur 4.3 illustreras lastcellen.



Figur 4.3. Lastcell av modell TAS606.

## 4.10 Lastcellsförstärkare - HX711

Förstärkare med syftet att möjliggöra avläsning av en lastcells resistansvärden för att få fram vikten eller kraften hos objektet över lastcellen. Detta fullbordas genom att ansluta förstärkaren till en mikrokontroller. Förstärkaren har totalt 9 ben, där två av dessa används för datakommunikation (DAT) och klocksignaler (CLK). Fyra andra anslutningar (RED, BLK, WHT och GRN) tillämpas för fyra töjningsgivare vilka finns inuti lastcellen och är kopplade i en Wheatstone-brygga. YLW-anslutningen används vanligtvis som skydd mot elektromagnetiska störningar genom inkoppling mot jord. Matningsspänningen (VCC) som förstärkaren kräver är mellan 2,7 V – 5 V [46]. I figur 4.4 nedan visas lastcellsförstärkaren.



Figur 4.4. Lastcellsförstärkare - HX711.

## 5 TESTER

Inför konstruktionen av mätriggen genomfördes ett antal tester för att verifiera komponenters funktion och kompatibilitet med Arduino samt möjliga framtida användning i mätriggen.

Nedan beskrivs testerna.

### 5.1 Spännings- och strömmätning

För att mäta spänning och ström i en motor utfördes flera tester för att undersöka hur det kan åstadkommas. I detta avsnitt presenteras några lösningsalternativ på sådan mätning.

#### 5.1.1 Ström- och spänningsmätning med ACS712

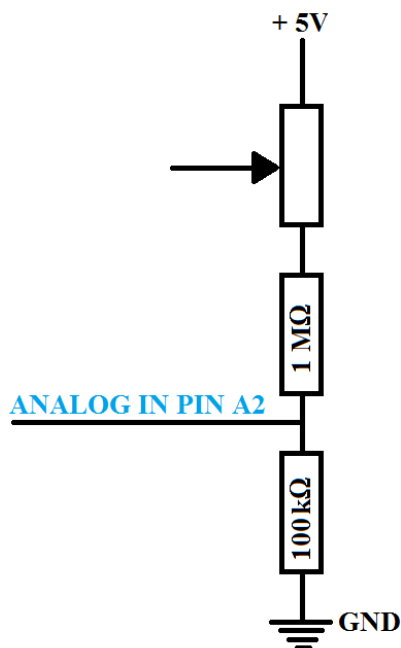
För detta test uppmättes strömflödet med lågströmsensorn, ACS712, läs mer om den i 4.6 *Lågströmsensor - ACS712*. Den uppkopplades seriellt tillsammans med en lysdiod, en resistans på  $100\ \Omega$  och ett spänningsaggregat på  $9\ \text{V}$ . Mer ingående information om hur testet genomfördes går att finna under bilaga B - *Ström- och spänningsmätning med ACS712* [47].

Sensortestet påvisade att en ström kan mätas i en krets, dock med instabila värden. Genom detta, samt att den var svårhanterlig och opålitlig, användes den inte för strömmätning av elmaskiner i mätriggen.

#### 5.1.2 Spänningsmätning i Arduino med spänningsdelning

Ett test genomfördes för att undersöka huruvida en Arduino kan mäta spänningen korrekt i en krets. Testet realiserades enklast genom spänningsdelning, då detta ledde till att en kontrollerad och hanterbar spänningsnivå kunde skickas till Arduinon.

Spänningsdelningen uppnåddes med hjälp av två resistorer som kopplades enligt figur 5.1.



Figur 5.1. Kopplingschema för spänningsmätning i Arduino med hjälp av spänningsdelning.

Spänningen mellan resistorerna skickades till Arduinoenheten via en av Arduinos analoga ingångar. Signalen som genererades var först en signal mellan 0-1023, vilket berodde på Arduinos inbyggda A/D-omvandlare. Sedan omvandlades signalen till en spänning enligt koden i bilaga A - *Voltage divider* [48].

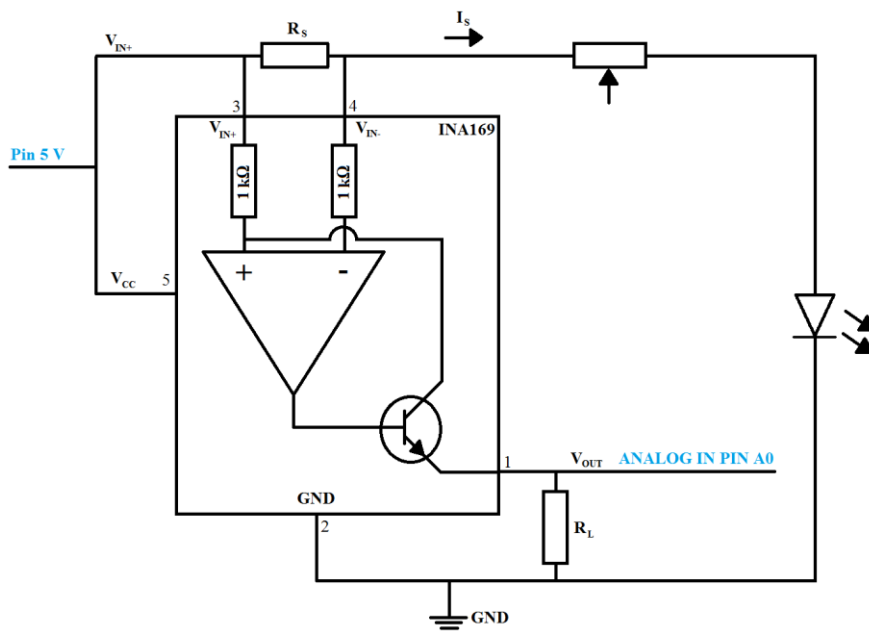
Därefter kopplades en potentiometer mellan spänningsmatningen och en av resistorerna. Detta möjliggjorde styrning av spänningen i realtid.

För att kontrollera att korrekt spänning uppmättes i Arduino, kopplades även en multimeter parallellt över en av resistorerna.

Med denna typ av konstruktion erhöles spänningsnivåer som varierade beroende på potentiometerns värde. Spänningsnivåerna begränsades av spänningsdelningen, vilket gjorde att spänningen till Arduino kunde väljas till en lägre nivå. Detta påvisade att det var fullt möjligt att mäta spänningen över en del i en krets med en Arduino, oavsett storlek på spänningen över ett mätobjekt.

### 5.1.3 Ström- och spänningsmätning med INA169

För att låga strömmar genom en ledning ska kunna mätas behövs en strömsensor med ett stort spänningsintervall och en låg vilostrom [36]. För det här testet användes en sådan sensor av modellen INA169, se 4.1 *Strömsensor-INA169*. Sensorn anslöts i en koppling där en Arduino Uno, en lysdiod och en potentiometer utnyttjades. Kretsschemat nedanför, figur 5.2, illustrerar hur kopplingen realiserades.



Figur 5.2. Hur mycket ström som passerar lysdioden kan mätas med en Arduino Uno och en INA169. Där strömmen genom lysdioden regleras med en potentiometer.

$V_{IN+}$  och  $V_{CC}$  kortslöts tillsammans till Arduinos spänningskälla på +5 V. Det spänningsfall som låg över resistorn  $R_S$  baserades på den inmatade spänningen och resistansens storlek,



vilka gav upphov till en ström  $I_S$  genom motståndet. Denna ström gick igenom mätobjektet och var därför av intresse. Ur ekvation (31) från avsnittet om 4.4 *Strömsensor - INA169* uttrycks strömmen enligt ekvation (32) nedan [49].

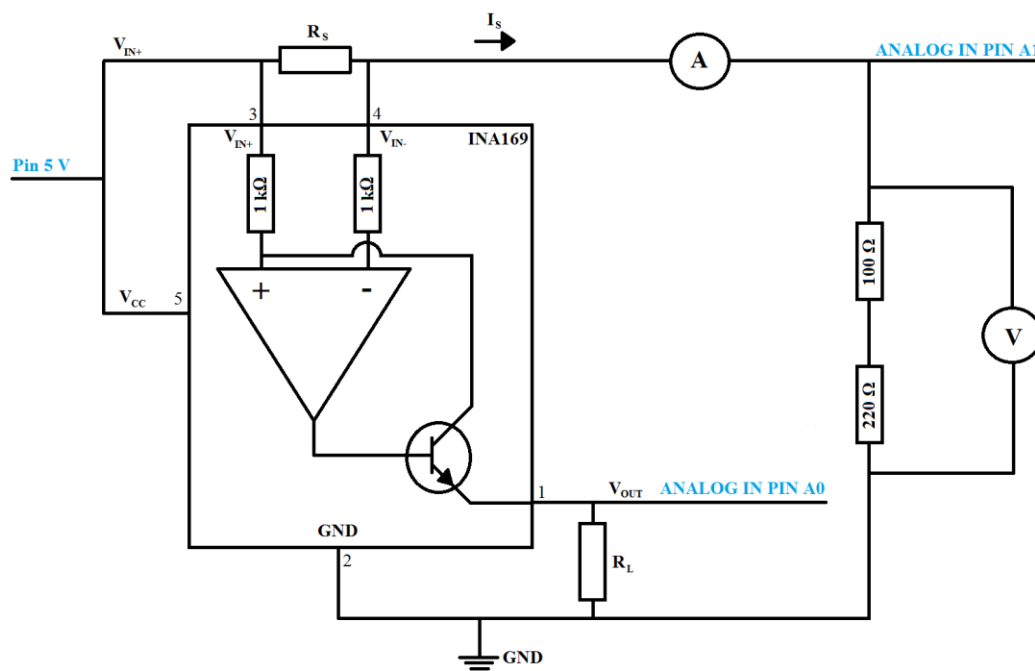
$$I_S = \frac{V_{OUT} \cdot 1k\Omega}{R_S \cdot R_L} \quad (32)$$

Strömmen flöt genom en last till jord och igenom en potentiometer med en maximal resistans på 10 M $\Omega$ . Dess resistans påverkade hur starkt den inkopplade lysdioden lyste, eftersom resistansen begränsade strömmen som flöt igenom dioden.

Utgången på strömsensorn,  $V_{OUT}$ , var ansluten till den analoga ingången, A0, som fanns på Arduinoenheten. När en ström passerade motståndet  $R_L$  till jord skapades en spänning på utgången. Sensorns GND och lysdiodens katod kopplades till samma jord [38].

Koden som användes i testet, se bilaga A - *INA169*, läste in utgångens analoga värde via A0. I ovanstående uttryck för strömmen, ekvation (32), utnyttjades värdet för att beräkna  $I_S$ . Där  $R_L$  och  $R_S$  antogs vara 10 k $\Omega$  respektive 10  $\Omega$ . Slutligen skrevs det verkliga värdet av strömmen ut på den seriella monitorn, i enheten Ampere med tre decimaler.

Potentiometern och dioden ersattes med ett par resistorer så att ett konstant mätresultat kunde erhållas. För att möjliggöra mätning av spänningen över lasten kopplades konstruktionen om enligt figur 5.3.



Figur 5.3. Kopplingsschema för ström- och spänningsövervakning med INA169 med fast värde på resistansen.

Signalen som skickades till pin A1 på Arduinon var spänningen som låg över mätobjektet.

Varje utskrift av spänningen och strömmen på den seriella monitorn fördröjdes med en halv millisekund.

Mätvärdena som presenterades på Arduinon verifierades genom att multimetrar kopplades in till kretsen, vilka mätte samma signal som Arduinon. Likheten hos dessa värden påvisade konstruktionens funktionalitet.

För att matematiskt bekräfta att värdena som skrevs ut på den seriella monitorn stämmer, tillämpades Ohms lag [38]. I detta experiment önskades även en verifiering av strömmen genom lasten  $R_L$  med hjälp av dess ekvation (32). Verifieringen utfördes genom mätning av spänningen över en resistor som var ansluten i serie med lysdioden. Motståndet som användes bestod av två seriellt kopplade resistorer på  $220 \Omega$  och  $100 \Omega$ . Med en multimeter uppmättes denna konstruktions spänning  $V_S$  över motståndet  $R_S$  till  $2,5 \text{ V}$ . Spänningen  $V_L$  över resistansen  $R_L$  och även utspänningen  $V_{OUT}$  uppmättes till  $0,26 \text{ V}$ . Strömmen  $I_S$  beräknades både med hjälp av Ohms lag samt ekvation (32). Resultaten för verifieringen presenteras i ekvationerna (33)-(34).

$$I_S = \frac{2,5 \text{ V} \cdot 1k\Omega}{10 \Omega \cdot 10k\Omega} = 0,025 \text{ A} \quad (33)$$

$$I_S = \frac{V_L}{R_L} = \frac{0,26 \text{ V}}{10 \Omega} = 0,026 \text{ A} \quad (34)$$

Enligt databladet för INA169 var felmarginalen på utgången  $\pm 2 \%$  [49]. Detta innebar att spänningen skiljde sig med några få millivolt, som var obetydligt i detta sammanhang. Mätningen gav en spänningsnivå på utgången  $2,6 \text{ V}$  med en multimeter och  $2,542 \text{ V}$  i Arduino. Uträkningarna nedan påvisade att avvikelserna höll sig ungefär inom felmarginalens gränser.

$$\text{Avvikelse} = 2,6 \text{ V} - 2,542 \text{ V} = 0,058 \text{ V} \quad (35)$$

$$\text{Felmarginal} = \frac{\text{Avvikelse}}{2,6} \cdot 100 \% = 2,231 \% \quad (36)$$

Syftet med testet var att få förståelse över hur INA169 fungerade praktiskt och undersöka hur den kunde tillämpas för strömmätning och eventuellt spänningsmätning med motorer. En prioritet i projektet var att först kunna mäta likström eftersom en likströmgenerator användes i mätriggen.  $V_{IN+}$  var kopplad till  $+5 \text{ V}$  och  $V_{IN-}$  till jord vilket gjorde att en spänningspotentialskillnad uppstod. Detta medförde att sensorn enbart kunde mäta likström. I testet var det inte nödvändigt att förändra sensorns inbyggda resistansvärden eftersom syftet med testet var att påvisa sensorns generella funktion.

## 5.2 Mätning av moment

Mättester av momentgivare visas under detta avsnitt.

### 5.2.1 Tryckmätning med tryckkänsligt motstånd

För att kunna mäta en motors moment kan ett tryckkänsligt motstånd användas tillsammans med en Arduino. Det här testet undersökte hur väl komponenten kunde tillämpas praktiskt, för att på så sätt påvisa dess lämplighet som momentgivare av elektriska maskiner.

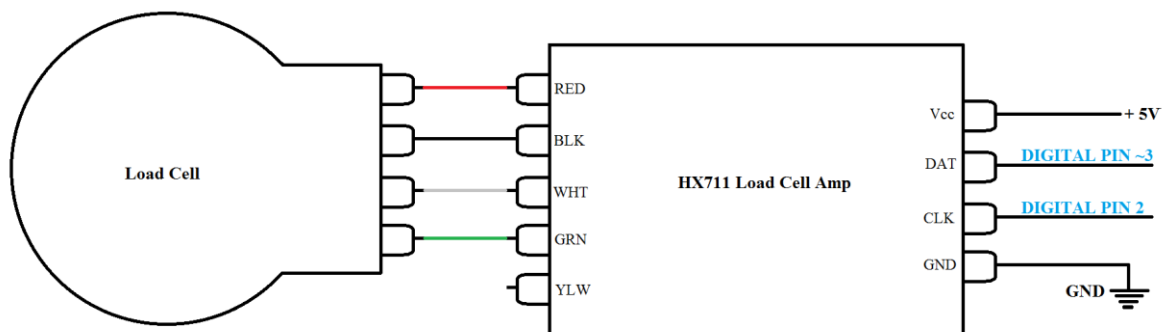
Motståndet fungerar i princip som en potentiometer genom dess resistans är varierbar, då dess motstånd minskar vid tryck. Om inget tryck påförs sensorn har den en resistans på 1 M $\Omega$ . För att snabbt kunna verifiera trycksensorns resistans tillkopplades en multimeter mot sensorn. Hur kopplingen återskapades och dess programkod fungerade i detta test kan studeras mer i bilaga B - *Tryckmätning med tryckkänsligt motstånd* [50].

Den olinjära karakteristiken och tryckytans betydelse gjorde att denna typ av komponent inte lämpade sig för att mäta momentet hos en elektrisk maskin.

### 5.2.2 Tryckmätning med lastcell TAS606

För att mäta vikt går det att använda sig av en lastcell. Nedan beskrivs ett test för att undersöka huruvida det är möjligt att mäta vikten av ett tryck med en Arduino.

Då en lastcell gav ytterst små signalförändringar vid olika tryck krävdes användning av en förstärkare. Förstärkaren som användes är en så kallad "HX711 Sparkfun Load Cell Amplifier". Denna förstärkare kopplades till viktsensorn samt en Arduino, se figur 5.4 nedan.



Figur 5.4. Kopplingschema för tryckmätning med lastcell och lastcellsförstärkare.

För att förstärkaren skulle fungera korrekt med Arduinon krävdes att ett bibliotek, kallat HX711.h installerades. Därefter kunde koden i bilaga A - *Load cell calibration* användas. Med den koden kunde sensorn både nollställas och kalibreras genom att en kalibreringsfaktor ställdes in. Detta genomfördes genom att ett finger pressades ner mot sensorn och vikten av trycket kunde läsas av på en våg som sensorn låg på, samtidigt som mätvärden presenterades i den seriella monitorn på Arduinon. När värdena på vågen överensstämde med mätvärdena som presenterades på Arduinon kunde kalibreringsfaktorn noteras.

Därefter användes en annan kod, se bilaga A - *Load cell measurement* [51].

Kalibreringsfaktorn från föregående test var sedan tidigare inställt och därför kunde det nya programmet direkt användas för att mäta vikten på sensorn med en felmarginal på  $\pm 100\text{g}$ . Högre noggrannhet krävdes ej.

Testet gav en ungefärlig uppfattning om sensorns funktionsduglighet samt genomfördes för att kalibrera sensorn för framtida användning.

### **5.3 Varvtalsmätning**

Varvtalet hos en motor kan uppmätas. Följande avsnitt beskriver två lösningsförslag på hur det kan göras.

#### **5.3.1 Varvtalsmätning med reflexdetektor breakout QRE1113**

En monterad skiva, som var fastsatt på en motors axel, hade en vit yta med ett mindre svart område. Reflexdetektorn QRE1113 skickade ut ljus mot den roterande skivan och ljusreflexion uppstod på de ljusare partierna. Den mindre mängd ljus som studsade från den mörka delen av skivan kunde, tillsammans med en Arduino, utnyttjas för att uppskatta varvtalet hos en motor. Mer information om hur komponenten kopplades upp och hur Arduinokoden användes, se bilaga B - *Varvtalsmätning med reflexdetektor breakout QRE1113* [52].

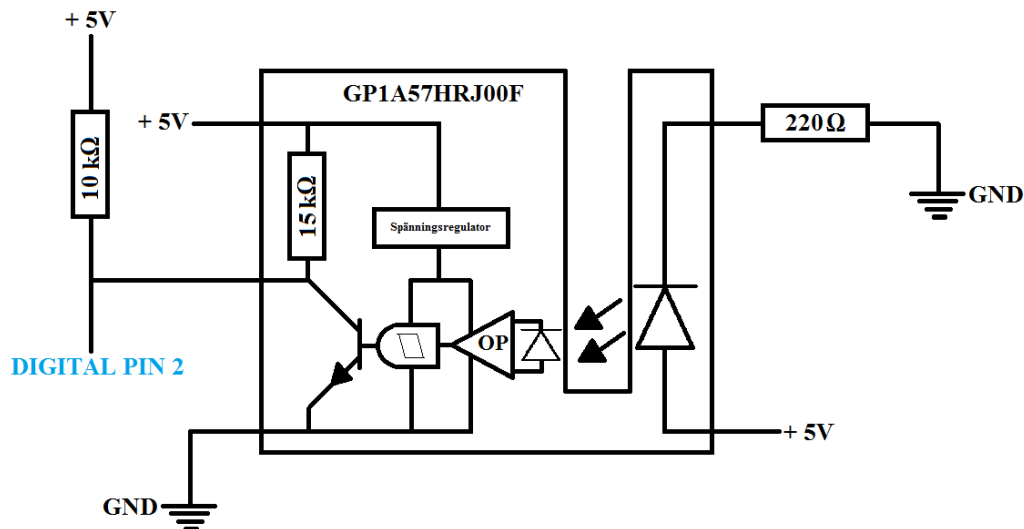
Resultaten från testet påvisade att givaren var känslig för den skugga som uppstod från motorn när den närmade sig sensorn. Följaktligen uppfattade modulen ytan som mörkare än vad den egentligen var, vilket försvårade uppräknningen varvtalet. Dessutom var avståndet tvunget att vara korrekt för att ljuset skulle kunna detekteras med en hög noggrannhet. Detta avstånd var svårt att ställa in. Dessa orsaker gjorde att denna testkonstruktion inte var en optimal lösning för mätning av en motors varvtal.

#### **5.3.2 Varvtalsmätning med ljusbrytare - GP1A57HRJ00F**

För att mäta varvtalet hos en motor går det att använda en ljusbrytare, vilket är i princip en ljusemitter samt ljusdetektor som samverkar.

På vardera sidan om komponenten sitter en lysdiod respektive fototransistor. Emittern skickar en ljusstråle mot detektorn. När strålen bryts av exempelvis ett föremål detekterar sensorn att en rörelse har skett.

Testet undersökte hur väl en ljusdetektor kunde implementeras i designen av en varvtalsräknare hos en motor. För att genomföra testet kopplades konstruktionen upp enligt figur 5.5.



Figur 5.5. Kretsschema för varvtalsövervakning med Arduino och ljusbrytare.

Figuren redovisar att ett så kallat ”Pull Down” motstånd användes. Ett sådant motstånd konstruerades via en resistor kopplad mellan utsignalen av en komponent och jord. Detta var för att säkerställa att en logisk låg signal låg så nära 0 V som möjligt [53].

En skiva, med ett litet hål i, monterades på änden av motorn. Syftet med detta var att sensorn skulle upptäcka varje gång hålet i skivan passerades, vilket i sin tur ledde till att en varvtalsgivare kunde implementeras. Skivan som användes för testet tejpades med svart tejp eftersom ljustrålen ej blockerades tillräckligt om ingen tejp användes. Skivan kunde modifieras ytterligare då den målades med svart nagellack för att säkerställa att ljuset blockerades effektivt. Möjligheten fanns att utöka antalet hål i skivan för att öka noggrannheten hos varvtalsgivaren. För detta test användes dock endast ett hål då noggrannheten inte var i fokus.

För att implementera varvtalsgivaren testades två olika koder. Första koden, se bilaga A - *Photo interrupter ver.1*, medförde att sensorn kunde detektera varje gång ett föremål fördes in mellan de två ”benen” av komponenten. Dock var denna typ av kod otillräcklig då den fortsatte räkna upp om ett föremål hölls stilla emellan komponentens delar. För att kunna implementera en bättre varvtalsräknare behövdes därför en typ av trigger som reagerade varje gång signalen gick från hög till låg. Detta ledde till en modifiering av koden enligt bilaga A - *Photo interrupter ver.2*, där en så kallad Interrupt Service Routine, ISR, användes. Denna typ av funktion var väldigt användbar då en signal skulle övervakas utan att övervakningen skulle förhindra andra funktioner i programmet [54].

Varvtalsgivaren verifierades genom att varvtalet som uppmättes med Arduinon skrevs ut i dess seriella monitor samtidigt som en tachometer riktades mot motorskivan. Tachometerens pålitlighet antogs vara god och påvisade på så sätt den undersökta varvtalsgivarens funktionsduglighet. Testet visade att denna konstruktion fungerade väl vid varvtalsövervakning hos roterande maskiner.

## 5.4 Motorstyrning

Styrning av en motor kan göras på flera olika sätt. Testerna nedanför presenterar flera olika lösningsalternativ på motorstyrning.

### 5.4.1 Motorstyrning med Arduino och transistor

En testkoppling konstruerades för att undersöka om en likströmsmotor kunde sättas på och av med en strömbrytare som styrde en MOSFET-transistor av typen IRF520. I kapitel 4.1 *Transistor - IRF520* beskrivs mer om dess uppbyggnad. Kopplingen och programkoden till testet kan undersökas noggrannare i bilaga B - *Motorstyrning med Arduino och transistor* [32].

Testets avsikt var att pröva om kopplingen kunde, tillsammans med en programmerad Arduino, styra en DC-motor. Resultatet visade att det fungerade eftersom motorn kunde sättas på eller stängas av genom strömbrytaren.

Nackdelen med lösningen var att batteriet inte genererade en konstant spänning till motorn. En närmare undersökning av batteriet på 9 V visade att motorn snurrade långsammare i takt med att batteriet laddades ur vid motordrift. Därför ersattes batteriet med en spänningsgenerator som gav ut en konstant spänning för att kunna bibehålla en jämn hastighet hos motorn.

### 5.4.2 Test av motorstyrning med Arduino och H-brygga

I detta test önskades en funktion som drev en likströmsmotor framåt och bakåt med varierande hastighet. För att få en motor till att vrida sig i en riktning och sedan vrida sig i den motsatta krävdes det att spänningens polaritet till motorn skiftades. Dessutom krävdes det att spänningens styrka kunde förändras för att styra rotorns vridhastighet. Genom att en DC-motor kopplades upp en med en H-brygga av modellen L293DNE mot en Arduinoenhet kunde motorn kontrolleras enligt ovan. Hur denna koppling såg ut kan läsas mer under bilaga B - *Test av motorstyrning med Arduino och H-brygga* [32]. Mer om H-bryggans funktion beskrivs i 4.2 *H-brygga - L293DNE*.

Lösningen var dock inte helt optimerad eftersom de komponenter som utgjorde styrningen upptog en del av utrymmet i kretsen. Dessutom skapades det onödiga värmeförluster hos resistorerna.

För att reducera antalet komponenter i lösningen kunde brytarnas och potentiometerns funktioner flyttas från kretsen in till programmet LabVIEW. Om motsvarande funktion av styrningen skulle kunna erhållas krävdes det först att en kommunikation mellan LabVIEW och Arduino-kortet etablerades, se bilaga E för lathund till installationsguide [55]. Den förminskade kretsen och LabVIEWKoden går att finna i bilaga B - *Test av motorstyrning med Arduino och H-brygga* som det nämndes tidigare [56].

Syftet med testet var att utreda huruvida komponenten L293DNE kunde utnyttjas för att få

kontroll över en DC-motor. Med en mindre koppling, konfiguration av kommunikationen och LabVIEW-kod gav testet en multifunktionell styrenhet. Där styrenheten kunde utföra kommunikationsinställningar, justera motorns hastighet och riktning, och övervaka motorn i en användarvänlig programmiljö.

### **5.4.3 Motorstyrning med motor shield**

För att testa funktionen hos en Arduino Motor Shield kopplades en display in till en Arduino. Displayen som användes är en LCD av typen HD44780. I bilaga B - *Motorstyrning med motor shield* kunde testkopplingen och dess programmeringskod studeras utförligt.

Syftet med det här testet var att undersöka hur väl en motor shield kunde styra en motor samt se om värden skickades ut från motor shielden till en display. Testet påvisade att detta var fullt möjligt och både motor shield och display fungerade på tillfredsställande sätt.

## 6. SLUTGILTIG KONSTRUKTION

Nedan beskrivs den slutgiltiga mätriggens uppbyggnad samt test som genomfördes för att verifiera mätriggens funktion.

### 6.1 Uppbyggnad

Mätriggen konstruerades genom att ena maskinen kunde rotera kring sin axel medan andra maskinen var fastsatt i en platta. Eftersom den ena maskinen var roterbar kunde momentet som verkade på maskinen uppmätas. Den fastsatta maskinens position kunde justeras med ett flertal skruvar vilket möjliggjorde centrering av maskinernas axlar oavsett utseende på maskinen.

Motorns och generatorns axlar sammankopplades till en gemensam axel genom en plastslang. Plastslangen gav en ”mjukare” övergång när motorn gick från tomgång till belastning och minskade risken för eventuell sprickbildning i konstruktionen. Centrering av axeln i mitten av kortsidan på maskinerna medförde en regelbunden rotation hos båda maskinerna.

Till maskinerna satt totalt fyra strömsensor INA169, en ljusbrytare - GP1A57HRJ00F, en lastcell - TAS606 och en lastcells förstärkare - HX711 anslutna. Lastcellen, lastcells förstärkaren och ljusbrytaren monterades på en gemensam breadboard medan de fyra strömsensorerna satt kopplade på en egen. Varje maskin övervakades av två av strömsensorerna, där ena strömsensorn övervakade ström och spänning hos statorn och den andra övervakade ström och spänning hos rotorn. Se bilaga F för kopplingschema. Ljusbrytaren placerades mellan motorernas gemensamma axel och lastcellen fastsattes under en metallskiva som förflyttades genom generatorns moment. Dessa kopplades på samma vis som enligt figur 5.4 samt figur 5.5.

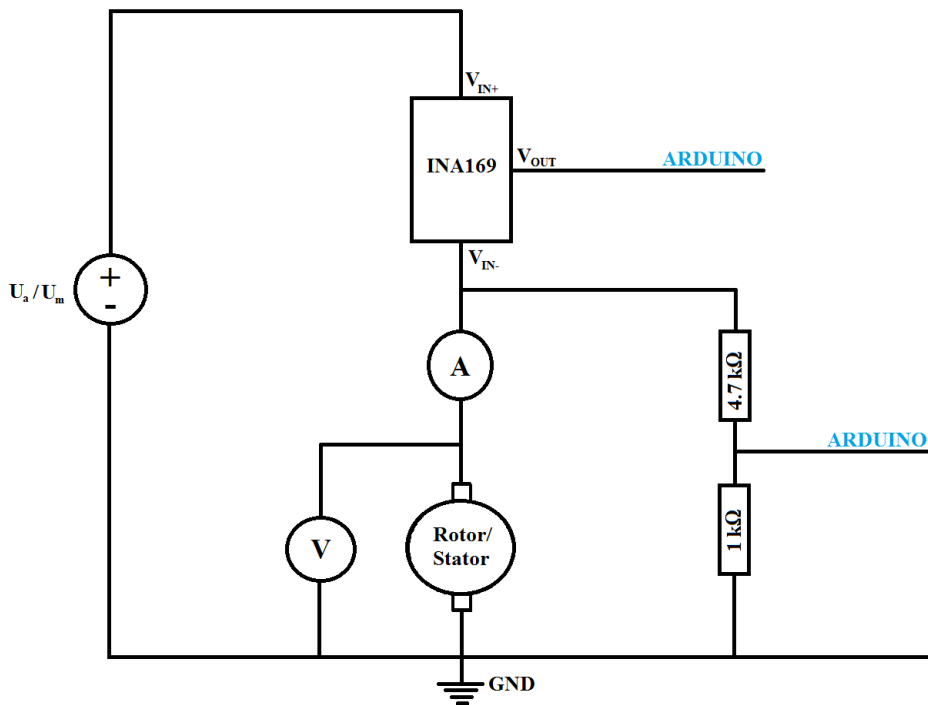
Strömsensorerna kopplades till en dator via en Arduino Mega. Lastcellen och ljusbrytaren kopplades till samma dator via en Arduino Uno. Strömmarna och spänningarna övervakades i en GUI (Graphical User Interface) i MATLAB, se bilaga C för kod och bilaga D för gränssnitt [57]-[58]. För att möjliggöra kommunikationen mellan Arduinon och MATLAB krävdes det att verktygspaketet *Matlab Support Package for Arduino Hardware* installerades på datorn. Paketet laddades ner från MathWorks hemsida.

Varvtalet samt momentet övervakades i Arduino Serial Plotter, se bilaga A - *Torque and rpm measurement* för kod.

### 6.2 Verifiering

För att undersöka sensorernas funktionsduglighet som ström- och spänningsövervakare till elmaskiner kopplades de enligt figur 6.1. Två multimetrar samt spänningsaggregatets inbyggda ström- och spänningsindikator användes för verifieringen.





Figur 6.1. Inkoppling av ström- och spänningssensorer till rotor respektive stator i elmaskiner.

När ena maskinen kördes under motordrift i tomgång erhöles resultat enligt tabell 6.1. Vilken maskin som kördes var oväsentligt eftersom resultaten från maskinerna var nästan likadana.

Tabell 6.1. Mätvärden på strömmar och spänningar från olika mätutrustning under motordrift i tomgång.

	$U_m$ [V]	$I_m$ [A]	$U_a$ [V]	$I_a$ [A]
INA169	3,52	0,43	4,86	0,41
Aggregat	3,50	0,44	4,90	0,41
Multimeter	3,42	0,44	4,76	0,41

Med dessa indikatorer erhöles mätvärden med en avvikelse på maximalt 2,9 %.

Varvtalsgivaren verifierades på samma vis som under 5.3.1 Varvtalsmätning med ljusbrytare - GP1A57HRJ00F, alltså med användning av en tachometer. Mätvärden från tachometern respektive mätrigg presenteras i tabell 6.2.

Tabell 6.2. Mätvärden på varvtal från olika mätutrustning under körning av mätrigg.

Mätrigg [rpm]	2280	2160	1800	1680	1260	660
Tachometer [rpm]	2295	2181	1800	1658	1268	660

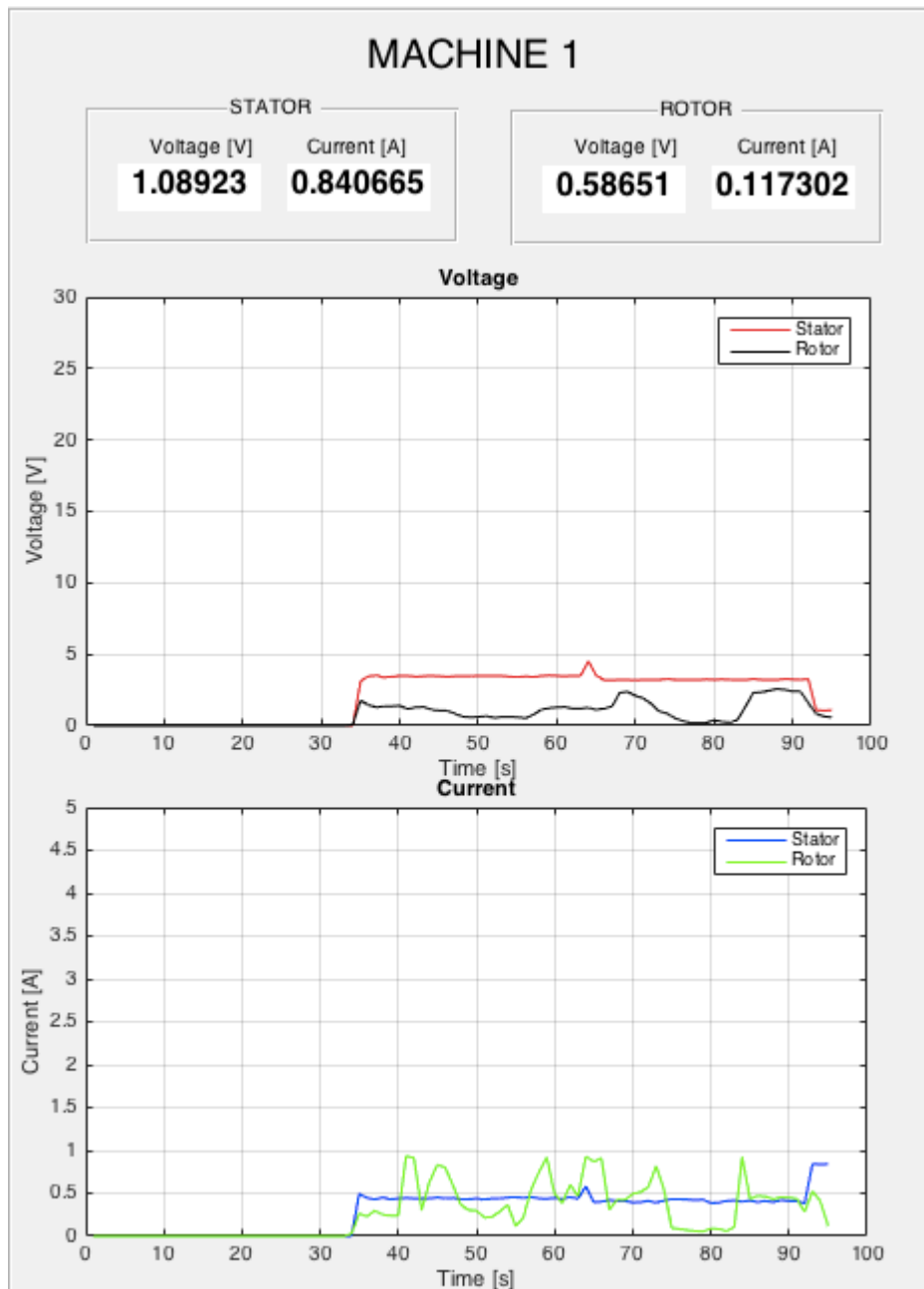
Jämförelse av värden från tachometer och mätrigg gav en maximal avvikelse på 1,3 %.

Momentgivare verifierades och kalibrerades under 5.2.2 *Tryckmätning med lastcell TAS606* och behövde därför inte verifieras på nytt.

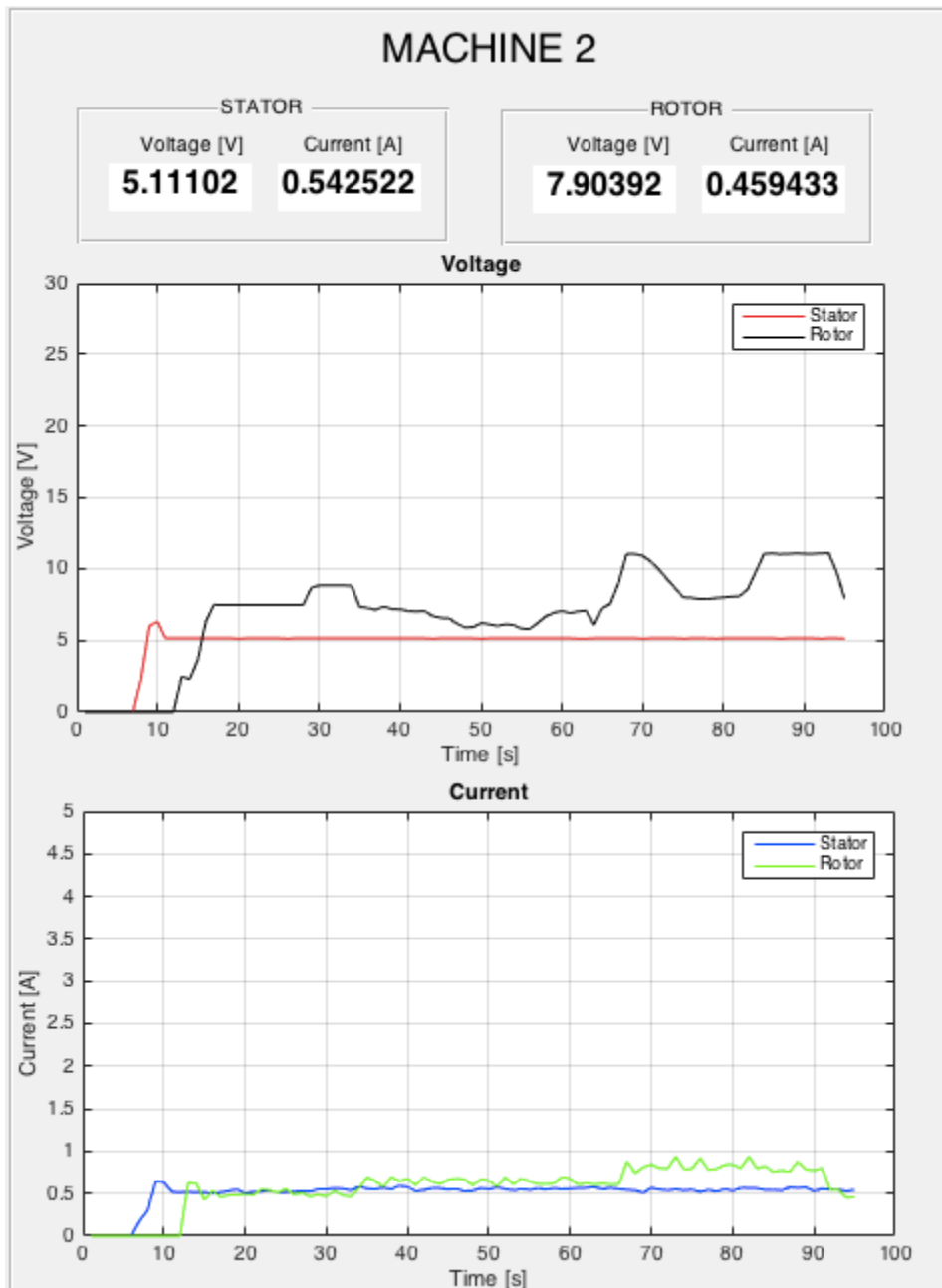
Därefter kördes ena maskinen i motordrift och den andra i generatordrift. Spänningar och strömmar övervakades precis som i föregående undersökning men därutöver kopplades även varvtals- och momentgivare in, se bilaga F för koppling. En testkörning av konstruktionen startades. Till en början befann sig motorn i tomgång. Därefter påbörjades en spänningsmatning i generatorns stator, vilket gav upphov till en magnetisering. Denna magnetisering resulterade i att varvtalet sjönk, momentet ökade och motorns ankarström ökade. När lasten till generatorn tilltog gav det upphov till ytterligare en varvtalssänkning och momentökning. Därutöver ökade motorns ankarström och generatorns ankarström minskade.

## 7 RESULTAT

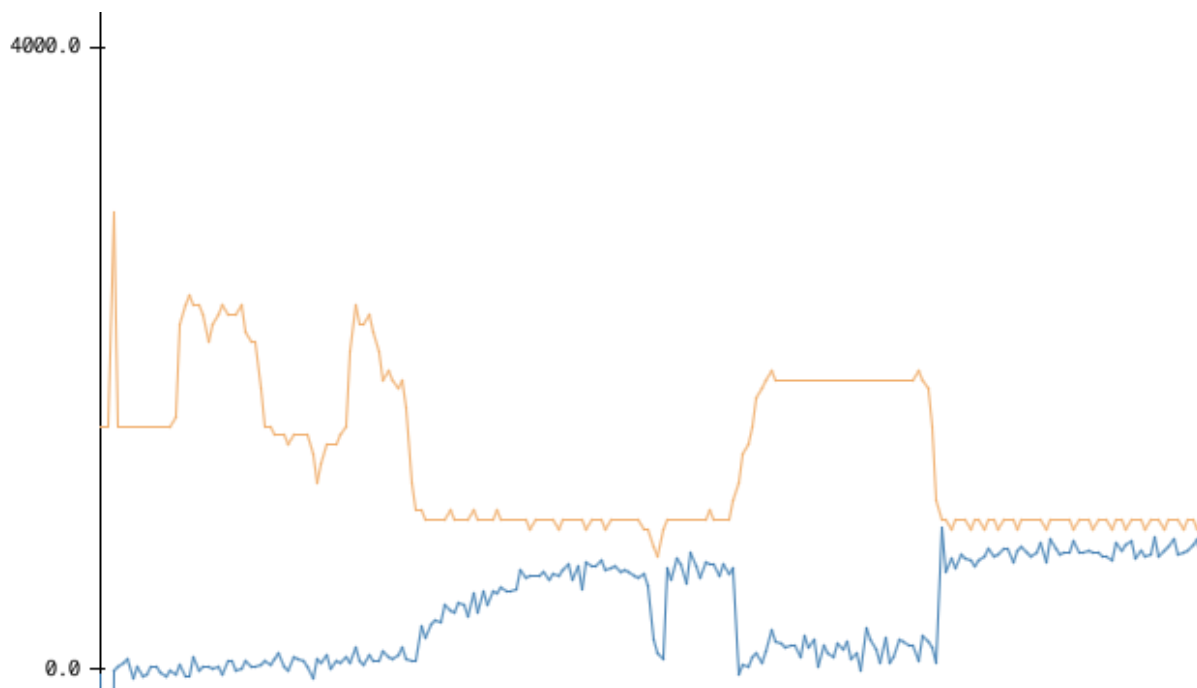
Mättriggen som har konstruerats kan övervaka likströmmar, likspänningar, moment och varvtal i realtid hos elektriska maskiner. I figur 7.1-7.3 presenteras hur denna övervakning ser ut.



Figur 7.1. Ena halvan av MATLABS gränssnitt när mättriggen körs. Strömmar och spänningar hos generatorns rotor respektive stator presenteras.



*Figur 7.2. Andra halvan av MATLABS gränssnitt när mätriggen körs. Strömmar och spänningar hos motorns rotor respektive stator presenteras.*



*Figur 7.3. Mätvärden på varvtal respektive generatorns moment hos mätriggen. Värdena presenteras i Arduino serial plotter, där gul kurva är varvtal och blå kurva är moment.*

Strömmar och spänningar hos både rotor och stator kan ritas ut i fyra olika fönster i MATLAB. Numeriska värden på dessa storheter presenteras även högst upp i MATLAB-gränssnittet. Moment och varvtal övervakas i Arduino där serial plotter kan startas för att ge grafer uppritade. Numeriska värden på moment och varvtal kan presenteras om Arduino serial monitor öppnas. Dock kan inte serial monitor köras samtidigt som serial plotter.

För en ordentlig överblick över hur programmen ser ut i sin helhet, se bilaga G.

## 8 DISKUSSION

En kravspecifikation framställdes aldrig då Chalmers inte hade några mer krav än att storheter hos elektriska maskiner skulle kunna uppmätas. En prototyp skulle kunna konstrueras i framtiden där en tillhörande kravspecifikation uppställs. Förslag på krav som kan ställas där är då exempelvis att säkerheten skall vara förbättrad genom inkapsling av elektriska och roterande delar.

Valet av INA169 grundades i att den kunde med god noggrannhet uppmäta både ström och spänning i en krets. ACS712 har fördelen att den fungerar för både växel- och likström men eftersom den hade problem med instabilitet gjordes en avvägning och det ansågs viktigare att korrekt kunna uppmäta strömmar oavsett om endast likströmmar kunde avläsas. Mätning av moment genomfördes med lastcell tillsammans med dess förstärkare eftersom det låg svårigheter i att kalibrera tryckmätningen med tryckmotståndet då ytan på trycket var svårt att hålla konstant. Detta problem upplevdes aldrig med lastcellen då dess knappliknande del underlättade kontroll av tryckyta. Varvtalet uppmättes med en ljusbrytare då problem kring störningsreflexioner och skuggor aldrig upplevdes hos denna komponent. En reflexdetektor visade sig vara alldeles för känslig för sin omgivning medan en ljusbrytare endast påverkas av hur bra ljussignalen mellan dess stolpar blockeras.

Avvikelsen på ström- och spänningssensorerna samt varvtalet och momentgivaren är väldigt låg och kommer inte påverka mätriggens funktionsduglighet. Studenter skall kunna få en generell uppskattning av sambanden hos elektriska maskiner, vilket innebär att mätningen ej behöver vara helt exakt.

När en elektrisk maskin går från att ha körts i tomgång till att en belastning läggs på kommer varvtalet sjunka om matningen till maskinen hålls konstant. Därutöver tilltar moment och ankarström i motorn. Detta eftersom en högre last kräver att motorn svarar med en högre ström för att kunna orka driva sin belastning. Om sedan belastningen till generatoren ökas kommer motorn återigen uppleva det som att dess belastning ökar. Det innebär att varvtalet sjunker ytterligare och momentet ökar. Därmed kommer generatorns ankarström att vara lägre eftersom en generator omvandlar mekanisk effekt till elektrisk. Det tidigare beskrivna beteende är likadant som det beteende som studerades vid driften av motorerna i mätriggen. Alltså är mätriggen konstruerad på tillfredsställande vis.

Eftersom ström- och spänningmätningen sker med en komponent av typen INA169 kan enbart likströmsmotorer övervakas i MATLAB. Denna begränsning av mätning kan överkommas om en sensor av typen ACS712 används. Principen för övervakning av växelströmsmaskiner är ungefär densamma men det finns ändå vissa skillnader som bör ha i åtanke. Däremot är det viktigt att komponenten är ytterst noggrant kalibrerad och därefter aldrig förflyttas eller rörs på något vis. Detta kan åstadkommas om hela mätriggens kopplingar integreras inuti någon typ av skyddslåda. Detta skulle även höja säkerheten hos hela konstruktionen vid framtida bruk.

Oavsett vilket teknikområde som ska behandlas är MATLAB ett lämpligt verktyg att använda. Synnerligen i detta arbete, där flera områden berörs, är det viktigt att det finns en flexibel hantering av informationsdata. Med en utvecklingsmiljö som är anpassningsbar för flera områden, användarvänlig, har hög tillgänglighet och en stark support lämpar sig MATLAB väl för detta arbete.

I konstruktionen används två stycken Arduino därför att vid övervakning av varvtal hos motorn krävs det att mätningen utförs under en viss tid. Detta leder till att programmet måste stanna upp för att kunna genomföra mätningen. Om programmet stannar upp ritas inte graferna för de andra mätvärdena ut i realtid. Denna fördröjning hos programmet skulle kunna avlägsnas om övervakningen av varvtal sker parallellt med övervakningen av de andra storheterna hos maskinen, vilket leder till att enbart en Arduino skulle krävas.

Moment samt varvtal övervakas inuti Arduino serial plotter, då det är en mindre komplex lösning än övervakning via MATLAB. Det är möjligt att förflytta momentövervakningen till MATLAB men eftersom det ändå krävdes en separat Arduino för varvtalsövervakningen implementerades de båda övervakningarna på Arduino.

Valet av Arduino grundas i att programmeringsspråket C ingår i utbildningen för Elektro-, Data- och Mekatronikingenjörer vid Chalmers tekniska högskola och är även ett språk som används globalt. Att Arduino är lätt att använda oavsett tidigare erfarenheter kring programmering spelar stor roll. Det ska vara lätt för studenter att, oavsett tidigare programmeringskunskaper, kunna sätta sig in i hur testtriggen fungerar. Detta för att studenterna skall kunna erhålla mer kunskap under laborationstillfället.

Arduinos öppna hårdvara och källkod gör att priset på produkten pressats ner. Det går även att tillverka egna Arduinoenheter, vilket betyder att priset kan minimeras till endast materialkostnader. Arduinos låga pris leder till att det, om så önskas, kan sättas upp flera stationer med mätriggen i laborationssalarna. Den öppna källkoden gör också att det finns en stor mängd information kring andra liknande projekt, vilket underlättar utvecklingsprocessen. Det spelar heller ingen roll vilket operativsystem som finns på datorerna i laborationssalarna eftersom Arduino har stöd för Windows, Mac och Linux. Av de ovan nämnda anledningarna lämpar sig Arduino väl till det här projektet.

Testen påvisade att motorstyrningen kunde genomföras på flera olika sätt. Att styra en motor med hjälp av en motor shield har fördelen att det är lätt att implementera. Däremot upptas ett flertal portar på en Arduino så snart en Arduino Shield används. Styrning med transistor har begränsningen att motorn endast kan slås på och av. Däremot med en h-brygga kunde flera funktioner implementeras till styrningen. Denna typ av styrning har dock nackdelen att en stor koppling krävs. Om LabVIEW integreras med h-bryggan minskar antal komponenter som krävs och styrningsmöjligheterna behålls. Denna typ av styrning med LabVIEW kräver dock en stor mängd invecklad kod i Arduino. Styrningen kunde även övervakas i LabVIEW. Eftersom denna typ av styrning och övervakning är så pass komplicerad gjordes en avvägning att styrningen av maskinerna enbart sker via spänningsaggregat. Att implementera styrning

via LabVIEW kan vara ett tänkbart framtida examensarbete.

För detta projekt föll valet på LabVIEW av flera anledningar. För det första är det ett användbart verktyg som lämpar sig för utbildningssyften. Studenter kan på ett visuellt sätt effektivisera sitt arbete, vilket är en fördel för ett projektarbete där tiden är begränsad. Samtidigt är programmet lämpligt för studenter inom elektroteknik eftersom den grafiska kodningen är lik elektronikkretsar där komponenter eller blockdiagram dras in och sätts samman med trådar. Vidare är programmet utvecklat främst för att samla in mätdata från exempelvis olika typer av givare. Mängden textad programmering som förekommer resulterar i att valet av program passar med fler av studenternas utbildningar.

LabVIEW är väletablerat, vilket betyder att andra universitet kan ta del av detta projektarbete. Dessutom kan programmet tillämpas i industrin om det önskas. Då programmet är tämligen dyrt kan det ses som en nackdel för användning av programmet. Dock har Chalmers redan investerat i programmet, vilket innebär att det inte leder till något problem för det här projektet.

Hade samma arbete påbörjats idag med de kunskaper som erhållits under arbetets gång hade vi beställt in fler olika typer av ström- och spänningssensorer. Detta hade kunnat leda till att mer stabila sensorer hade påträffats som kan övervaka växelspanningar, vilket i sin tur hade lett till att övervakning av växelströmsmaskiner hade kunnat implementeras. Mer tid hade kunnat läggas på att göra mätriggen säker genom att isolera kretsar och då höja säkerheten hos mätriggen. En begränsning av spänningsmatningen hade även kunnat implementeras vilket hade lett till att maskinerna aldrig kan matas med högre spänningar än vad de klarar av. Detta hade varit särskilt användbart i utbildningssyfte då ej erfarna personer lätt hade kunnat mata riggen med högre spänningar än den klarar av.



## 9 SLUTSATS

Valet av de olika sensorerna påvisade att de kunde återge tillräckligt goda mätvärden då de undersöktes individuellt. Tester på den slutgiltiga riggen visar att spänning, ström, varvtal och moment kan övervakas med rimlig noggrannhet. Utöver detta har alla sensorer implementerats så att de kan nyttjas samtidigt, vilket underlättar undersökning av maskinernas karakteristik.

Mätvärdena finns tillgängliga digitalt och presenteras i realtid i ett grafiskt gränssnitt på datorer. Då gränssnittet hos den slutgiltiga riggen har en simplistisk design är den enkel att nyttja och lämpar sig därmed för undervisning.

Eftersom ett spänningsaggregat används istället för direktinkoppling i vägguttag minimeras riskerna för elskador. Dessutom är aggregatet säkrat. Mättriggen bör ej användas utan sällskap av laborationspartner därför att om en olycka sker kan den skadade snabbt få hjälp. Detta kan minska tiden som kroppen utsätts för ström.

Maskinerna i mättriggen kan maximalt matas med en spänning på 24 V. Däremot krävs det inte att de körs i sin maximala drift för att studenterna skall kunna erhålla kunskaper om maskinerna. Eftersom mättriggen körs vid väldigt mycket lägre effekter än den gamla utrustningen kommer laborationernas miljöpåverkan att vara betydligt lägre än tidigare. Att fler mättriggas kommer köras under laborationstillfället spelar ingen roll eftersom den totala effektenkonsumtionen under laborationstillfället uppskattningsvis kommer vara lägre än vad den för nuvarande är.

Undersökningen av olika varianter av motorstyrning påvisade att det är fullt möjligt att konstruera en mättrigg där motorn styrs av annat än ett spänningsaggregat.

Sammanlänkas det ovan beskrivna beteendet hos en mättrigg erhålls en konstruktion som kan digitalt registrera och presentera mätdata hos elektriska maskiner, vilket innebär att målen samt syftet med projektet uppfylls.

## REFERENSER

- [1] Bergström, L. och Nordlund, L. (2012) *Ellära – Krets- och fältteori*. Tredje upplagan. Stockholm, Sverige: Liber AB, ISBN 978-91-47-10619-6.
- [2] Baigrie, B. (2007) *Electricity and Magnetism: A Historical Perspective*. Westport, Connecticut, London, Storbritannien: Greenwood Press, ISBN-13: 978-0313333583.
- [3] Nilsson, J. S. och Svensson, B. E. Y. (2016) Växlerverkan. *NE*.  
<http://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/v%C3%A4xlerverkan>  
(22 mars 2016).
- [4] Engström, L. A. (2000) *Elektromagnetism – från bärnsten till fältteori*. Lund, Sverige: Studentlitteratur, ISBN 91-44-01510-0.
- [5] *Världens historia - Vem uppfann elmotorn?* (2009)  
<http://varldenshistoria.se/teknik/maskiner/vem-uppfann-elmotorn> (23 mars 2016).
- [6] Doppelbauer, M. (2014) The invention of the electric motor 1800-1854. *Kit.edu*.  
<https://www.eti.kit.edu/english/1376.php> (23 mars 2016).
- [7] Hallenius, K.-E. (1972) *Elektriska maskiner*. Upplaga 2:1. Stockholm, Sverige: Liber AB, ISBN 91-23-13175-6.
- [8] Alfredsson, A. (2011) *Elkraft*. Fjärde upplagan. Stockholm, Sverige: Liber AB, ISBN-978-91-47-10057-6.
- [9] Lindahl, P.-E. (1983) *Elkraftteknik*. Andra upplagan. Lund, Sverige: Studentlitteratur, ISBN 91-44-02172-0.
- [10] Chapman, S. J. (2002) *Electric Machinery and Power Systems Fundamentals*. New York, USA: McGraw-Hill, ISBN 978-0-07-229135-3.
- [11] Alfredsson, A et al. (1996) *Elkraft Handboken Elmaskiner*. Berlings, Arlöv, Sverige: Liber AB, ISBN 91-47-00066-X.
- [12] Boldea, I. och Nasar, S.A. (2006) *Electric drives*. Andra upplagan. New York, USA: CRC Press, ISBN 9780849342202.
- [13] Institutionen för Energi och miljö (2014) *Elteknik*. Chalmers tekniska högskola, Göteborg, Sverige: Teknologtryck.
- [14] Bengtsson, L. (2012) *Elektriska mätsystem och mätmetoder*. Upplaga 1:1. Lund, Sverige: Studentlitteratur, ISBN 978-91-44-08068-0.
- [15] Al-Mutlaq, S. (2015) Getting Started with Load Cells. *Sparkfun*.  
<https://learn.sparkfun.com/tutorials/getting-started-with-load-cells> (13 maj 2016).
- [16] *Phidgets - Load Cell Primer* (2012) [http://www.phidgets.com/docs/Load\\_Cell\\_Primer](http://www.phidgets.com/docs/Load_Cell_Primer)  
(13 maj 2016).

- [17] Nordlund, L. och Wiklund, I. (2012) *Grundläggande elektronik*. Andra upplagan. Stockholm, Sverige: Liber AB, ISBN 978-91-47-10532-8.
- [18] Tefy - Personskador av elström [PDF] (2016)  
[http://www.tefy.se/personskador\\_av\\_elstrom.pdf](http://www.tefy.se/personskador_av_elstrom.pdf) (24 maj 2016).
- [19] *Arduino.cc* - Arduino Products (2016) <https://www.arduino.cc/en/Main/Products> (18 april 2016).
- [20] Seidle, N. (2013) Analog to Digital Conversion. *Sparkfun*.  
<https://learn.sparkfun.com/tutorials/analog-to-digital-conversion> (26 april 2016).
- [21] *Arduino.cc* - Read Analog Voltage (2015)  
<https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage> (26 april 2016).
- [22] *Charterhouse Solutions* - LabVIEW (2006).  
[http://www.charterhousesolutions.co.uk/labview\\_more.htm](http://www.charterhousesolutions.co.uk/labview_more.htm) (21 april 2016).
- [23] *National Instruments* - LabVIEW Environment Basics (2016)  
<http://www.ni.com/getting-started/labview-basics/environment> (21 april 2016).
- [24] *National Instruments* - What's New in LabVIEW 2015 (2016)  
<http://www.ni.com/labview/whatsnew/> (21 april 2016).
- [25] *National Instruments* - LabVIEW 2015 Features and Changes (2015)  
[http://zone.ni.com/reference/en-XX/help/371361M-01/lvupgrade/labview\\_features/](http://zone.ni.com/reference/en-XX/help/371361M-01/lvupgrade/labview_features/) (23 april 2016).
- [26] *National Instruments* - Benefits of Programming Graphically in NI LabVIEW (2013)  
<http://www.ni.com/white-paper/14556/en/>, (23 april 2016).
- [27] *National Instruments* - LabVIEW or C? (Could You Rephrase the Question?) (2013)  
<http://www.ni.com/newsletter/51675/en/> (23 april 2016)
- [28] *National Instruments* - Advantages of Using LabVIEW in Academic Research (2013)  
<http://www.ni.com/white-paper/8534/en/> (23 april 2016).
- [29] Moler, C. (2004) The Origins of MATLAB. *Mathworks*.  
<http://se.mathworks.com/company/newsletters/articles/the-origins-of-matlab.html> (15 april 2016).
- [30] *Mathworks* - The Language of Technical Computing (2016)  
<http://se.mathworks.com/products/matlab/?requestedDomain=www.mathworks.com> (15 april 2016).
- [31] Jönsson, P. (2010) *MATLAB beräkningar inom teknik och naturvetenskap*, Upplaga 3:6. Lund, Sverige: Studentlitteratur, ISBN 978-91-44-06926-5

- [32] Fitzgerald, S. och Shiloh, M. (2012) *Arduino Projects Book*. Andra upplagan. Turin, Italien: Arduino LLC.
- [33] *Alldatasheet* - IRF520 Datasheet 1 Page - STMicroelectronics [PDF] (1993) <http://html.alldatasheet.com/html-pdf/22389/STMICROELECTRONICS/IRF520/1620/1/IRF520.html> (13 maj 2016).
- [34] *Alldatasheet* - L293DNE Datasheet 1 Page - Texas Instruments [PDF] (2004) <http://html.alldatasheet.com/html-pdf/26889/TI/L293DNE/21/1/L293DNE.html> (13 maj 2016)
- [35] *Electrokit* - Tryckkänsligt motstånd 0.5" (2016) <http://www.electrokit.com/tryckkansligt-motstand-0-5.46954> (2 maj 2016).
- [36] *Sparkfun* - SparkFun Current Sensor Breakout – INA169 (2016) <https://www.sparkfun.com/products/12040> (29 april 2016).
- [37] *Reuk* - What Is A shunt (2015) <http://www.reuk.co.uk/What-is-a-Shunt.htm> (29 april 2016).
- [38] Hymel, S. (2013) INA169 Breakout Board Hookup Guide. *Sparkfun*. <https://learn.sparkfun.com/tutorials/ina169-breakout-board-hookup-guide> (29 april 2016).
- [39] *Sparkfun* – INA169\_Breakout [PDF] (2013) [http://cdn.sparkfun.com/datasheets/Sensors/Current/INA169\\_Breakout.pdf](http://cdn.sparkfun.com/datasheets/Sensors/Current/INA169_Breakout.pdf) (29 april 2016).
- [40] *Electrokit* - Low Current Sensor Breakout - ACS712 (2016) <http://www.electrokit.com/low-current-sensor-breakout-acs712.53200> (19 maj 2016).
- [41] *Electrokit* - Reflexdetektor breakout (2016) <http://www.electrokit.com/reflexdetektor-breakout.50105> (3 maj 2016).
- [42] *Sparkfun* - SparkFun Line Sensor Breakout - QRE1113 (Digital) (2016) <https://www.sparkfun.com/products/9454> (3 maj 2016).
- [43] *Lawicel-shop* - Photo Interrupter GP1A57HRJ00F (2016) [http://www.lawicel-shop.se/prod/Photo-Interrupter-GP1A57HRJ00F\\_881942/Reflektion--Ljus\\_58459/SWE/SEK](http://www.lawicel-shop.se/prod/Photo-Interrupter-GP1A57HRJ00F_881942/Reflektion--Ljus_58459/SWE/SEK) (3 maj 2016).
- [44] *Sparkfun* - HD44780U (LCD-II) (Dot Matrix Liquid Crystal Display Controller/Driver) [PDF] (1998) <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf> (24 april 2016).
- [45] *Sparkfun* - Load Cell - 50kg, Disc (TAS606) (2016) <https://www.sparkfun.com/products/13331> (11 maj 2016).
- [46] *Sparkfun* - SparkFun Load Cell Amplifier - HX711 (2016) <https://www.sparkfun.com/products/13230> (11 maj 2016).

- [47] *14core* - Introduction of ACS712 Current Sensor Module – 30A with Arduino (2015) <http://www.14core.com/introduction-of-ac712-current-sensor-module-30a-with-arduino/> (19 maj 2016).
- [48] Smith, W.A. (2013) Measuring DC Voltage using Arduino. *Startingelectronics*. <https://startingelectronics.org/articles/arduino/measuring-voltage-with-arduino/> (3 juni 2016).
- [49] *Sparkfun* - High-Side Measurement CURRENT SHUNT MONITOR [PDF] (2005) <http://cdn.sparkfun.com/datasheets/Sensors/Current/ina169.pdf> (4 maj 2016).
- [50] Jimb0 (2016) Force Sensitive Resistor Hookup Guide. *Sparkfun*. <https://www.sparkfun.com/tutorials/389> (18 april 2016).
- [51] Al-Mutlaq, S. (2015) Load Cell Amplifier HX711 Breakout Hookup Guide. *Sparkfun*. <https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide> (18 april 2016).
- [52] *Bildr.blog* - Line Sensing. QRE1113 Reflectance Sensor + Arduino (2011) <http://bildr.org/2011/06/qre1113-arduino/> (3 maj 2016).
- [53] *Utopiamechanicus* - Adding an Opto-Interruptor Part 3 (2012) <http://www.utopiamechanicus.com/article/opto-interrupter-part-3/> (8 maj 2016).
- [54] *Arduino.cc* - attachInterrupt() (2016) <https://www.arduino.cc/en/Reference/AttachInterrupt> (10 maj 2016).
- [55] AllAboutEE (2012) How To Program The Arduino With LabVIEW Tutorial [YouTube] <https://www.youtube.com/watch?v=RGRhIQneO6w> (18 april 2016).
- [56] Arduino Robotics LabVIEW Solidworks (2014) controlling a dc motor from labview using arduino [YouTube] <https://www.youtube.com/watch?v=uMkRuFQsd84> (21 april 2016).
- [57] Nedelkovski, D. (2015) Arduino and Matlab GUI Tutorial. *How to mechatronics*. <http://howtomechatronics.com/tutorials/matlab/matlab-gui/> (15 april 2016).
- [58] Zaman, S. (2015) Video Player with Pause/Play. *Mathworks*. [http://www.mathworks.com/matlabcentral/fileexchange/51014-video-player-with-pause-play/content/html/Video\\_Player.html](http://www.mathworks.com/matlabcentral/fileexchange/51014-video-player-with-pause-play/content/html/Video_Player.html) (15 april 2016).

# BILAGOR

## A – Arduinokod

```

// ----- Voltage divider -----
void setup(){
  Serial.begin(9600);
}
void loop(){
  // Read the input on analog pin 2:
  int sensorValue = analogRead(A2);
  // calculate the voltage
  // use 5.0 for a 5.0V ADC reference voltage
  // 5.015V is the calibrated reference voltage
  float voltage = 0.0; // calculated voltage
  voltage = sensorValue * (5.0 / 1023.0);
  // send voltage for display on Serial Monitor
  Serial.print(voltage * 11);
  Serial.println (" V");
  delay(2); // allow ADC to stabilize
}
// ----- End of voltage divider code -----

// ----- INA169 -----
// Constants
const int SENSOR_PIN = A0; // Input pin for measuring Vout
const int VOLTAGE_PIN = A1; //Input pin for measuring voltage over load
const float RS = 0.1; // Shunt resistor value (in ohms)
const int VOLTAGE_REF = 5; // Reference voltage for analog read
// Global Variables
float sensorValue; // Variable to store value from analog read
float current; // Calculated current value
float voltage;
void setup() {
  // Initialize serial monitor
  Serial.begin(9600);
}
void loop() {
  // Read a value from the INA169 board
  sensorValue = analogRead(SENSOR_PIN);
  voltage = analogRead(VOLTAGE_PIN);
  // Remap the ADC value into a voltage number (5V reference)
  sensorValue = (sensorValue * VOLTAGE_REF) / 1023;
  voltage = (voltage * VOLTAGE_REF) / 1023;
  // Is = (Vout x 1k) / (RS x RL)
  current = sensorValue / (10 * RS);
  // Output value (in voltage) to the serial monitor to 3 decimal
  // places
  Serial.print("Voltage: ");
  Serial.print(voltage, 3);
  Serial.print(" V");
}

```

```

// Output value (in amps) to the serial monitor to 3 decimal
// places
Serial.print("\t\t Current: ");
Serial.print(current, 3);
Serial.println(" A");
// Delay program for a few milliseconds
delay(500);
}
// ----- End of INA169 code -----

// ----- Load cell calibration -----
#include "HX711.h"
#define DOUT 3
#define CLK 4
HX711 scale(DOUT, CLK);
float calibration_factor = -68000; // change this to calibrate
void setup() {
  Serial.begin(9600);
  Serial.println("HX711 calibration sketch");
  Serial.println("Remove all weight from scale");
  Serial.println("After readings begin, place known weight on scale");
  Serial.println("Press + or a to increase calibration factor");
  Serial.println("Press - or z to decrease calibration factor");
  scale.set_scale();
  scale.tare(); //Reset the scale to 0
  long zero_factor = scale.read_average(); //Get a baseline reading
  Serial.print("Zero factor: "); //This can be used to remove the need to
  tare the scale. Useful in permanent scale projects.
  Serial.println(zero_factor);
}
void loop() {
  scale.set_scale(calibration_factor);
  Serial.print("Reading: ");
  Serial.print(scale.get_units(), 2);
  Serial.print(" kg");
  Serial.print(" calibration_factor: ");
  Serial.print(calibration_factor);
  Serial.println();
  if(Serial.available()) {
    char temp = Serial.read();
    if(temp == '+' || temp == 'a')
      calibration_factor += 10;
    else if(temp == '-' || temp == 'z')
      calibration_factor -= 10;
  }
}
// ----- End of load cell calibration code -----

```

```

// ----- Load cell measurement -----
#include "HX711.h"
#define calibration_factor -68000.0 //This value is obtained using the
SparkFun_HX711_Calibration sketch
#define DOUT 3
#define CLK 4
HX711 scale(DOUT, CLK);
void setup() {
  Serial.begin(9600);
  Serial.println("HX711 scale demo");
  scale.set_scale(calibration_factor);
  scale.tare(); //Assuming there is no weight on the scale at start up,
reset the scale to 0
  Serial.println("Readings:");
}
void loop() {
  Serial.print("Reading: ");
  Serial.print(scale.get_units(), 3); // reads weight and prints to serial
monitor
  Serial.print(" kg");
  Serial.println();
}
// ----- End of load cell measurement code -----

```

```

// ----- Photo interrupter ver.1 -----
// Based on code written by: Martyn Currey 16 september 2013
int ruptPin = 2; // select the input pin for the interrupter
int val = 0; // variable to store the value coming from the sensor
int count = 0; //counter
void setup(){
  Serial.begin(9600); // set up Serial library at 9600 bps
}
void loop(){
  val = analogRead(ruptPin); // read the value from the sensor
  if (val!=0){
    count++;
  }
  Serial.print("Number of rotations: ");
  Serial.println(count); // print the sensor value to the serial monitor
  delay(2);
}
// ----- End of photo interrupter ver.1 code -----

```

```

// ----- Photo interrupter ver.2 -----
// Based on code written by: Silverjoda 8 april 2015
// Target: Arduino
volatile int counter = 0;
void setup() {
  Serial.begin(9600);
  attachInterrupt(0, count, RISING);
}

```



```

void loop() {
  delay(999); // Delay almost 1 second.
  Serial.print(counter*60); // Counter * 60 seconds.
  Serial.println("rpm.");
  counter = 0;
}
void count(){
  counter++;
}
// ----- End of photo interrupter ver.2 code -----

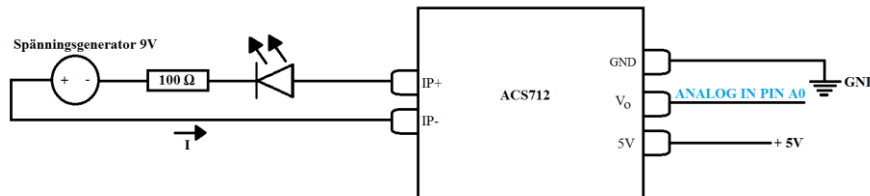
// ----- Torque and rpm measurement -----
#include "HX711.h"
#define calibration_factor -68000.0
#define DOUT 3
#define CLK 4
volatile int counter = 0;
HX711 scale(DOUT, CLK);
void setup() {
  Serial.begin(9600);
  attachInterrupt(0, count, RISING);
  scale.set_scale(calibration_factor);
  scale.tare();
}
void loop() {
  float torque = ((scale.get_units()) 10 * 0.11 * 1000); // converts kg to
N and then multiplies by length to get Nm. Multiplies by 1000 for scaling
in serial plotter
  delay(999);
  Serial.print(torque);
  Serial.print(" ");
  Serial.println(counter*60);
  counter = 0;
}
void count()
{
  counter++;
}
// ----- End of torque and rpm measurement code -----

```

## B – Tester på lösningsalternativ

### Ström- och spänningsmätning med ACS712

Kretsschemat presenteras i figur B.1 nedan.



Figur B.1. Kopplingsschema för ström- och spänningsmätning med en ACS712.

Programkoden, se nedan, som överfördes till Arduinos mikroprocessor började med att en seriell kommunikation öppnades upp till datorn. Sensorn kände av spänningen som sedan skickades till en av Arduinos analoga ingångar där spänningen sedan multiplicerades med en konstant. Denna multiplikation gjorde att strömmen genom mätobjektet erhöles eftersom sensorn hade ett linjärt samband mellan spänningen som skickades ut från  $V_O$  och strömmen i den avlästa kretsen. Slutligen skrevs det beräknade värdet ut med enheten "A" på den seriella monitorn.

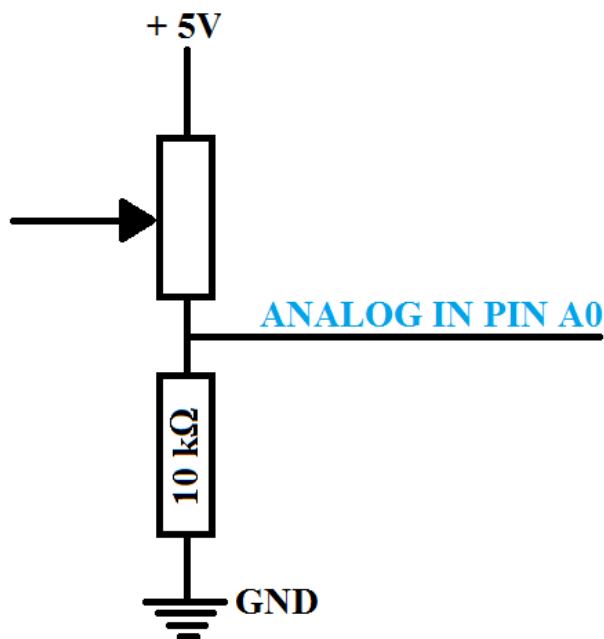
Genom att en multimeter och en AC/DC clampmeter kopplades in till kretsen kunde värdet som visas i serial monitor i Arduino bekräftas. Båda instrumenten var inställda på likströmsmätning.

Dock gav inte komponenten tillförlitliga resultat, eftersom mätvärdena var oregelbundna. Det kunde förklaras av att kalibrering av strömmen var justerbar genom att komponentens förstärkning och referensspänning ändrades via "GAIN" och "Vref". Regleringen av dessa var ytterst känslig vilket gjorde att det var nästintill omöjligt för en människa att genomföra regleringen tillräckligt korrekt för låga strömmar. Det resulterade i att strömmen aldrig fick ett tillräckligt stabilt värde, då in- och utkoppling av komponenten medförde att referensspänning och förstärkning ändrades.

```
// ----- ACS712 -----
void setup() {
  Serial.begin(9600);
}
void loop() {
  float average = 0;
  for(int i = 0; i < 1000; i++) {
    average = average + (.0264 * analogRead(A0) -13.51)/1000;
    delay(1);
  }
  Serial.print(average);
  Serial.println("A");
}
// ----- End of ACS712 code -----
```

## Tryckmätning med tryckkänsligt motstånd

Därefter kopplades en krets, enligt figur B.2, upp.



Figur B.2. Skiss på koppling mellan tryckgivare och Arduino.

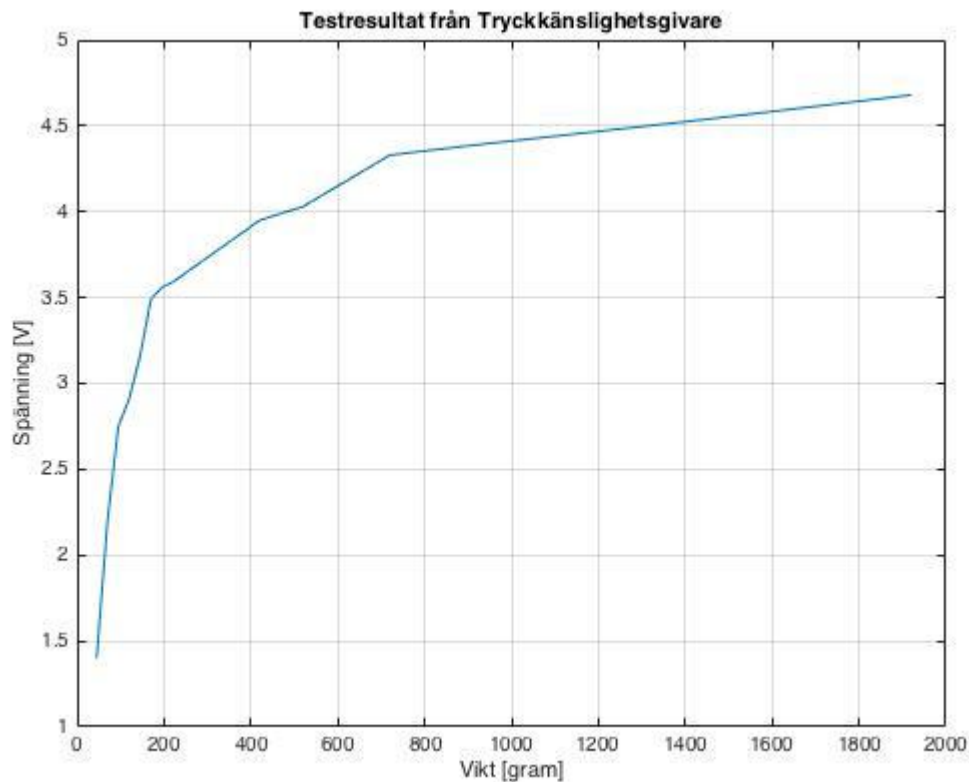
Kretsen matades med en spänning på 5 V via en Arduinoenhet och signalen som skickades från trycksensorn skickades in i en av de analoga ingångarna på Arduinon. Eftersom kretsen matades med en spänning på 5 V kunde utsignalen aldrig överstiga 5V. En resistor kopplades mellan ena trycksensorbenet och jord vilket innebar att en spänningsdelning uppstod. Storleken på resistorn valdes till 10 kΩ då det gav tillräckligt bra känslighet hos kretsen. Trycksensorgivaren tejpades fast på en våg för att en ungefärlig uppfattning av sambandet mellan spänning och vikt hos komponenten skulle kunna inhämtas.

Koden som användes för testet, se längst ner i detta avsnitt, läste in värdet som skickades från trycksensorn och omvandlade värdet till en spänning och skrev sedan ut det i Arduinons seriella monitor 10 gånger per sekund.

När ett finger lades ovanpå trycksensorn avlästes vikten av trycket på vågen samt spänningen som Arduinon presenterade på den seriella monitorn. Eftersom trycket erhöles med hjälp av ett finger kunde inte värdet bli konstant. Darrningar i handen ledde till att vikten varierade med ca  $\pm 10$  gram, dock var det försumbart för testet då inga exakta värden eftersöktes utan snarare det övergripande förhållandet mellan storheterna.

Det är värt att notera att om trycksensorn utsattes för samma tryck men med olika kontaktyta mot komponenten gav det olika utslag. En större kontaktyta erhöles en högre utsignal än vad samma tryck på en mindre yta gav.

Komponentens uppmätta karakteristik presenteras i figur B.3 nedan.



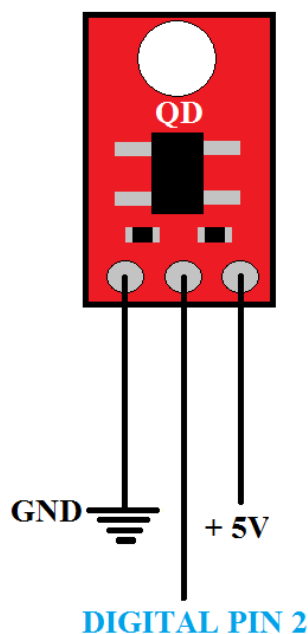
Figur B.3. Förhållandet mellan vikt och spänning hos trycksensorgivaren.

I figur B.3 syns det att vid högre vikt krävdes det större ökning av tryck för att sensorn skulle påvisa någon märkbar förändring hos spänningen.

```
// ----- Pressure sensor -----
// Author: Shriram 22 oktober 2013
void setup() {
  // Start serial at 9600 baud
  Serial.begin(9600);
}
void loop() {
  // Read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0
- 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // Print out the value you read:
  Serial.println(voltage);
  // Wait 100 milliseconds
  delay(100);
}
// ----- End of pressure sensor code -----
```

## Varvtalsmätning med reflexdetektor breakout QRE1113

Reflexdetektorn var tillkopplad mot Arduinions digitala port 2, jord och inmatningsspänning 5 V. Mer detaljerat om komponenten går att finna under kapitlet 4.7 *Reflexdetektor breakout - QRE1113*. Figur B.4 illustrerar hur testkopplingen ser ut.



Figur B.4. Skiss på uppkoppling mellan Arduino och reflexdetektor.

Mängden reflekterat ljus som träffade givaren omvandlades till ett värde som sparades till en variabel enligt koden nedtill. En räknare för varvtalet hos motorn nollställdes vid uppstart och registrerade därefter varje gång det reflekterade ljuset var ett visst värde. Eftersom ljus både skulle skickas in och ut krävdes det att en variabel deklarerades både som en in- och utgång. Detta erhöles genom funktionen "readQD()". För att kunna erhålla en uppdaterad information om ljuset var en variabel tvungen kunna skicka och ta emot data näst intill samtidigt utan att dessa krockade med varandra. En kort fördröjning löste detta. I slutet av programmet fanns en if-sats där varvtalet inkrementerades vid varje registerat värde från readQD().

När ett vitt papper med en svart linje fördes mot modulen ökade varvtalet i koden, vilket påvisade att modulen var funktionell. För att detta system skulle efterlikna projektets slutgiltiga produkt byggdes en konstruktion i LEGO upp. Där reflexdetektorn sattes fast på en arm. Intill den placerades en motor med en vitfärgad pappkartongsskiva med ett mindre, svartmålat område. Avståndet mellan detektorn och skivan var justerbar då armen kunde röra sig fritt.

```
// ----- Reflex detector -----
int QRE1113_Pin = 2;
int rpm = 0;
void setup() {
  Serial.begin(9600);
}
```

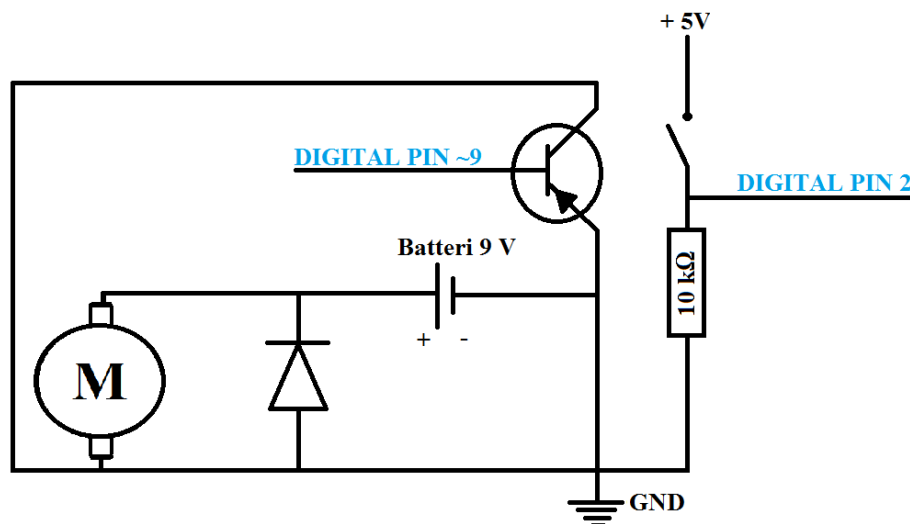
```

void loop(){
  int QRE_Value = readQD();
  Serial.print("\t Digital value from Reflexion_QRE1113: ");
  Serial.println(QRE_Value);
  if (QRE_Value > 3000) {
    rpm++;
  }
  Serial.print("Rotational speed: ");
  Serial.print(rpm);
}
int readQD(){
  //Returns value from the QRE1113
  //Lower numbers mean more reflective
  //More than 3000 means nothing was reflected.
  pinMode( QRE1113_Pin, OUTPUT );
  digitalWrite( QRE1113_Pin, HIGH );
  delayMicroseconds(10);
  pinMode( QRE1113_Pin, INPUT );
  long time = micros();
  //time how long the input is HIGH, but quit after 3ms as nothing happens
  after that
  while (digitalRead(QRE1113_Pin) == HIGH && micros() - time < 3000);
  int diff = micros() - time;
  return diff;
}
// ----- End of reflex detector code -----

```

## Motorstyrning med Arduino och transistor

För att få motorn att rotera krävdes det en högre tillförd spänning än vad Arduinokortet alstrade, därför kopplades ett batteri på 9 V till motorn. Dessutom anslöts en diod parallellt över motorn för att förhindra att kretsen skadades av backspänning, vilket kunde uppstå från motorn när den inte längre var belastad. Strömbrytaren kopplades till +5 V och till en resistor på 10 k $\Omega$  för att skydda kretsen mot överströmmar. Figur B.5 är en illustration av testkopplingen.



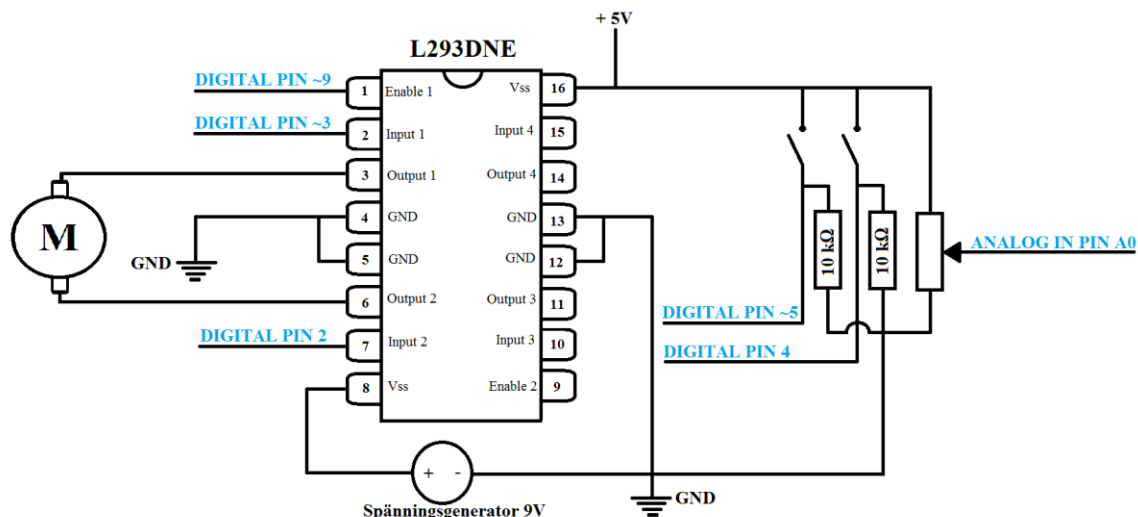
Figur B.5. Kretsschema för styrning av en motor med hjälp av en transistor.

I Arduinokoden, se under, registrerades till- och frånslagningar på strömbrytaren och tilldelades till en variabel. Denna variabel användes som ett jämförelsevärde i en if-sats för att bestämma om motorn var av eller på.

```
// ----- Motor control with Arduino & transistor -----
const int switchPin = 2;
const int motorPin = 9;
int switchState = 0;
void setup() {
  pinMode(motorPin, OUTPUT);
  pinMode(switchPin, INPUT);
}
void loop() {
  switchState = digitalRead(switchPin);
  if(switchState == HIGH) { // HIGH - motor start
    digitalWrite(motorPin, HIGH);
  }
  else {
    digitalWrite(motorPin, LOW); // LOW - motor stop
  }
}
// ----- End of motor control with Arduino & transistor code -----
```

## Test av motorstyrning med Arduino och H-brygga

I kopplingen fanns två switchar, en H-brygga av samma modell, två resistorer på 10 kΩ vardera, en spänningsgenerator och potentiometer har ersatt batteriet på 9 V från föregående test, se figur B.6. Motorns sattes på och av genom strömbrytaren som var kopplad till digital pin 5 på Arduinon. Motorns riktning ändrades när brytaren som var kopplad till digital pin 4 trycktes ner. Riktningen ändrades därför att polariteten över motorn ändrades via H-bryggan. Resistorerna var kopplade mellan brytarna och spänningsgeneratoren för att fungera som överbelastnings-skydd vid för höga strömmar. Med potentiometern kunde motorns hastighet styras.



Figur B.6. Kretsbeskrivning för motorstyrning med Arduino och H-brygga.

Med denna uppkoppling och programmeringskoden i Arduino, se hela koden längst ner i detta avsnitt, kunde motorn styras som önskat.

När koden från installationsguiden var färdiguppladdad i Arduinon krävdes en ny kod för att styra DC-motorn. Denna kod programmerades denna gång i LabVIEW och kan ses i figur B.8.

Till vänster i koden fanns blocket "Init" som var knutpunkten mellan Arduino-kortet och LabVIEW. Genom inställningarna VISA resource (Resource Name), "Bound Rate", "Board Type" och "Connection Type" valdes enligt de villkor som finns hos datorn och Arduino initierades en koppling mellan hård- och mjukvaran. För detta test valdes inställningarna enligt följande tabell B.1:

Tabell B.1. Konfiguration som ger fungerande kommunikation mellan Arduino och LabVIEW.

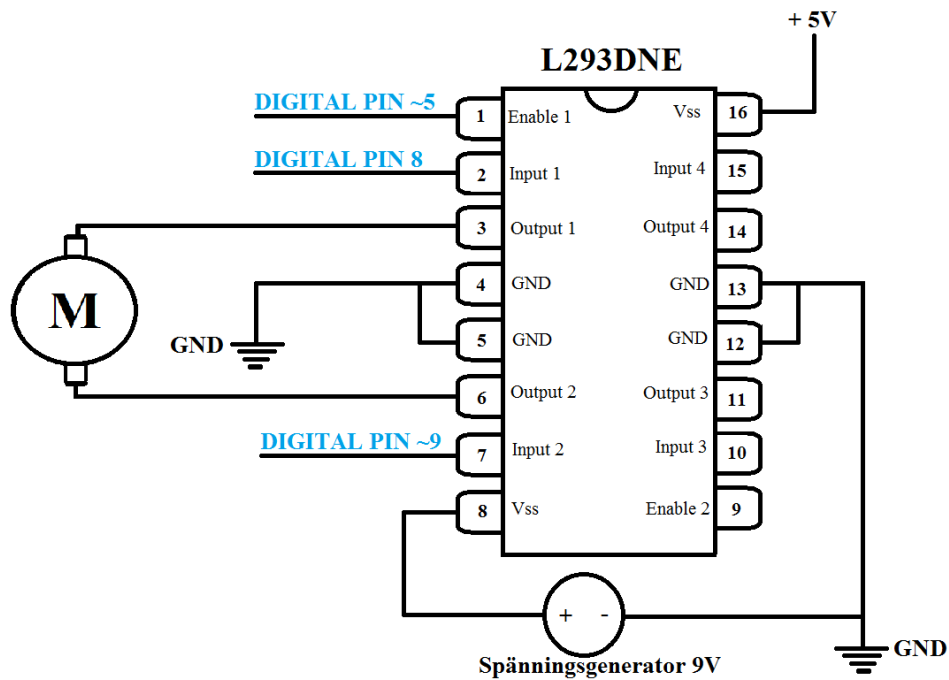
VISA resource	COM3
Bound Rate	115200
Board Type	Uno
Connection Type (USB/Serial)	USB/Serial

Blocket var kopplat till en while-loop där koden återupprepades tills den avbröts med knappen "stop". Tre av Arduino-kortets portar användes. Pin 5 användes för att kontrollera motorns hastighet, medan Pin 8 och Pin 9 styrde motorns riktning. Dessa deklarerades som utgångar med modulen "Set Digital Pin Mode". För att värdena skulle skrivas korrekt till Pin 8 och Pin 9 behövde dessa omvandlas från typen boolean till 16-bitars heltal, detta gjordes med blocket "Boolean To (0,1)". Vidare skickades värdena till "Digital Write Pin" där de skrevs ut till respektive pinne. Värdet som kom från en ratt till pin 5 var av typen double (64-bitar) som skrevs till PWM Write Pin eftersom den tog duty cycle-värden mellan 0-255. På samma kabling som till ratt, lades två moduler till en som visade motorns hastighet och en

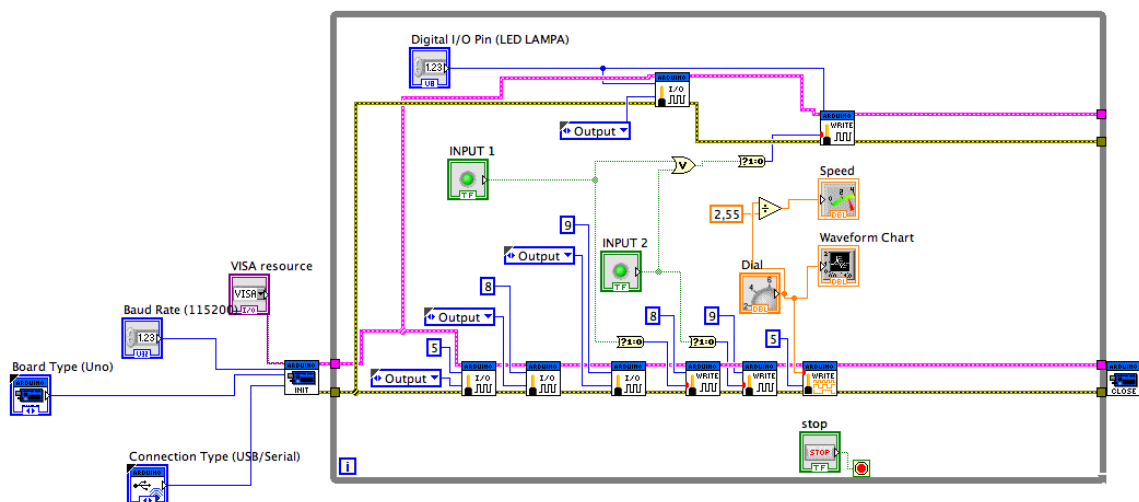


som ritade ut denna hastighet som en kurva i en graf.

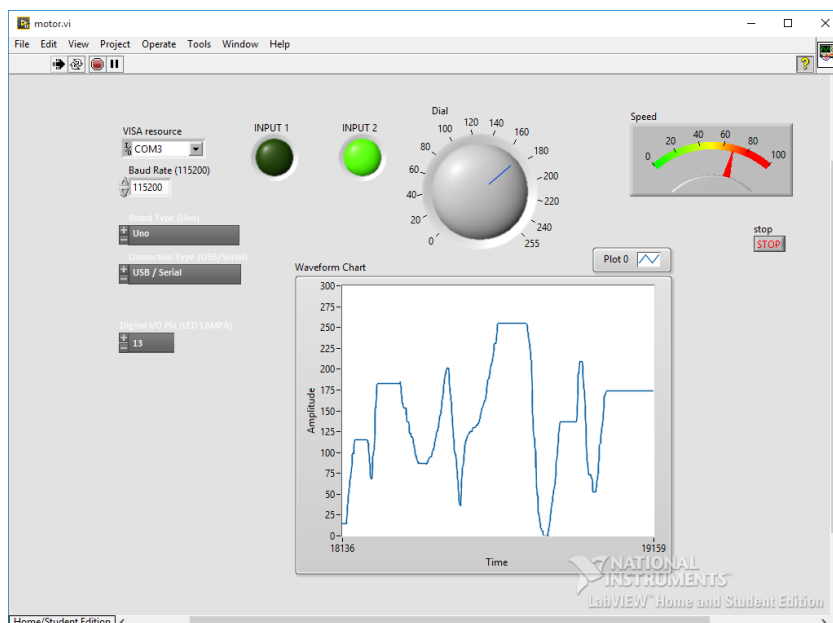
När kommunikationen var uppsatt och LabVIEW-koden var programmerad kunde komponenterna tas bort från den tidigare kretsen, vilket figur B.7 illustrerar. I kretsschemat åskådliggörs vilka av Arduinos portar som värden från LabVIEW skickade ut till. I LabVIEW styrdes signalerna som önskat för att uppnå en drivande motor som gick med en varierbar hastighet i två riktningar. Figur B.9 visar hur programmiljön såg ut när LabVIEW-koden kördes.



Figur B.7. Kopplingsschema för motorstyrning med Arduino och LabVIEW.



Figur B.8. LabVIEW-kod för test av motorstyrning med Arduino och LabVIEW.



Figur B.9. Frontpanel i LabVIEW där motorn kan, på ett illustrativt sätt, övervakas och styras.

// ----- Motor control with Arduino & h-bridge -----

```

const int controlPin1 = 2;
const int controlPin2 = 3;
const int powerPin = 9;
const int directionSwitchPin = 4;
const int OnOffSwitchStateSwitchPin = 5;
const int potPin = A0;
int OnOffSwitchState = 0;
int previousOnOffSwitchState = 0;
int directionSwitchState = 0;
int previousDirectionSwitchState = 0;
int motorPower = 0;          // motor on/off
int motorSpeed = 0;         // motor speed
int motorDirection = 1;     // motor direction
void setup() {
  pinMode(directionSwitchPin, INPUT);
  pinMode(OnOffSwitchStateSwitchPin, INPUT);
  pinMode(controlPin1, OUTPUT);
  pinMode(controlPin2, OUTPUT);
  pinMode(powerPin, OUTPUT);
  digitalWrite(powerPin, LOW);
}
void loop() {
  OnOffSwitchState = digitalRead(OnOffSwitchStateSwitchPin);
  delay(1);
  directionSwitchState = digitalRead(directionSwitchPin);
  motorSpeed = analogRead(potPin)/4;
  if(OnOffSwitchState != previousOnOffSwitchState) {          // difference
between current switch state and previous and if current switch state is
HIGH = turn on motor
    if(OnOffSwitchState == HIGH) {                             // difference and
current switch state is LOW = turn off motor

```

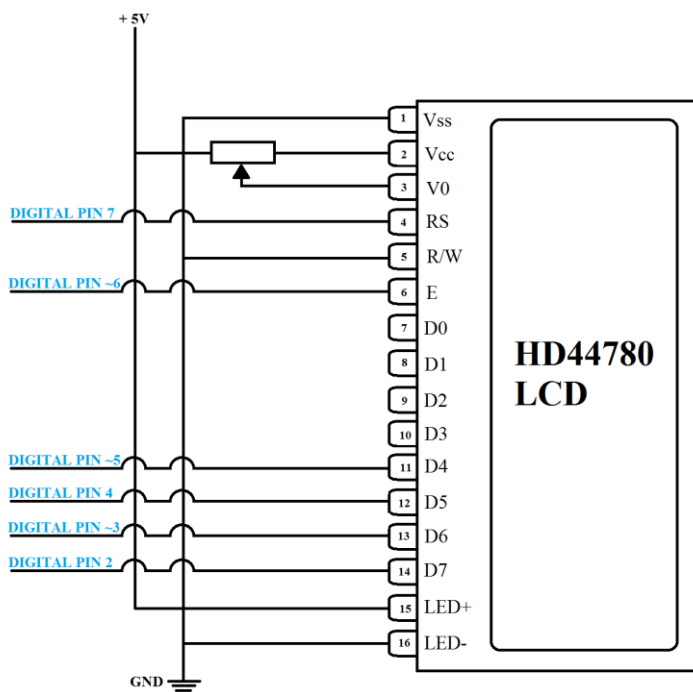
```

    motorPower = !motorPower;
  }
}
if(directionSwitchState != previousDirectionSwitchState) {           //
same principle as motor on/off however this time direction is controlled
  if(directionSwitchState == HIGH) {
    motorDirection = !motorDirection;
  }
}
if(motorDirection == 1) {      // run motor in one direction
  digitalWrite(controlPin1, HIGH);
  digitalWrite(controlPin2, LOW);
}
else {      // run motor in other direction
  digitalWrite(controlPin1, LOW);
  digitalWrite(controlPin2, HIGH);
}
if(motorPower == 1) {      // power on
  analogWrite(powerPin, motorSpeed);
}
else {      // power off
  analogWrite(powerPin, 0);
}
previousDirectionSwitchState = directionSwitchState;    // save states for
next loop
previousOnOffSwitchState = OnOffSwitchState;
}
// ----- End of motor control with Arduino & h-bridge code -----

```

## Motorstyrning med motor shield

Figur B.10 visar hur LCD - HD44780 kopplades upp.



Figur B.10. Kopplingschema för ihopkoppling av display och Arduino.

Inbyggt i en Arduino Motor Shield finns möjligheten att se hur stor ström som för tillfället används vid styrning av en motor. En liten servomotor tillkopplades, vilken sedan styrdes fram och bak i olika hastigheter, med ett par sekunders intervall. Koden för att möjliggöra detta finns att se nedtill.

Till Arduinos serial monitor skrevs värdena på strömmen kontinuerligt genom att en loop i programmet skapades. Däremot skrevs inte strömvärdena ut kontinuerligt på LCD-displayen eftersom det var svårt att, med ögat, uppfatta en stor mängd data som skrevs ut med en hastighet som var under en sekund.

```
// ----- Motor Shield and LCD test -----
// LCD declaration
#include <LiquidCrystal.h>
LiquidCrystal lcd(32, 30, 28, 26, 24, 22);
void setup() {
  lcd.begin(16,2); // Sets 16 columns and 2 rows on the LCD
  // Setup Channel A
  pinMode(12, OUTPUT); // Initiates Motor Channel A pin
  pinMode(9, OUTPUT); // Initiates Brake Channel A pin
  Serial.begin(9600); // Opens a serial port
  Serial.println("Motor Test Results");
}
void loop(){
  // Declaration of local variables:
  int count;
  // Display setup
  lcd.clear(); // Clears old results on display
  lcd.setCursor(0,0); // Sets the cursor in upper left corner on LCD
  lcd.print("Current:");
  lcd.setCursor(0, 1); // Moves cursor down a row
  // Motor runs forward at full speed
  digitalWrite(12, HIGH); // Establishes forward direction of Channel A
  digitalWrite(9, LOW); // Disengages the Brake for Channel A
  analogWrite(3, 255); // Spins the motor on Channel A at full speed
  delay(50); // Avoids high startup currents
  lcd.print(analogRead(A0)); // Prints the current in the motor once to
  // Prints a series of values of the current in the motor onto the serial
  // monitor. (More accurate than the LCD because there is a continuous stream
  // of values.)
  for(count=0; count <150; count++){
    Serial.print("Current: ");
    Serial.println(analogRead(A0));
    delay(2); //Wait for ad to stabilize
  }
  digitalWrite(9, HIGH); // Engages the Brake for Channel A
  // Readies LCD for a new current reading
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Current:");
  lcd.setCursor(0, 1);
```

```
delay(1000); // Pauses motor for a second
// Motor runs backwards at half speed
digitalWrite(12, LOW); // Establishes backward direction of Channel A
digitalWrite(9, LOW); // Disengages the Brake for Channel A
analogWrite(3, 123); // Spins the motor on Channel A at half speed
delay(50); // Avoids the high currents during startup
lcd.print(analogRead(A0)); // Prints current value to LCD
for(count=0; count <74; count++){
  Serial.print("Current: ");
  Serial.println(analogRead(A0));
  delay(2); //Wait for ad to stabilize
}
digitalWrite(9, HIGH); // Engages the Brake for Channel A
delay(1000); // Stops motor for a second
}
// ----- End of Motor Shield and LCD code -----
```

## **C – MATLABkod**

% The code is based on code originally written by Sk Saha Riaz Zaman, 29 May 2015.

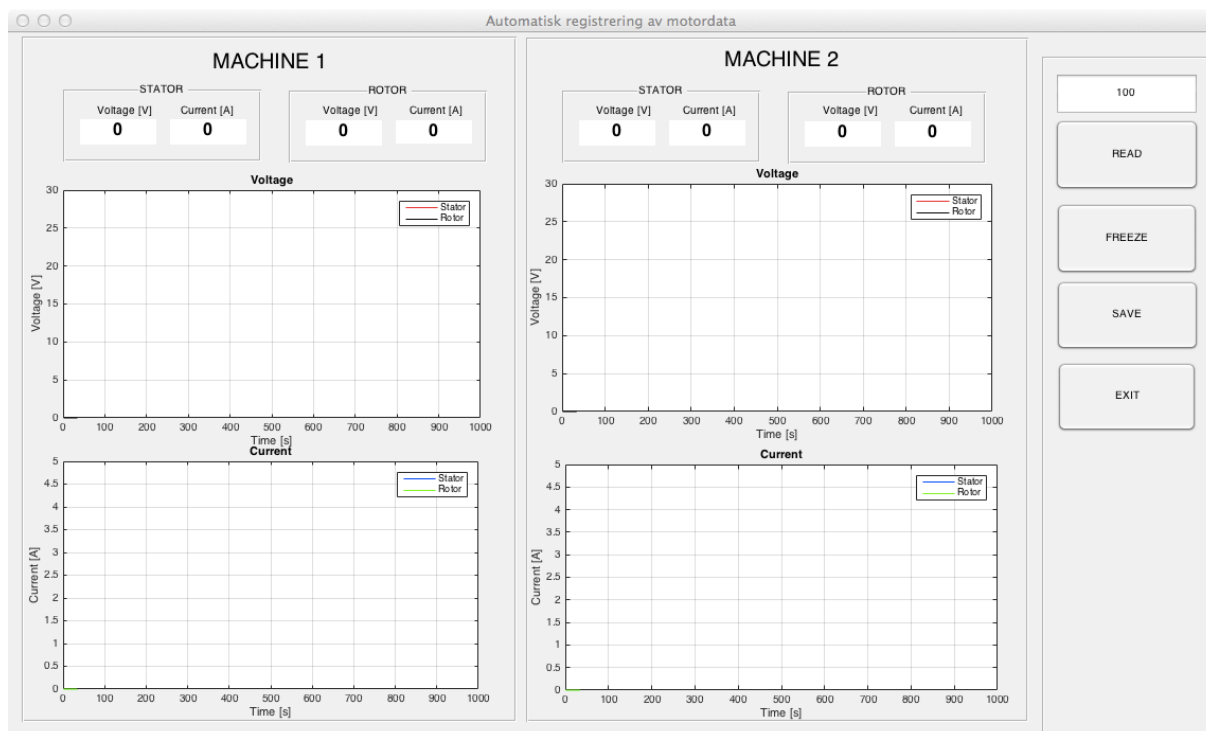
```
function varargout = MATLABGUI(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @MATLABGUI_OpeningFcn, ...
                  'gui_OutputFcn',   @MATLABGUI_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
function MATLABGUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
% or /dev/cu.usbmodem621 according to NI-VISA
delete(instrfind({'Port'}, {'/dev/cu.usbmodem621'}))
clear a;
global a;
a = arduino('/dev/cu.usbmodem621','mega2560'); % this is for Mac. In PC it
would be ex. a = arduino('COM3','Uno'). With Arduino mega write "mega2560"
configurePin(a, 'D7', 'DigitalOutput');
function varargout = MATLABGUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
% ***** READ *****
function pushbutton1_Callback(hObject, eventdata, handles)
% ***** allocations *****
global t a passo
x=0;
passo = 1;
t=1;
volt1=0;
current1=0;
volt2=0;
current2=0;
volt3=0;
current3=0;
volt4=0;
current4=0;
rpm=0;
n=0;
interv=1000; % change the the size of x-axis
while(t<interv)
% *****MOTOR 1*****
% ***** STATOR 1*****
##### Voltage1 #####
x1=readVoltage(a, 'A0');
v1=x1/0.175;
volt1=[volt1,v1];
axes(handles.axes1);
plot(volt1, 'r')
hold on
```

```

title('Voltage')
xlabel('Time [s]')
ylabel('Voltage [V]')
axis([0 interv 0 30]);
set(handles.text1, 'String',v1);
legend('Stator', 'Rotor')
grid on
##### Current1 #####
c1=readVoltage(a, 'A1');
current1=[current1,c1];
axes(handles.axes2);
plot(current1, 'b')
hold on
title('Current')
xlabel('Time [s]')
ylabel('Current [A]')
axis([0 interv 0 5]);
set(handles.text2, 'String',c1);
legend('Stator', 'Rotor')
grid on
% ***** ROTOR 1*****
##### Voltage2 #####
x2=readVoltage(a, 'A2');
v2=x2/0.175;
volt2=[volt2,v2];
axes(handles.axes1);
plot(volt2, 'k')
set(handles.text3, 'String',v2);
##### Current2 #####
c2=readVoltage(a, 'A3');
current2=[current2,c2];
axes(handles.axes2);
plot(current2, 'g')
set(handles.text4, 'String',c2);
% ***** MOTOR 2 *****
% ***** STATOR 2 *****
##### Voltage3 #####
x3=readVoltage(a, 'A4');
v3=x3/0.175;
volt3=[volt3,v3];
axes(handles.axes3);
plot(volt3, 'r')
hold on
title('Voltage')
xlabel('Time [s]')
ylabel('Voltage [V]')
axis([0 interv 0 30]);
set(handles.text5, 'String',v3);
legend('Stator', 'Rotor')
grid on
##### Current3 #####
c3=readVoltage(a, 'A5');
current3=[current3,c3];
axes(handles.axes4);
plot(current3, 'b')
hold on
title('Current')
xlabel('Time [s]')
ylabel('Current [A]')
axis([0 interv 0 5]);
set(handles.text6, 'String',c3);

```

## D – MATLAB GUI



Figur D.1. Användargränssnittet i MATLAB.

Beskrivning av kontrollpanelen, se till höger i figur D.1:

- *Number of samples* – antal samplings av det inlästa värdet.
- *READ* – startar mätningen av ström och spänning.
- *FREEZE* – fryser mätvärdena.
- *SAVE* – sparar en skärmbild av programfönstret.
- *EXIT* – avslutar programmet.

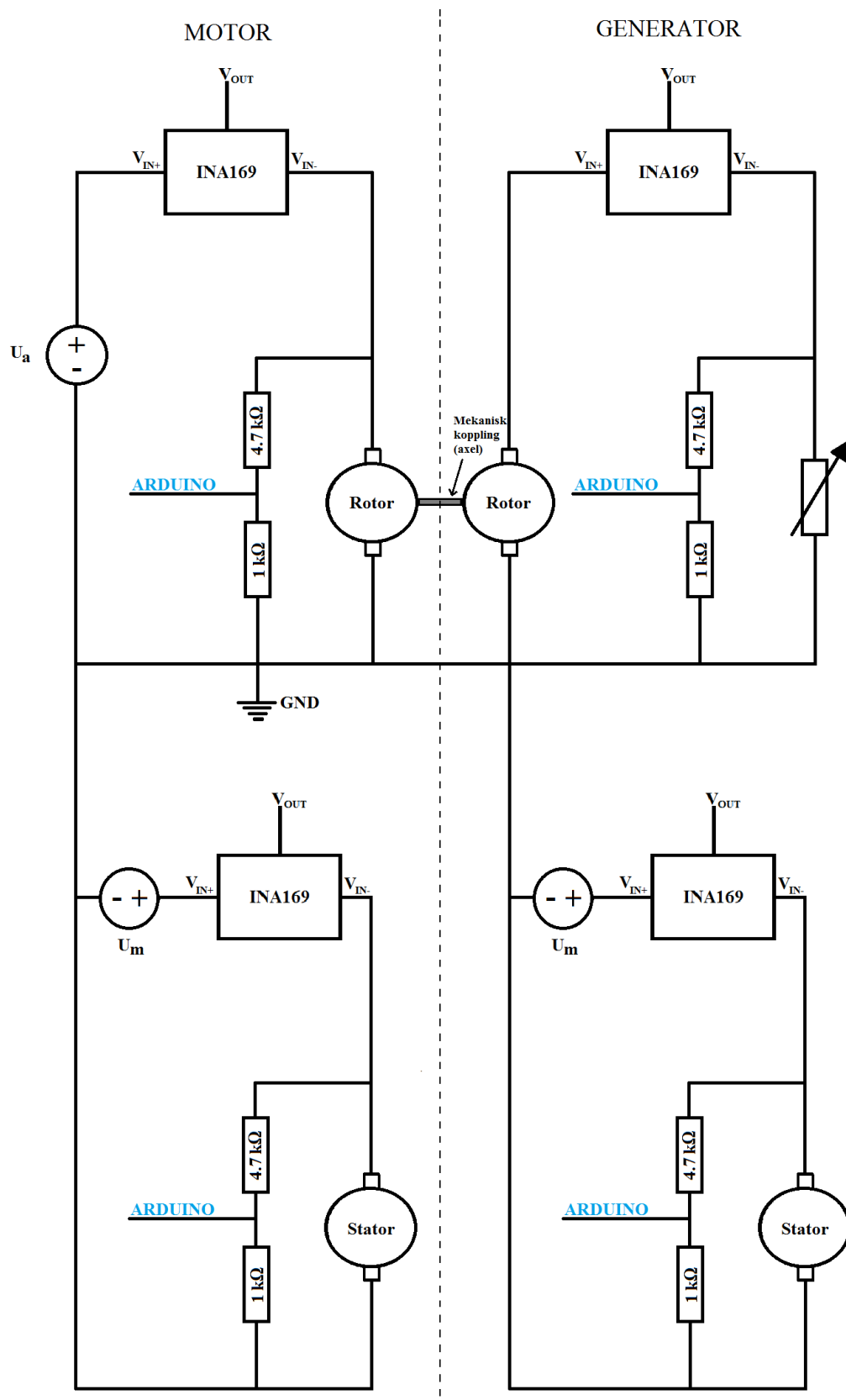


## **E – Lathund: LabVIEW + Arduino**

Följande steg utförs för att åstadkomma kommunikation mellan LabVIEW och Arduino:

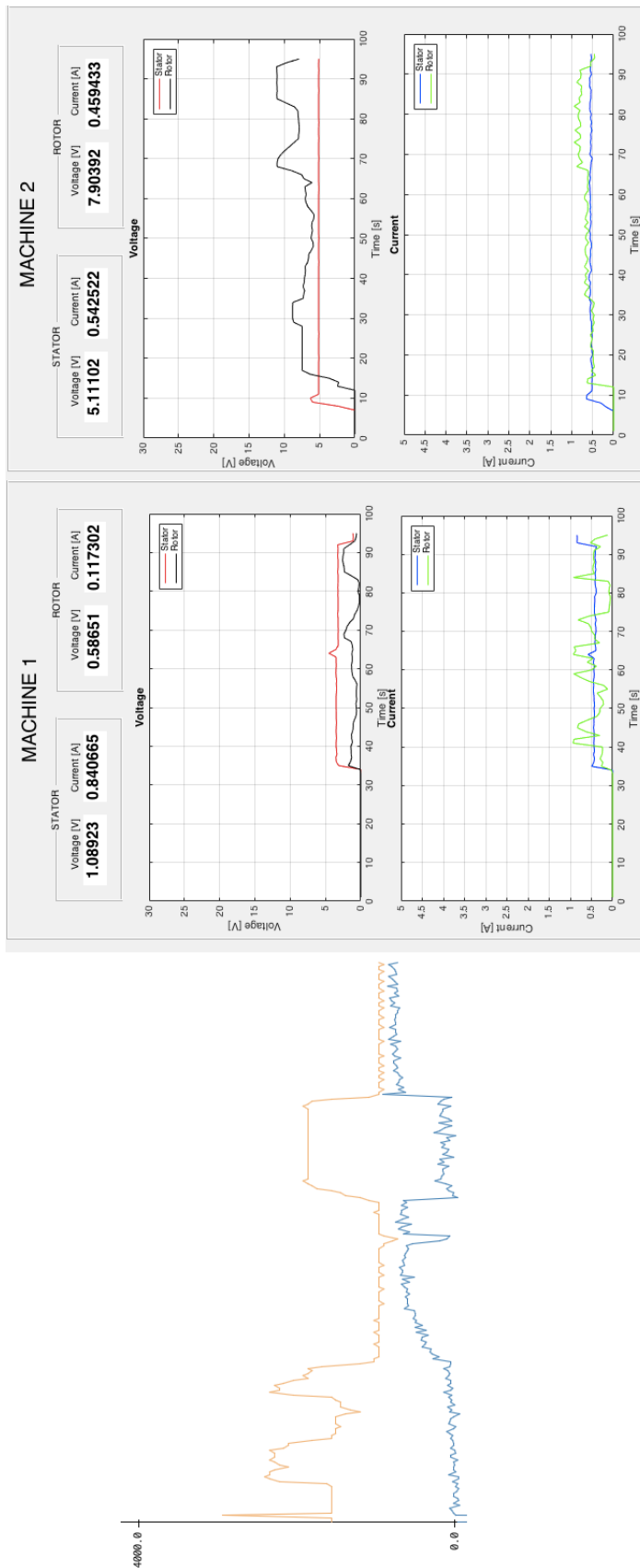
1. Installera uppsättningspaketet LabVIEW Interface for Arduino i programmet VI Package Manager: Starta programmet -> skriv i sökrutan "Arduino" -> tryck på sökträffen "LabVIEW Interface for Arduino" -> tryck på "install" (om knappen visar "uninstall" kan detta steg hoppas över).
2. Installera NI-VISA, där kan inställningar av datorns portar justeras. Välj aktuell Baud Rate och Resource Name.
3. Öppna Arduino programmet -> File -> Open... -> Välj LVIFA\_BASE filen som hittas under Local Disk (C:)/Program Files/National Instruments/LabVIEW 2015/vi.lib/LabVIEW Interface for Arduino/Firmware/LVIFA\_BASE.
4. Välj aktuell kort (Board) och serie port (Port).
5. Kompilera och ladda upp Arduino-koden.

## F – Kopplingschema för slutgiltig konstruktion



Figur F.1. Kretsschema för ström- och spänningsövervakning i slutgiltiga konstruktionen.

## G – Gränssnitt hos slutgiltig konstruktion



Figur G.1. Gränssnitt hos den slutgiltiga konstruktionen. Till vänster syns Arduino serial plotter och till höger syns MATLABS GUI.