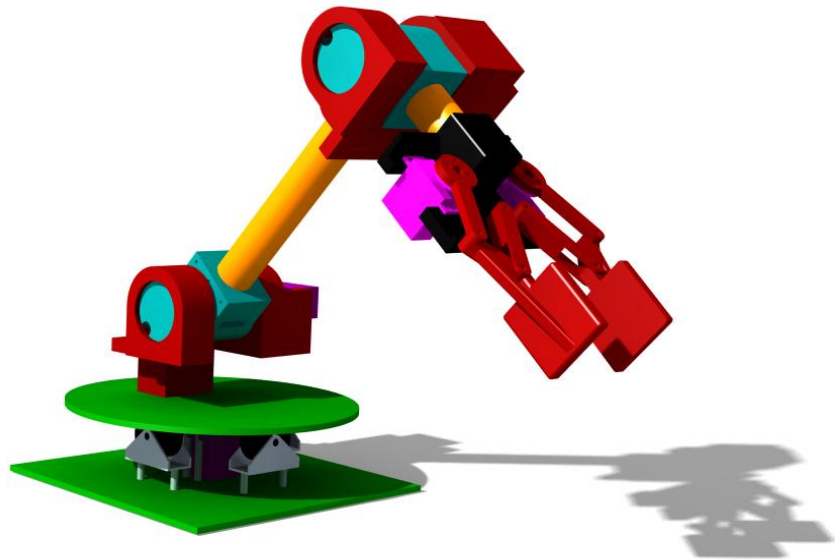




# CHALMERS

---



## Identifiering av utmaningsområden vid framtagning av robotarm

### För teknologer på kandidatnivå

Kandidatarbete inom civilingenjörsprogrammet Automation och Mekatronik

Rani El-Maalouf  
Jeanna Fahlin Strömberg  
Meltem Temur  
Frida Widelund  
Kristoffer Zakrisson

## Förord

Detta är ett kandidatarbete utfört på institutionen Signaler och System (S2) på Chalmers Tekniska Högskola vårterminen 2014. Projektet utgör totalt 15 hp fördelat på två läsperioder. Vi vill tacka:

- Vår handledare *Kristofer Bengtsson* - Technologie doktor i forskargruppen Automation
- *Göran Stigler* - Forskningsingenjör på institutionen Produkt- och produktionsutveckling
- *Payman Tehrani* - Ph.D Academic Field Sales Engineer på *National Instruments*

## Abstract

The purpose of this report is to identify the challenges that a bachelor engineering student might experience during the designing process of a robot. The aim is to present the challenges which have been identified. Both a theoretical and a practical study were performed in order to identify these. The theoretical study was based on an analysis of scientific studies, while the practical study consisted of developing a prototype of a robotic arm. The method of rapid prototyping has been used during the iteration process in order to produce components for joints and gripper by using a 3D-printer.

Developing a robot from scratch is an extensive and complicated process that requires interdisciplinary knowledge. The most difficult challenges during the process were discovered in construction, control and drive system. During construction it was hard to create a strong and lightweight prototype. Friction is another big challenge which affect the dimensioning of the drive system and the control.

The prototype was controlled by the embedded controller *CompactRIO* using the software *LabVIEW*. The challenge in controlling the prototype was to create a program structure where the drive system and the sensors can interact and also control the movements of the drive system. *LabVIEW* has numerous example programs that can be used to obtain a neat program structure. In the project sensors and a motor control profile were implemented to create controlled movements.

The process of developing a robotic arm from the beginning is a complex task and because of this a sequence of operations is needed. Since every area of the project is depending on other areas, these have to be developed in a parallel way. With this in mind, the dimensioning of the drive system was a challenge since it is depending on the fact that the other parts of the robot need to be specified first. Because of this, it is important to keep in mind that several parts need to be developed in a parallel way.

Keywords: Robot design, 3D-printing, Challenges in robot construction, Drive system

## Sammanfattning

Att utveckla en robot från grunden är omfattande, komplicerat och kräver tvärvetenskaplig kunskap. Denna rapport syftar till att identifiera vilka utmaningar teknologer på kandidatnivå kan uppleva vid konstruktion av en robot. Målet är att presentera dessa utmaningar som har identifierats genom både en teoretisk och en praktisk undersökning. Den teoretiska undersökningen baserades på analys av information från vetenskapliga publikationer och den praktiska undersökningen bestod av att utveckla en prototyp av en robotarm. Metoden *rapid prototyping* har använts under iterationsarbetet för att producera komponenter till leder och gripdon med hjälp av en 3D-skrivare.

De utmaningar som uppfattades som svårast fanns inom områdena konstruktion, styrning och drivsystem. Inom konstruktion var det en utmaning att tillverka en prototyp med hög hållfasthet och låg vikt. Friktion var en annan stor utmaning som påverkade dimensionering av drivsystem och styrning.

Robotarmsprototypen styrdes av styrplattformen *CompactRIO* och programvaran *LabVIEW*. Inom styrningen var det en utmaning att skapa en god programstruktur samt kontrollera rörelserna av drivsystemen. För att skapa en god struktur har *LabVIEW* ett flertal färdiga exempelprogram vilka kan fungera som grund. Inom projektet har sensorer och en rörelseprofil implementerats för att kontrollera prototypens rörelser.

Processen för att utveckla en robotarm från grunden är komplex och på grund av att varje område bygger på ett annat krävs det en arbetsgång där utveckling av flera områden sker parallellt. Av denna anledning var dimensioneringen av drivsystemen en utmaning då den bygger på att resterande delar av roboten är specificerade.

Nyckelord: Robotdesign, 3D-utskrivning, Utmaningar inom robotkonstruktion, Drivsystem

## Innehållsförteckning

1	Inledning.....	1
1.1	Bakgrund .....	1
1.2	Syfte och mål.....	1
1.3	Problemformulering .....	1
1.4	Avgränsningar .....	2
2	Metod.....	3
3	Teoretisk undersökning.....	4
3.1	Mekanismer .....	4
3.1.1	Robotarmsmekanismer .....	4
3.1.2	Ledmekanismer .....	6
3.1.3	Greppningsmekanismer .....	7
3.2	Kinematik .....	9
3.3	Drivsystem .....	11
3.3.1	Stegmotor .....	11
3.3.2	Servo .....	12
3.3.3	Linjärmotor .....	12
3.4	Styrning.....	13
3.4.1	Hårdvara .....	13
3.4.2	Mjukvara.....	15
3.4.3	Pulsbreddsmodulering.....	15
3.4.4	Reglering av styrsystem.....	16
4	Framtagning av robotarm .....	18
4.1	Val av konstruktion .....	18
4.1.1	Robotarmskoncept .....	18
4.1.2	Baskoncept .....	19
4.1.3	Ledkoncept .....	20

4.1.4	Länkkoncept .....	22
4.1.5	Gripdonskoncept .....	22
4.2	3D-utskrivna komponenter.....	23
4.3	Val av drivsystem .....	24
4.4	Framtagen prototyp.....	25
4.5	Kinematik .....	28
4.6	Styrning .....	30
4.6.1	Programstruktur .....	30
4.6.2	Styrning av gripdon.....	31
4.6.3	Styrning av arm.....	34
4.6.4	Reglering av servorörelse .....	36
4.6.5	Generering av PWM-signal.....	38
4.7	Elektronik .....	41
4.8	Dimensionering av drivsystem.....	43
5	Resultat .....	49
5.1	Konstruktion.....	49
5.2	Drivsystem .....	52
5.3	Styrning .....	53
6	Diskussion.....	54
6.1	Konstruktion.....	54
6.2	Drivsystem .....	55
6.3	Styrning.....	56
6.4	Fortsättningsarbete .....	57
7	Slutsats .....	58
8	Litteraturförteckning.....	59

# 1 Inledning

## 1.1 Bakgrund

Ordet robot kommer från det tjeckiska ordet *robota* som betyder tvångsarbete eller slav (1). Enligt internationell standard ISO8373 definieras robot som "automatiskt reglerbar, omprogrammerbar multifunktionell manipulator som är programmerbar i tre eller flera axlar"(2).

I dagens tillverkningsindustri spelar robotar och automationssystem en allt större roll. Fördelarna med industrirobotar är många och de utgör en viktig del inom automation. Industrirobotar ger en ökning av kvalitet och kontinuitet inom tillverkning samt högre flexibilitet och tillverkningskvantitet (1). Robotar kan utföra arbeten som är farliga eller omöjliga för människor att genomföra och kan minska driftskostnader i framförallt höginkomstländer. (3)

Robotar inspireras i vissa fall av den mänskliga anatomin. Det mänskliga skelettet består av mer än 200 ben, vilka är sammanfogade med leder. Dessa leder har rotation runt upp till tre axlar och människans rörelser genereras av totalt 640 skelettmuskler(4). Detta ger upphov till 244 kinematiska frihetsgrader(5). När endast en liten bråkdel av den mänskliga flexibiliteten implementeras i konstruktionen och styrningen av en robotarm blir komplexiteten bakom rörelser uppenbar.

En robots många fördelar och komplexitet medför svårigheter och problem vid robotkonstruktion och robotstyrning. Denna rapport avhandlar de utmaningar och problem som kan uppstå vid en enklare typ av robotkonstruktion.

## 1.2 Syfte och mål

Syftet med arbetet är att identifiera de utmaningar som teknologer på kandidatnivå kan uppleva vid konstruktion av en robot. Målet är att presentera dessa utmaningar dels utifrån teoretisk undersökning, men även utifrån praktisk tillämpning av dessa teorier.

## 1.3 Problemformulering

Uppgiften är att finna lösningar till de utmaningar som identifieras vid framtagandet av en robotarm. För att identifiera dessa utmaningar ska en prototyp av en robotarm med tillhörande gripdon tas fram och styras. Denna prototyp ska ha en precision som är tillräcklig för att flytta runda, känsliga föremål som exempelvis ett ägg utan att skada på detta uppkommer. Gripdonet ska klara av att gripa föremål med en vikt upp till 100 g och armen skall kunna förflytta föremålet mellan två punkter i ett tredimensionellt koordinatsystem.

## 1.4 Avgränsningar

Styrplattformen *CompactRIO* har använts för styrning av robotarmen då den fanns tillgänglig för projektet. Detta har medfört att andra styrplattformar inte har undersökts närmare. Programmeringsmiljön *LabVIEW* har använts eftersom *CompactRIO* är anpassat efter denna och därav har inga andra programmeringsspråk undersökts. Projektet har tillgång till en begränsad summa på 5000 kr för samtliga områden i projektet och färdiga komponenter har prioriterats framför att konstruera egna. Aspekter gällande robotarmens livscykel och dess miljöpåverkan har inte tagits i beaktande.



## 2 Metod

Målet har nåtts genom en teoretisk undersökning i kombination med ett konstruktionsarbete av en robotarmsprototyp. Den teoretiska undersökningen genomfördes i form av analys av insamlad information framtagen genom sökningar i Chalmers Discoverytjänst *Summon* och *Google Scholar*. Källor som har använts utgörs av vetenskaplig litteratur i form av böcker och artiklar samt tekniska hand- och läroböcker.

En robotarmsprototyp togs fram för att visa på de praktiska utmaningarna. Ett antal delkoncept utvecklades och dessa realiserades enligt tekniken rapid prototyping. Tekniken används för att snabbt kunna producera komplexa, tredimensionella fysiska komponenter direkt från CAD-modeller, genom till exempel 3D-skrivare (6).

### 3 Teoretisk undersökning

Detta kapitel avhandlas den teoretiska undersökningen av mekanismer, kinematik, drivsystem och styrning för en robotarm.

#### 3.1 Mekanismer

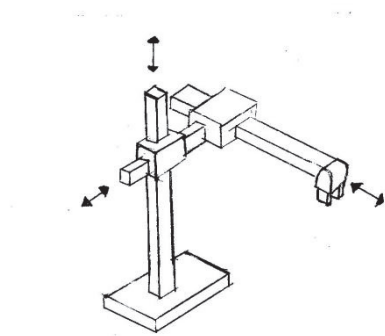
Detta avsnitt behandlar de olika mekanismer vilka ger en robotarm dess rörelse. Ordet mekanism definieras enligt Nationalencyklopedin som; *“en anordning av mekaniska komponenter som åstadkommer eller överför en rörelse av något slag”* och spelar en huvudsaklig roll vid konceptframtagning. (7)

##### 3.1.1 Robotarmsmekanismer

En industrirobot består av ett flertal stela kroppar i form av länkar, vilka är seriekopplade med hjälp av leder. Dessa bildar tillsammans en arm som i ena änden är fäst i en bas och vars andra ände är utrustad med ett verktyg. (8)

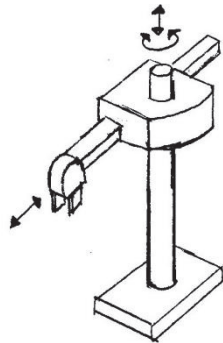
I det tredimensionella rummet finns det sex translationsfrihetsgrader, tre linjära längs de typiska x-,y- och z-axlarna samt tre rotationsfrihetsgrader kring respektive axel. En enkel robotarm rör sig typiskt i tre frihetsgrader, som tillsammans positionerar verktyget till önskat läge. Det finns ett flertal möjliga robotarmsmekanismer som kan användas för att åstadkomma tredimensionella rörelser. Tre av dessa baseras på de vanligaste koordinatsystemen:

- Kartesisk koordinatprincip: tre linjära axlar, se Figur 1. (8)



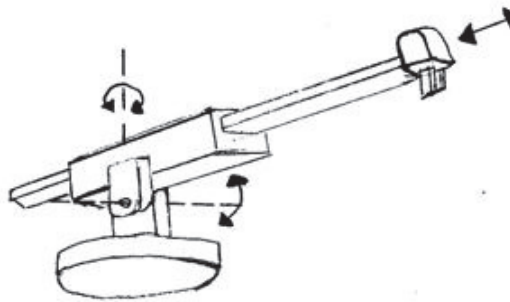
Figur 1 - Kartesisk koordinatprincip.

- Cylindrisk koordinatprincip: två linjära axlar och en rotationsaxel, se Figur 2. (8)



Figur 2 - Cylindrisk koordinatprincip.

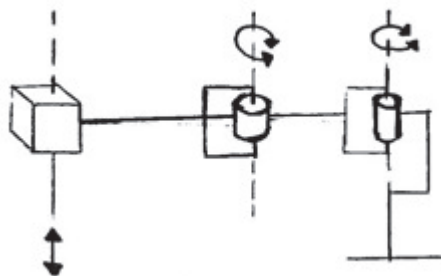
- Sfärisk koordinatprincip: en linjär axel och två rotationsaxlar, se Figur 3. (8)



Figur 3 - Sfärisk koordinatprincip.

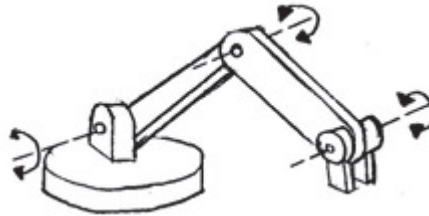
Det finns därutöver två mekanismer vilka saknar analogi till de vanligaste koordinatsystemen, men som är kapabla att lokalisera verktyget i rummet.

- Skalär robot: består av två rotationsleder och en linjärled. Strukturen är fördelaktig i produktionssystem då den har en stor arbetsyta horisontellt och samtidigt kan röra sig vertikalt, se Figur 4. (9)



Figur 4 - Skalär robot.

- Artikulerad robot: består av tre rotationsleder, likt en mänsklig arm, och kallas även för armbågsrobot. Denna robot har en stor flexibilitet och mångsidighet samt är den vanligaste strukturen för robotarmar, se Figur 5. (9)



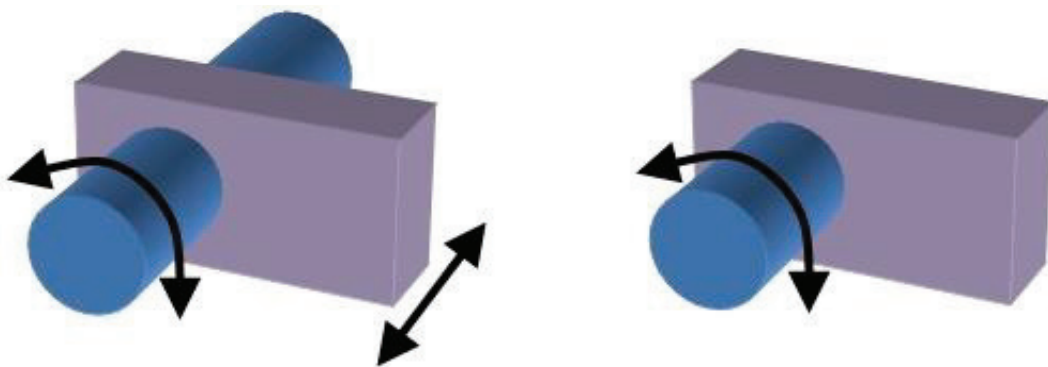
Figur 5 - Artikulerad robot.

### 3.1.2 Ledmekanismer

För att robotarmen skall kunna nå samtliga punkter inom dess räckvidd, så är det nödvändigt att armen kan vridas och böjas i olika vinklar.

#### Cylindrisk led

Den cylindriska leden är en vanligt förekommande typ av rotationsled, som tillåter rörelser i två frihetsgrader, vilken visas till vänster i Figur 6. Den cylindriska leden består av två komponenter som monteras mot varandra så att de båda komponenternas rotationscentrum sammanfaller. Är en av komponenterna ihålig så kan denna komponent rotera kring samt glida längs den cylindriska komponenten, vilket tillåter rörelser i två frihetsgrader. Den cylindriska leden kan låsas till rörelser i en frihetsgrad, en så kallad gångjärnsled, vilken visas till höger i Figur 6. Genom en statisk montering i längsgående led så hindras den ihåliga komponenten från att glida längs den cylindriska komponenten. (10)



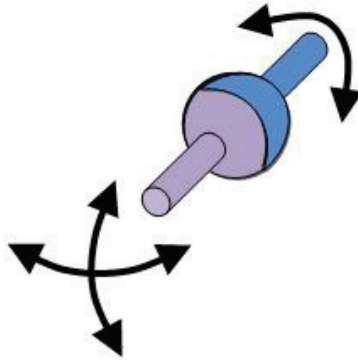
Figur 6 - Cylindrisk led med 2 frihetsgrader samt cylindrisk gångjärnsled med 1 frihetsgrad.

För att minska friktionen, som tenderar att minska rörelsehastigheten samt nötta sönder materialet i leden, används lagrade leder. Lagret minskar friktionen genom att låta släta

rullar eller kulor glida och rulla mot släta metallytor. Lasten som uppstår i leden bärs upp av dessa rullar eller kulor, vilket ger leden möjlighet att operera mjukt. (11)

### Kulled

Kulleden används för att ge möjlighet till rörelse med tre frihetsgrader, x-, y- och z-led. De tre frihetsgraderna uppnås i och med att leden är av sfärisk form, enligt Figur 7. (12)



Figur 7 – Kulled.

### Linjärled

Linjärled, se Figur 8, även kallad prismatisk led, är en led som ofta används i hydrauliska system. Robotar består ofta av två typer av leder, gångjärnsleder och prismatiska leder. Den prismatiska leden ger möjlighet till linjära rörelser, rörelser i en frihetsgrad, längs ledens egna axel. (13)



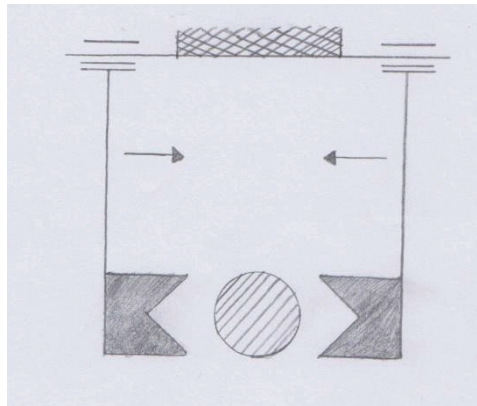
Figur 8 – Linjärled.

### 3.1.3 Greppningsmekanismer

Ett gripdon är en sorts verktyg, ofta bestående av två till fyra kontaktpunkter, som används för att kunna greppa ett föremål (14). För att ett gripdon ska kunna stängas och öppnas, kan ett flertal mekanismer utnyttjas vilka beskrivs nedan.

### Parallell

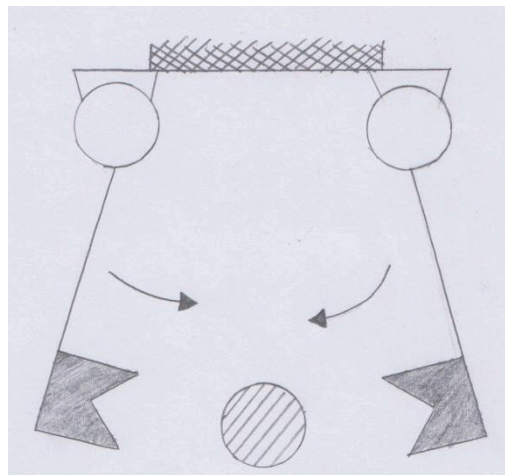
En parallell mekanism innebär att greppytorna på gripdonet rör sig längs en rak linje vid öppning och stängning, se Figur 9. För att uppnå en parallell rörelse krävs det i många fall en komplicerad kugghjulsmekanism. (14)



Figur 9 – Parallell, inspirerad från Robot Grippers.

### Cirkulär

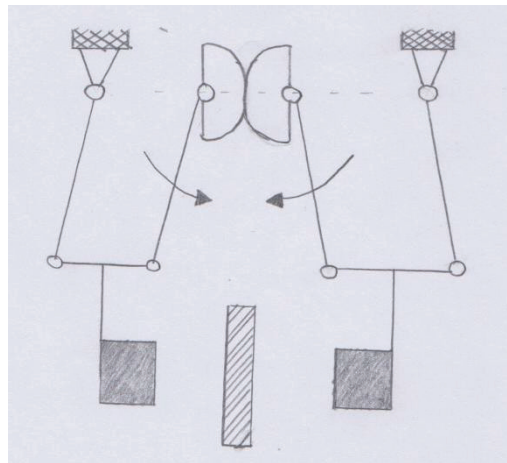
Den cirkulära mekanismen, se Figur 10, kännetecknas av en rörelse kring en fix punkt där de båda greppytorna under rörelse formar en cirkel. (14)



Figur 10 – Cirkulär, med inspiration från Robot Grippers.

### Cirkulär translation

En cirkulär translation, se Figur 11, innebär att greppytorna rör sig parallellt i förhållande till varandra i en cirkulär rörelse. (14)



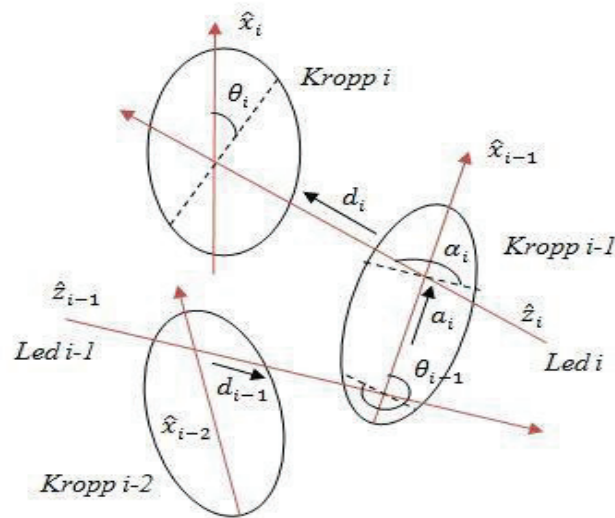
Figur 11 - Cirkulär translation, med inspiration från Robot Grippers.

### 3.2 Kinematik

Kinematik är en del inom mekaniken vilken beskriver rörelsen av kroppar utan att krafter tas i beaktande.(15) Dynamik å andra sidan är den del som beskriver kroppars rörelse med hänsyn taget till rörelsens orsaker(16).

Rörelsen hos en robotarm beskrivs av robotarmsdynamik som består av matematiska formuleringar baserade på olika ekvationer. Sådana ekvationer kan vara användbara vid datorsimulering, regleringsekvationer och utvärdering av kinematisk design av en robotarm.(8)

En robotarms geometri beskrivs vanligtvis genom att definiera referensramar till varje länk. Referensramarna skulle kunna placeras godtyckligt, men det kan vara fördelaktigt att systematiskt ange dessa med hjälp av någon typ av konvention för att få god struktur och en ökad beräkningseffektivitet vid problemlösning. En grundläggande konvention är Denavit och Hartenbergs, vilken har legat som grund till en rad andra konventioner, bland annat Khalil och Dombres. Det krävs fyra parametrar för att kunna lokalisera en referensram i förhållande till en annan i alla konventioner som bygger på Denavit-Hartenberg-konventionen. De fyra parametrarna är: länklängden  $a_i$ , vridningsvinkel för länk  $\alpha_i$ , förskjutningsavståndet  $d_i$  och ledvinkeln  $\theta_i$ , se Figur 12 som är inspirerad av (17). Fyra parametrar är relativt få, detta eftersom att antalet parametrar har minimerats genom smarta placeringar av referensramarnas origo och axlar. Referensramarna orienteras på ett sådant sätt att  $\hat{x}$ -axeln både skär och är vinkelrät mot  $\hat{x}$ -axeln hos föregående referensram. Konventionerna kan användas för robotarmar bestående av gångjärns- eller linjärleder, men eventuella kullleder kan beskrivas som kombinationer av dessa två. (17)



Figur 12 - Schematisk bild vilken beskriver numrering av kroppar och leder med tillhörande geometriska parametrar, för en robotarm.

Khalil och Dombres konvention kan sammanfattas enligt följande:

Antalet kroppar i robotarmen definieras enligt följande:

- Antalet kroppar  $N$  är numrerade från 1 till  $N$ , där basen betecknas som 0.
- Antalet leder  $N$  är numrerade från 1 till  $N$ , där led  $i$  är placerad mellan kropp  $i-1$  och kropp  $i$ .

Numreringen ovan leder till följande referensramskoordinatsystem:

- $\hat{z}_i$ -axeln är placerad längs axeln för led  $i$ .
- $\hat{x}_{i-1}$  är placerad längs med den gemensamma normalen mellan  $\hat{z}_{i-1}$  och  $\hat{z}_i$  axlarna.

Parametrarna för lokalisering av referensramar i förhållande till varandra blir då:

- $a_i$  är avståndet mellan  $\hat{z}_{i-1}$  till  $\hat{z}_i$  längs med  $\hat{x}_{i-1}$ .
- $\alpha_i$  är vinkeln från  $\hat{z}_{i-1}$  till  $\hat{z}_i$  kring  $\hat{x}_{i-1}$ .
- $d_i$  är avståndet från  $\hat{x}_{i-1}$  till  $\hat{x}_i$  längs med  $\hat{z}_i$ .
- $\theta_i$  är vinkeln från  $\hat{x}_{i-1}$  till  $\hat{x}_i$  kring  $\hat{z}_i$ .

(17)



Positionen och orienteringen av en stel kropp i rummet kallas tillsammans för pose. Robotkinematik beskriver pose, hastighet, acceleration och alla högre derivator av posen hos kropparna som ger robotmekanismen. (17)

Inom robotik används direkt och invers kinematik, vilka är metoder som används för att beräkna vinklar och positioner(8). Det direkta kinematikproblemet formuleras generellt som: finn den relativa positionen och orienteringen av två delar i en mekanism, givet den geometriska strukturen av roboten och värdena för ledpositionerna för varje frihetsgrad av mekanismen. För en robotarm blir det direkta kinematikproblemet att finna positionerna och orienteringen av den definierade centrumpunkten på verktyget, TCP (Tool Center Point), relativt basen givet positionerna av alla leder och tillhörande geometriska länkparametrar. (17)

Det inversa kinematikproblemet kan generellt formuleras som: att givet de relativa positionerna och orienteringarna av två delar i en mekanism, finna värdena för alla ledpositioner. För en robotarm används invers kinematik för att finna värdena för ledpositionerna, givet positionen och orienteringen av TCP i gripdonet relativt basen och värdena för de övriga geometriska länkparametrarna. (17)

Det finns olika sätt att lösa det inversa kinematikproblemet: homogen transformationsmatris och metoder som ger slutna ekvationssystem. Vid användning av det förstnämnda sättet kan numeriska metoder komma att krävas för att lösa olinjära ekvationer. I detta fall kan de andra metoderna vara fördelaktiga, eftersom de är snabbare än numeriska samt lätt hittar alla möjliga lösningar. Nackdelen för dessa snabbare metoder är att de inte är generella utan är olika för varje robottyp. (17)

Det finns två metoder som ger slutna ekvationssystem: algebraisk och geometrisk. De algebraiska metoderna innebär en identifiering av signifikanta ekvationer, bestående av ledvariabler och en modifiering av dessa ekvationer till en lösbar form. I geometriska metoder identifieras punkter på roboten utifrån vilka positioner och/ eller orienteringar kan uttryckas som en funktion av ledvariabler. De resulterande ekvationerna löses sedan genom algebraiska operationer. (17)

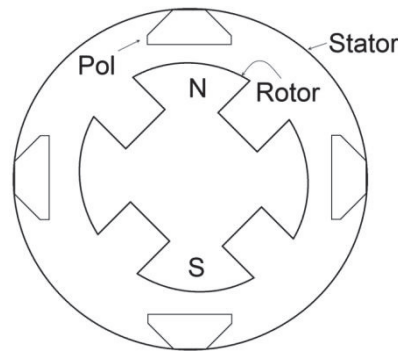
### 3.3 Drivsystem

En robot kan drivas av flera olika typer av drivsystem för att generera en rörelse genom dess mekanismer.

#### 3.3.1 Stegmotor

En stegmotor, se Figur 13, är en typ av elektrisk motor med tillhörande rotor och stator. Då statorns lindningar matas med strömpulser vrids rotorn en viss bestämd vinkel, ett så kallat steg. Antalet poler i statorn har en avgörande roll för hur stort steget är, längden på

steget minskar med ökat antal poler i statorn.(18) Vid höga hastigheter är rörelsen jämn, men i låga hastigheter är den ofta hackig. Dessa kan även förlora sin positionsuppfattning om återkoppling saknas. (19) Vanligt förekommande användningsområden för stegmotorer är i utrustningar för varvtals- och positionsreglering.(18)



Figur 13 - Schematisk bild av en stegmotor, inspirerad av Nationalencyklopedin.

### 3.3.2 Servo

Servo används ofta för krävande applikationer där hög precision för kraft- och positionsreglering är centralt. Denna typ av drivsystem skiljer sig från stegmotorer då servos använder sig av någon typ av återkoppling. Det finns två olika alternativ när det kommer till servo, digital och analog. Den digitala har en kortare insvängningstid och högre upplösning på grund av den digitala mikroprocessorn och ger därmed sitt maxmoment från start. Nackdelen med digitala servon är att de, på grund av de tidigare nämnda egenskaperna, har en högre energiförbrukning och att det i många situationer krävs en högre prisklass för att få ett digitalt servo med lika bra prestanda som ett analogt. (20)

Servon läser av sin vinkelposition, vanligtvis med hjälp av en potentiometer. En potentiometer är en resistor med en glidkontakt, där denna kontakt kan variera den elektriska resistansen (21). Relationen mellan potentiometerns resistans och glidkontaktens läge är antingen linjär eller logaritmisk (22). Potentiometern måste vara placerad på ett sådant sätt att glidkontakten får en direkt kontakt till hubben, motoraxelns yttre fäste. På så vis kommer resistansen att variera i proportion till hubbens rörelse och genom mätning av resistansen kan vinkeln bestämmas.

### 3.3.3 Linjärmotor

En linjärmotor producerar en linjär rörelse istället för en roterande och kan liknas vid en öppnad och utrullad roterande induktionsmotor. Linjärmotorns funktion baseras på att dess stator består av flera permanentmagneter placerade i linje och dess translator består

av ett antal järnkärnor med spolar. Genom att applicera en trefasström till tre angränsande spolar i translatoren genereras en sekvens av attraherande och repellerande krafter, vilka ger en tryckkraft på translatoren som då rör sig i en linjär rörelse. Fördelarna med en linjärmotor är att det uppkommer mindre friktion och andra motreaktioner i jämförelse med den traditionella roterande motorn. En minskning av motreaktioner kan bidra till en högre noggrannhet. Accelerationen och hastigheten begränsas inte av det mekaniska utan av bandbredden på positionsmätningssystemet och kraftelektroniken. På grund av linjärmotorns mekaniskt enkla konstruktion kan även en lång livstid och hög tillförlitlighet förväntas. (23) En linjärmotors effektivitet är dock något mindre än en roterande motor och den har dessutom en större strömförbrukning än en roterande motor. Detta på grund av att luftgapet mellan rotor och stator vanligtvis är större vilket leder till att det krävs mer ström för att producera magnetfältet. (24)

### 3.4 Styrning

En kontrollerad styrning och hög precision kan uppnås genom att använda sensorer som läser in information från omgivningen. Detta avsnitt beskriver hur en robot kan styras med hjälp av hårdvara, mjukvara och reglering.

#### 3.4.1 Hårdvara

*CompactRIO* cRIO-9074 är en kombinerad kontroll- och datainsamlingsplattform, tillhandahållen av *National Instruments*, se Figur 14. Denna kan användas i både industriella och akademiska tillämpningar. *CompactRIO* utgörs av en realtidsstyrenhet, utbytbara I/O-moduler och en FPGA-modul (Field Programmable Gate Arrays), vilka ryms i ett chassi med ethernetutgång. (25) FPGA, på svenska översätts som *programmerbara grindmatriser*, är en halvledarkomponent baserad på en matris bestående av logiska grindar vars anslutningar är omprogrammerbara. (26)



Figur 14 - CompactRIO med tillhörande moduler.

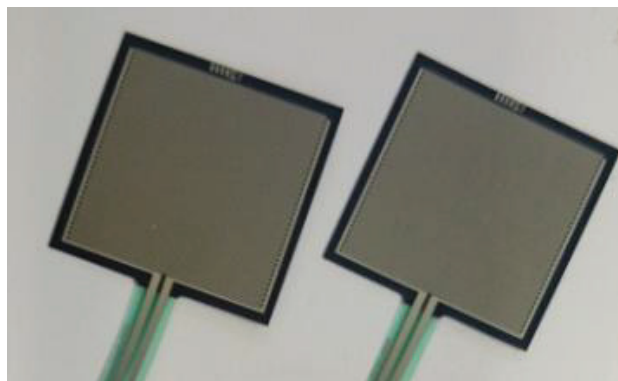
Det finns en stor uppsättning I/O-moduler för läsning av både digitala och olika typer av analoga signaler som spänning och resistans.

NI 9403 är en digital modul med 32 kanaler vilka kan konfigureras för både in- och utdata. (27) Till denna modul kopplas en digital sidecar. Digital sidecar är designad för att anpassa de digitala kanalerna till ett set av olika in- och utgångar, som till exempel PWM-utgångar vilket går igenom i 3.4.3. (28)

NI 9201 är en analog ingångsmodul med åtta kanaler samt en samplingshastighet på 500 kS/sekund och upplösning på 12 bit. Inspänningsområdet är mellan -10V och +10 V. (29) Med den analoga ingångsmodulen kan information från sensorer läsas in till *CompactRIO*.

En sensor är en anordning som känner av en förändring eller ett värde av en fysikalisk storhet. Exempel är temperatur, tryck eller hastighet som av en sensor omvandlar detta till information som är lämplig för datainsamlade system som till exempel spänning. (30)

Om ett föremål ska gripas är första steget att detektera föremålet som ska greppas. Det finns två huvudsakliga metoder vid kontaktavkänning; kinestetisk och taktil. Den förstnämnda mäter rörelse av arm och krafter med hjälp av inre sensorer. Den andra använder sensorer i kontaktytan för att känna av kontakt och en sådan återkoppling ger en mycket högre precision. Det finns flera sätt att mäta taktil aktivitet, till exempel med; piezoelektrisk, piezoresistiv, kapacitiv och induktiv sensor. En piezoelektrisk sensor konverterar tryck eller drag till en elektrisk signal. Den piezoresistiva sensorn ändrar sin resistans relativt den pålagda kraftens storlek. Kapacitiv avkänning använder föremålets ledningsförmåga istället för fysisk avkänning för att avgöra och förändra föremålets position. Den induktiva och den kapacitiva sensorn har många gemensamma egenskaper, dock använder den induktiva ett magnetfält istället för ett elektriskt. Denna är mindre känslig relativt den kapacitiva sensorn, men anses passa bättre i strängare miljöer. (31)



Figur 15 - Två FSR – sensorer.

En FSR (Force Sensing Resistor) sensor, se Figur 15, är en typ av piezoresistiv sensor, vars resistans minskar när en pålagd kraft ökar på det aktiva området på sensorn. FSR-sensorer består av tre lager; ett undre, ett mellan, och ett övre lager. Det undre lagret består av ett flexibelt substrat med utskrivna, interdigitala elektroder medan det övre lagret består av ett flexibelt substrat med utskrivna halvledare. Mellanlagret fungerar som en isolator mellan de andra två ledare. Relationen mellan kraft och resistans är olinjär. (32)

En annan piezoresistiv sensor är trådtöjningssensorn med vilken längdförändringar kan bestämmas genom att mäta hur resistansen förändras i en metalltråd som deformeras. Genom dessa mätningar kan ett lokalt töjningstillstånd i föremålet, eller storleken på belastningen bestämmas, om dess elastiska egenskaper är kända. (33)

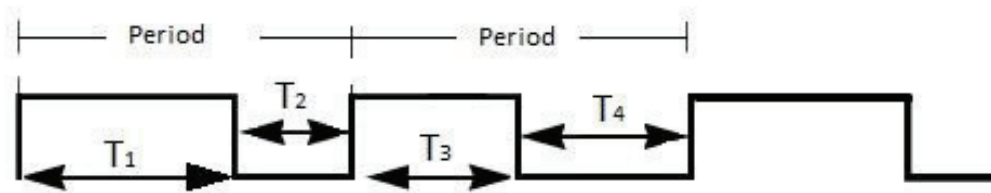
#### 3.4.2 Mjukvara

*CompactRIO* är anpassad till den grafiska programmeringsmiljön *LabVIEW*, *Laboratory Virtual Instrumentation Engineering Workbench*, från *National Instruments*. (25) *LabVIEW*-program kallas för *Virtual instruments*, förkortat VI, eftersom programmets utseende och operationer påminner om fysiska instrument såsom oscilloskop och multimeter.

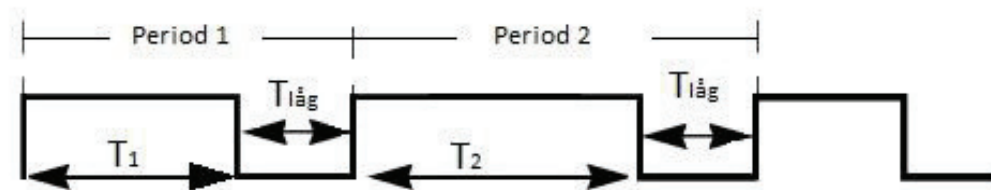
Programmen består av två fönster: *block diagram* (blockdiagram) där all kod skrivs och *front panel* (användargränssnitt) där indatamekanismer som knappar och skjutreglage samt indikatorer som grafer och LED-lampor finns. (34) Exempel på tillämpningsområden av *LabVIEW* är inläsning och analys av mätdata, styrning och övervakning av inbyggda system samt automatiserade testsystem. *LabVIEW*-koden kan exekveras på *CompactRIO*s processor utan att vara kopplad till värddatorn. (25)

#### 3.4.3 Pulsbreddsmodulering

Pulsbreddsmodulering (förkortas PWM från engelska termen *Pulse Width Modulation*) är en metod för att kontrollera en analog krets genom en microprocessors digitala utgångar. Den analoga signalen skapas digitalt genom att ett digitalt pulståg genereras vilket har de analoga signal-nivåerna. När en PWM-signal ska designas är frekvensen den viktigaste faktorn. Det finns två varianter av PWM-metoder; en där frekvensen varierar och en där frekvensen är konstant. Att den har konstant frekvens innebär att summan av tiden för låg period och hög period är konstant, det vill säga den har konstant periodtid, vilket ses i Figur 16. PWM-metoden med varierande frekvens innebär att periodtiden varierar från period till period. Då är det antingen hög period som är konstant och låg period varierar eller att hög period är konstant och låg period varierar. I Figur 17 ses en PWM-signal med varierande frekvens där den låga perioden är konstant och den höga perioden varierar. (35)



Figur 16 – PWM-signal med konstant frekvens, där hög och låg period varierar med varandra.



Figur 17 - PWM-signal med varierande frekvens. Konstant låg period och varierande hög period.

Den analoga kretsen som styrs av en PWM-signal kommer att få en signal som utifrån PWM-signalens *duty cycle* får en viss analog spänningsnivå. *Duty cycle* är en procentsats som säger hur stor del av signalen som är hög, så att om en signal har 20 % *duty cycle* innebär det att den har hög signal i 20 % av periodtiden. En PWM-signals *duty cycle* beräknas enligt ekvation 1. (36)

$$Duty Cycle = \frac{T_1}{T_1 + T_{låg}} \quad (1)$$

Den analoga spänningsnivån som levereras till den PWM-styrda kretsen fås genom att multiplicera signalens *duty cycle* med matningsspänningen. (36)

#### 3.4.4 Reglering av styrsystem

För att styrningen av ett drivsystem ska vara stabil och effektiv krävs regulatorer. Den enklaste formen av regulator är P-regulatorn vilken innebär proportionell reglering. Styrsignalen  $u(t)$ , signalen som skickas från regulatorn, är proportionell mot regulatorns insignal, reglerfelet  $e(t)$ , med förstärkningskonstanten  $K_p$  enligt ekvation 2. Reglerfelet är differensen mellan referenssignalen och den verkliga utsignalen. (37)

$$u(t) = K_p * e(t) \quad (2)$$

Risken med denna typ av regulator är att det kan bli ett kvarstående fel om en konstant processtörning uppstår, till exempel en stegstörning. För att kompensera för den typen av störning krävs ett stort  $K_p$ , så stort att systemet inte klarar av att svängas in vid ändring av referenssignalen, det vill säga systemet blir mycket ostabilt. Överlag ska  $K_p$  hållas så låg som möjligt för att styrsignalaktiviteten ska minimeras, men samtidigt tillräckligt hög för att den ska klara av att förstärka signalen så att den avviker från referenssignalen så lite som möjligt. (37)

För att hantera fallet med kvarstående fel krävs en mer avancerad regulator som även har en integrerande del, PI-regulatorn. PI-regulatorn är den regulatorfunktion som är vanligast förekommande inom exempelvis processindustrin. Den integrerande delen innebär att reglerfelet integreras över tid samt multipliceras med en integralförstärkningskonstant,  $K_i$ , enligt ekvation 3. Om reglerfelet  $e(\tau)$  är positivt kommer integralverkan att vara positiv vilket medför en ökad styrsignal som därmed kompenserar för felet. (37)

$$u(t) = K_p \cdot e(t) + K_i \cdot \int e(\tau) d\tau \quad (3)$$

$K_i$  måste vara tillräckligt stor för att felet ska kompenseras men med ett ökat  $K_i$  får systemet försämrade stabilitetsmarginaler på grund av att integralverkan har ett negativt bidrag till kretsöverföringens faskurva. Detta är viktigt att tänka på vid utformningen av regulatorn. (37)

Genom att addera deriveringsverkan till regulatorn fås en PID-regulator, vilket uttrycks i ekvation 4. Med deriveringsverkan blir systemet snabbare samtidigt som det kompenserar bättre för både hög- och lågfrekventa störningar. Den integrerande delen kompenserar för lågfrekventa störningar medan deriveringsverkan kompenserar för de högfrekventa störningarna. Deriveringsverkan har dessutom ett positivt bidrag till kretsöverföringens faskurva vilket gör att de försämrade stabilitetsmarginalerna som blir av integralverkan kan kompenseras av deriveringsverkan. (37)

$$u(t) = K_p \cdot e(t) + K_i \cdot \int e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (4)$$

## 4 Framtagning av robotarm

I detta kapitel behandlas den praktiska tillämpningen av de undersökta teorierna beskrivna i föregående kapitel.

### 4.1 Val av konstruktion

I detta avsnitt beskrivs och motiveras de undersökta respektive de valda koncepten, vilka ligger till grund för den framtagna prototypen.

#### 4.1.1 Robotarmskoncept

Dessa koncept bygger direkt på utvalda robotarmsmekanismer beskrivna i avsnitt 3.1.1. Samtliga robotarmar består av leder med mellanliggande länkar samt har en typ av gripdon fäst på den yttersta länken.

##### Koncept 1: Armbågsrobot

Detta koncept baseras på en artikulerad robotarmsmekanism och innebär att robotarmen har tre frihetsgrader genom tre rotationsleder. Rotationsleder finns i bas, axel och armbåge. Basleden har en vertikal orientering och axelleden är vinkelrät till och genomskär denna. Armbågsleden är parallell med leden i axeln.

##### Koncept 2: Skalär robotarmstyp

Denna robotarmstyp har tre frihetsgrader i form av två rotationsleder och en linjärled. Basen består av en rotationsled vilken via första länken, är parallell med efterföljande rotationsled. I änden av den andra länken finns en linjärled vars orientering är parallell med de två rotationsledningarnas axlar.

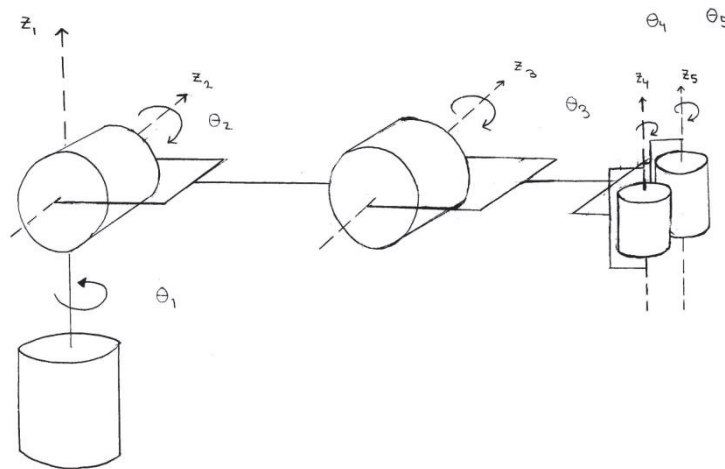
##### Koncept 3: Kartestisk robotarmstyp

Konceptet består av tre linjära leder, vilka möjliggör rörelse i tre frihetsgrader. Samtliga leder är vinkelräta till varandra i sin orientering.

##### Val av koncept

Koncept 1, armbågsrobot valdes eftersom den ger en stor flexibilitet och mångsidighet samt används i de flesta robotarmskonstruktioner. Den ansågs vara ett lämpligt val för konstruktionen av en robotarm då många vanliga problemområden kan undersökas. De resterande två koncepten har olika speciella tillämpningsområden, men är inte lika flexibla och mångsidiga som den artikulerade typen, se Figur 18.





Figur 18 – Armbågsrobot.

#### 4.1.2 Baskoncept

Samtliga baskoncept har ett rotationscentrum som sammanfaller med nästa leds rotationscentrum. Konceptens huvudsakliga skillnad är vilket sorts lager som används i konstruktionen.

##### Koncept 1: Kullager

En axel, centrerad i basens mitt, roterar med stöd av kullager som sitter monterade runt axeln. Till botten av axeln monteras ett drivsystem som roterar axeln. Kullagret är ett smidigt, pålitligt och beprövat lager med mindre friktion än ett glidlager. Kullagret bör rotera flervälet varv åt gången för att underhållas genom smörjning.

##### Koncept 2: Glidlager

Baskoncept 2 bygger likt baskoncept 1 på en axel som är centrerad i basens mitt. Istället för att använda kullager så används glidlager. Som tidigare nämnts så har glidlager högre friktion än kullager, men de är tunna och tar på så vis inte mycket plats och bidrar till en kompakt konstruktion.

##### Koncept 3: Hjul

Baskoncept 3 är lagerlöst och bygger istället på att ett antal hjul används. Hjulen monteras i en cirkel på en platta på vilken även ett drivsystem monteras. Drivmedlets axel monteras mot ytterligare en platta som i sin tur ligger an mot hjulen. Detta gör så att hjulen tar upp det moment som uppstår då robotarmen sträcks ut, ett moment som annars hade belastat servot. Lösningen är enkel att implementera och avsaknaden av lager innebär att smörjning ej krävs.

### Val av koncept

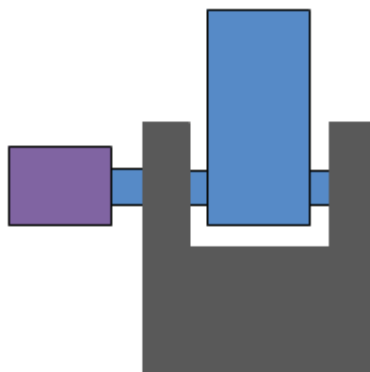
Kullager fungerar optimalt då de får rotera flertalet varv åt gången. På grund av drivsystemval begränsas rotationen till 160 grader, vilket medför en begränsning gällande kullagrets funktion och därmed anses baskoncept 1 som ett olämpligt val. Baskoncept 2 valdes bort på grund av att det ansågs vara för komplicerat och tidskrävande att anpassa ett glidlager efter den utskrivna axeln. På grund av dessa nämnda anledningar ansågs baskoncept 3 vara lämpligast att använda och realisera för basen.

#### 4.1.3 Ledkoncept

Samtliga ledkoncept baseras på en cylindrisk ledmekanism, beskrivet i avsnitt 3.1.2.

##### Koncept 1: Klyka, servo med ensidig axel

Konceptet bygger på att en länk med en genomgående axel monteras i en klyka vilket resulterar i en cylindrisk led, se Figur 19. Delarnas rotationscentrum sammanfaller när axeln placeras i klykan. På axeln i klykan fästs ett drivsystem vilket ger upphov till rörelse. I och med att nästa länk fästs i två symmetriskt formade och placerade fästen fördelas momentet jämnt över klykan.

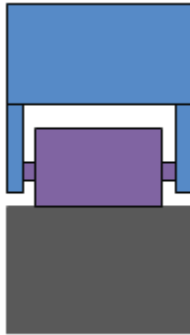


Figur 19 - Koncept 1. Länk och axel i blå, klyka i grå och drivsystem i lila färg.

##### Koncept 2: Klyka, motor med genomgående axel

Konceptet fungerar likt ledkoncept 1, men här fyller motorn den funktion som länken med den genomgående axeln ger. Motorn fästs på en länk och monteras i en klyka på en annan länk. Motors rotationsaxel sammanfaller då med klykans, se Figur 20.

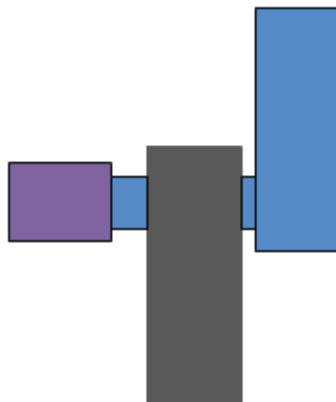
Fördelen med konceptet är att motorn integreras i leden, vilket gör att leden blir kompakt och utsätts därmed för liten belastning eftersom hävarmen är kort.



Figur 20 - Koncept 2. Länk med klyka i blå, drivsystem i lila och länk i grå färg.

### Koncept 3: Parallella armar

Konceptet liknar ledkoncept 1 i och med att drivmedlets hubb monteras på en cylindrisk axel. Axeln genomgår två länkar som tillsammans bildar en gångjärnsled. När leden är monterad blir resultatet två länkar som ligger parallellt mot varandra, se Figur 21. Till skillnad mot ledkoncept 1 blir belastningen på drivsystemet högre i detta koncept i och med att hela momentet fördelas på endast ett fäste.



Figur 21 - Koncept 3: Drivsystem i lila, länk ett i grå och länk två i blå färg.

### Val av koncept

Ledkoncept 1 valdes eftersom fördelen att klykkonstruktionen gör att alla länkar i armen får samma genomgående centrum. I och med att alla leders fästpunkter ligger på en linje kommer också TCP att ligga i linje med fästet i basen, det vill säga i basens koordinatsystem. Detta gör det enkelt att använda den kinematiska konventionen.

Upphovet av friktion från 3D-skrivarens material ansågs vara så pass lågt att det rättfärdigade att plocka bort lagret. Det lagerlösa alternativet gav fördelen att leden kunde dimensioneras och anpassas friare än alternativen att anpassa hela leden efter de

fasta måtten på ett köpt lager eller att anpassa måtten på ett utskrivet lager efter 3D-skrivarens felmargin.

3D-skrivarens felmargin, precision och utskriftskvalitet visade sig dock i slutändan inte leva upp till förväntningarna. Sannolikt bidrar 3D-skrivarens bristfälliga precision till uppkomsten av friktion, men det är ingenting som går att slå fast med säkerhet. Om utrymme funnits för fler iterationer hade dock ett lagrat ledkoncept undersökts och testats noggrannare. De olika lagerkoncepten hade jämförts mot varandra ytterligare och möjligtvis hade ett lagrat ledalternativ visat sig vara ett lämpligare val sett till friktion.

#### 4.1.4 Länkkoncept

Robotens leder hålls samman av mellanliggande länkar och dessa bidrar även till dess räckvidd.

##### **Koncept 1: PVC-rör**

Konceptet bygger på ett cylindriskt PVC-rör som monteras i lederna med hjälp av en genomgående bult.

##### **Koncept 2: Metallstänger**

Konceptet bygger på att fyra gängade metallstänger skruvas in i fyra gängade hål i de två lederna som ska kopplas samman.

##### **Val av koncept**

PVC-rören valdes då de är praktiska eftersom kablarna från servo och sensorer kan dras inuti röret. De är också stabila samtidigt som vikten är relativt låg. Konceptet med stänger har nackdelen att kablarna utsätts för fler risker då de inte går att dölja och skydda. Dessutom har metallstänger hög densitet relativt PVC-röret, vilket gör att den lösningen kan kräva kraftigare drivsystem.

#### 4.1.5 Gripdonskoncept

Gripdonskoncepten baseras på greppningsmekanismerna beskrivna i avsnitt 3.1.3.

##### **Koncept 1: Asynkron cirkulär rörelse med två kontaktpunkter och trycksensor**

Konceptet går ut på att två greppvänliga stöd greppar föremål i en cirkulär rörelse genom att utöva tryck på var sida om detta. Stöden liknas vid två klor som greppar föremålets yta vid två punkter tillräckligt hårt för att kunna lyfta detta. De två stöden drivs asynkront med två drivsystem. På gripdonets greppytan är två trycksensorer placerade.

##### **Koncept 2: Synkron cirkulär rörelse med gummimatta och trådtöjningsgivare**

Koncept 2 kan liknas vid en studsatta och utgörs av två cirkulära ringar med små hål var i en cirkulär gummimatta fästs. De två ringarna styrs av ett drivsystem samt kugghjul vilket

skapar en synkron, cirkulär rörelse där ringarna omsluter föremålet. För att undvika att trycket på föremålet blir för stort används trådtöjningsgivare som kan avgöra hur stor utböjningen är i gummimattorna. Dessa fästs på baksidan av varje gummimatta. En cirkulär utformning av delen som omsluter föremålet prioriteras då föremålet kan greppas från alla vinklar.

### **Koncept 3: Synkron cirkulär rörelse med omslutande grepp och trycksensor**

Konceptet liknas vid två cylindrar vilka omsluter föremålet i en cirkulär rörelse. För att föremålet inte ska gå sönder utrustas cylindrarna med ett mjukt gummimaterial i botten. Under gummimaterialet placeras en trycksensor som känner av trycket. De två cylindrarna styrs synkront med ett drivsystem samt kugghjul.

### **Koncept 4: Asynkron cirkulär translation med trycksensor**

Greppytorna på gripdonet förflyttas i en cirkulär bana samtidigt som de rör sig framåt, parallellt i förhållande till varandra, detta blir alltså en rörelse av typen cirkulär translation. Detta koncept innehåller ett flertal olika leder vilka möjliggör den önskade rörelsen. De två griparmarna styrs asynkront med två drivsystem. Greppytorna är försedda med trycksensorer.

### **Val av koncept**

För att välja ett passande mekaniskt koncept till gripdonet analyserades de presenterade greppningsmekanismerna. Den cirkulära rörelsen i gripdonskoncept 2 och gripdonskoncept 3 anses vara ett olämpligt val då greppytorna på gripdonet inte greppar föremålet parallellt. Detta kan medföra svårigheter vid greppning av runda föremål då det kan hända att föremålet trycks framåt. Problemet uppstår vid både synkron och asynkron styrning. En parallell greppningsmekanism med dess, i många fall, komplicerade kugghjulsmechanism prioriteras inte då fördelarna med den rörelsen kan genereras utifrån enklare greppningsmekanismer. Exempel på en sådan mekanism är cirkulär translation som kan konstrueras utan de avancerade kugghjulen. Med denna enklare mekanism greppas föremålet utan att det trycks framåt. Därför kommer gripdonet att designas utifrån koncept 4 med greppningsmekanismen cirkulär translation.

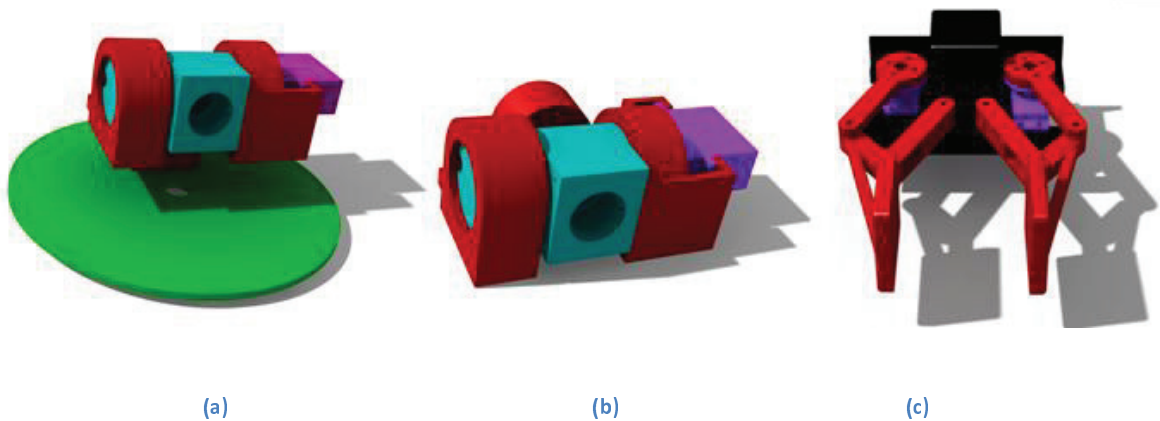
## **4.2 3D-utskrivna komponenter**

För axelled, armbågsled samt gripdon har komponenter skrivits ut med hjälp av en 3D-skrivare.

Axelleden, se Figur 22a, består av 5 stycken utskrivna komponenter och har ett tillhörande drivsystem, lila i Figur 22a. 3 av komponenterna bildar tillsammans en klyka, röd i Figur 22a, och resterande två bildar en led, turkos i Figur 22a.

Armbågsleden, se Figur 22b, är identisk med axelleden förutom att en av komponenterna i klykan, röd i Figur 22b, skiljer sig mot motsvarande komponent i axelleden.

Gripdonet, se Figur 22c, består av 7 stycken utskrivna komponenter och två stycken drivsystem, lila i Figur 22c. Gripdonets basplatta är utskrivna som en hel komponent, svart i Figur 22c, och griparmarna, röda i Figur 22c, består av 3 stycken komponenter vardera. Båda griparmarna består av samma uppsättning komponenter.



Figur 22 - 3D-modeller över axelled (a), armbågsled (b) samt gripdon (c).

Parallellt med att komponenter för detta projekt har skrivits ut så har även komponenter för andra projekt skrivits ut. Detta gör att det ej går att fastställa helt huruvida alla komponenter har skrivits ut i samma material. Troligen består de flesta komponenterna av ABS-plast, men någon kan möjligen bestå av PLA-plast. Den exakta typen av ABS-plast som har använts är oklar, men detta påverkar materialegenskaperna mycket. De termiska egenskaperna påverkas av färgen, det vill säga pigmenttyp och pigmentinblandning. Detta märks något när komponenterna skrivs ut. (38) ABS-plasten har fördelar i att materialet har låg densitet och därmed bidrar dessa komponenter med låg tyngd till den övriga konstruktionen.

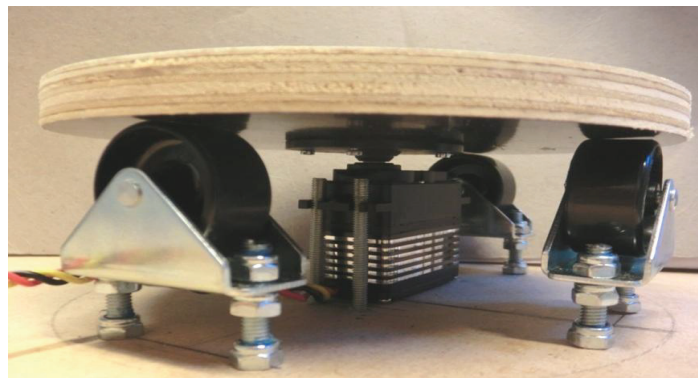
#### 4.3 Val av drivsystem

En linjärmotor som genererar en linjär rörelse, se avsnitt 3.3.3, är inte aktuell eftersom den valda robotarmsmekanismen endast använder sig av rotation. Stegmotorers rörelse tenderar att vara hackig vid låga hastigheter och de saknar återkoppling för en positionsuppfattning, se avsnitt 3.3.1. Prototypens rörelser bör vara jämna och då hastigheterna är relativt låga finns det då en risk att stegmotorer ger hackiga rörelser. För att kinematiken ska fungera på ett bra sätt är det dessutom viktigt att det finns en god positionsuppfattning. För att få en positionsuppfattning är det enklare att implementera ett servo som har inbyggd återkoppling och reglering. Servon är kända för att vara

lämpliga i tillämpningar som kräver hög precision för kraft- och positionsreglering, vilket prototypen kan anses vara, se avsnitt 3.3.2. Digitala versioner av servon är fördelaktiga i jämförelse med analoga då de ger en högre upplösning och ett konstant vridmoment med maxmoment från starten av rörelsen. Alla mekanismer i prototypen drivs på grund av dessa anledningar av digitala servon.

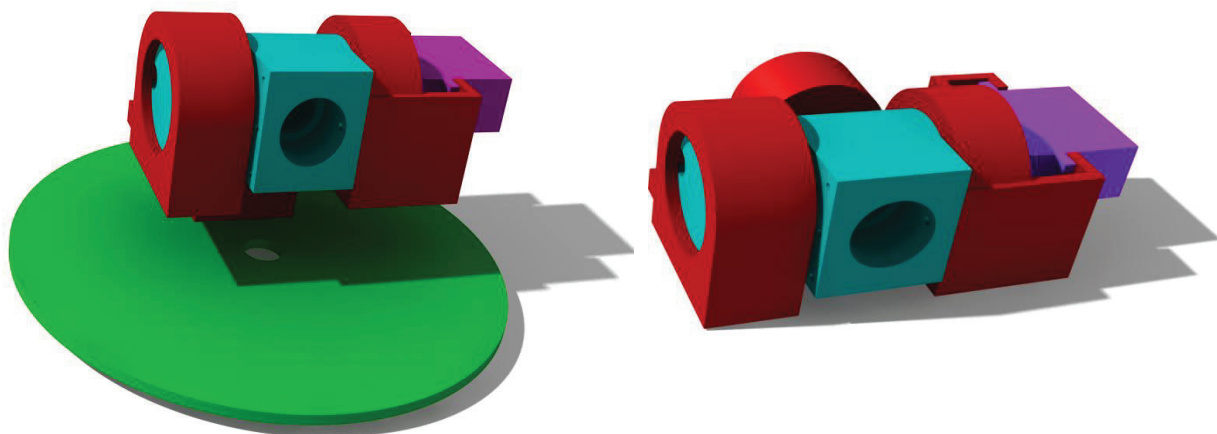
#### 4.4 Framtagen prototyp

Den slutgiltiga robotarmsprototypen, se Figur 27, bygger på en artikulerad robotarmsmekanism, vilken består av tre rotationsleder. Mekanismen ger vanligtvis en 360 graders rotation i basen men begränsas i denna robotprototyp till 160 grader på grund av vald servo. Rotationsvinkeln kan utökas till 180 grader med hjälp av en programmerare. Prototypen kan nå upp till 450 mm då den befinner sig i utsträckt läge.



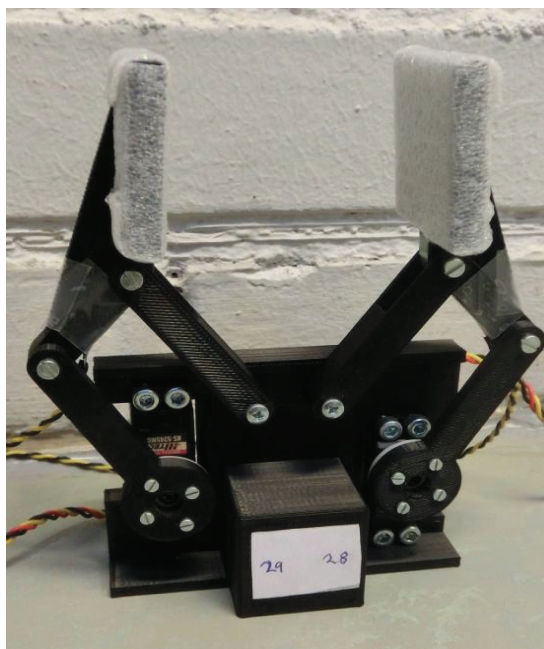
Figur 23 – Basled.

Designen av prototypens roterande bas baseras på baskoncept 3. Resultatet av det realiserade konceptet utgörs av en stor bottenplatta, fyra hjul och en mindre, roterande cirkulär platta av plywood, se Figur 23. Bottenplattan utgörs av en spånskiva som är anpassad för att hålla komponenterna; styrenhet, kretskort, sidecar och nätaggregat. På bottenplattan är hjulen monterade i en cirkel med drivsystemet i centrum. Hjulen är fastsatta på plattan med bultar och låsmuttrar med hjälp av vilka höjden på hjulen kan justeras så att dessa hamnar i höjd med drivsystemet varpå den cirkulära plattan är fäst.



Figur 24 – Axelled till vänster och armbågsled till höger.

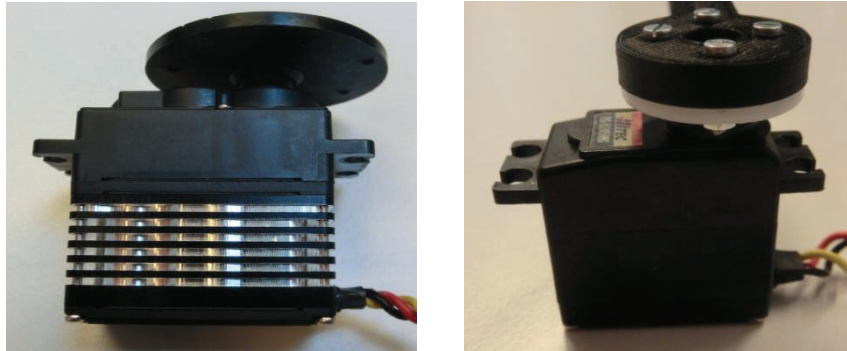
Ledernas design baseras på ledkoncept 1 och består av plastkomponenter som är utskrivna i en 3D-skrivare. Dessa leder är identiska bortsett från att infästet till angränsande komponent skiljer sig på grund av konstruktionen, se Figur 24. Länkarna, som utgörs av PVC-rör, fästs i lederna med hjälp av genomgående bultar. Infästet på leden närmast rotationsleden på basen har en annan konstruktion än armbågsleden, då det inte är någon länk som ska fästas i den ena änden av leden. Denna led fästs på den undre plywoodskivan.



Figur 25 – Till vänster CAD-modell av gripdon och till höger utskrivet gripdon.

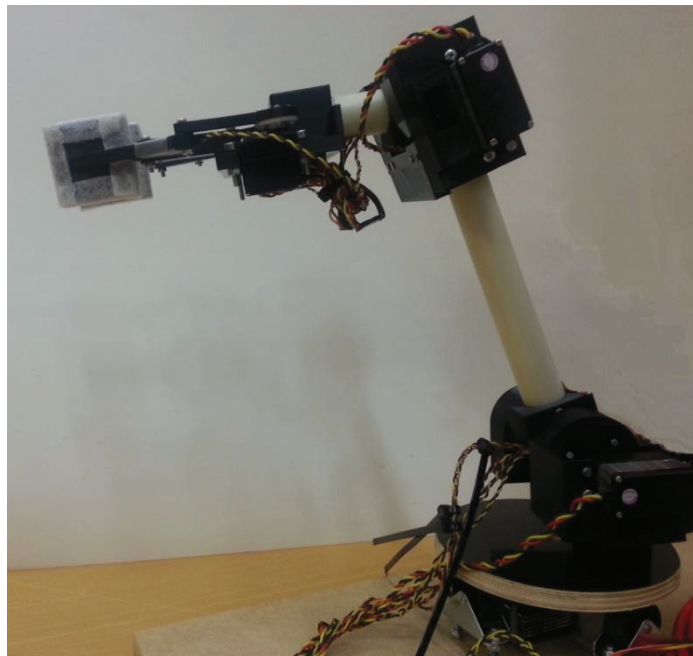


Gripdonet, se Figur 25, är inspirerat av gripdonskoncept 4 och är utskrivet i en 3D-skrivare. Delarna i gripdonet är ihopmonterade med bult och mutter. På greppytorna sitter det två FSR-sensorer fastlimmade. Sensorerna har täckts över med vitt skumgummi för att skyddas.



Figur 26 - Till vänster *Hitec HS-7980TH* och till höger *Hitec HS-5245MG*.

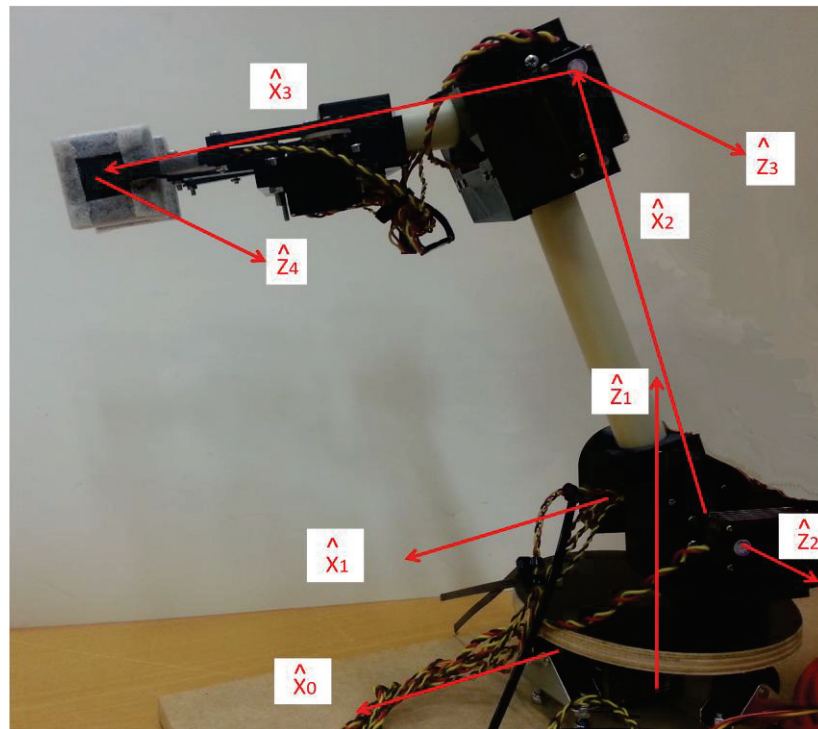
Som drivsystem används två olika typer av servo, en starkare modell, *Hitec HS-7980TH*, i leder och bas samt en något svagare modell, *Hitec HS-5245MG*, i gripdonet, se Figur 26. Mellan de två plattorna i basen sitter ett servo som styr den roterande rörelsen. De två lederna är utrustade med ett servo vardera och gripdonet är utrustat med två servos som styrs asynkront.



Figur 27 - Framtagen prototyp.

#### 4.5 Kinematik

Robotarmens geometri beskrivs med hjälp av Khalil och Dombres konvention, beskriven i avsnitt 3.2, för att kunna relatera robotarmens länkar till varandra. De geometriska parametrarna för en artikulerad robotarm tas fram med hjälp av Figur 28 och redovisas i Tabell 1 nedan. Gripdonet som är fastsatt på änden av andra länken ses här som en led för att längden på andra länken skall kunna tas med som parameter. Ett streck i tabellen innebär att parametrarna ej går att definiera enligt konventionen.

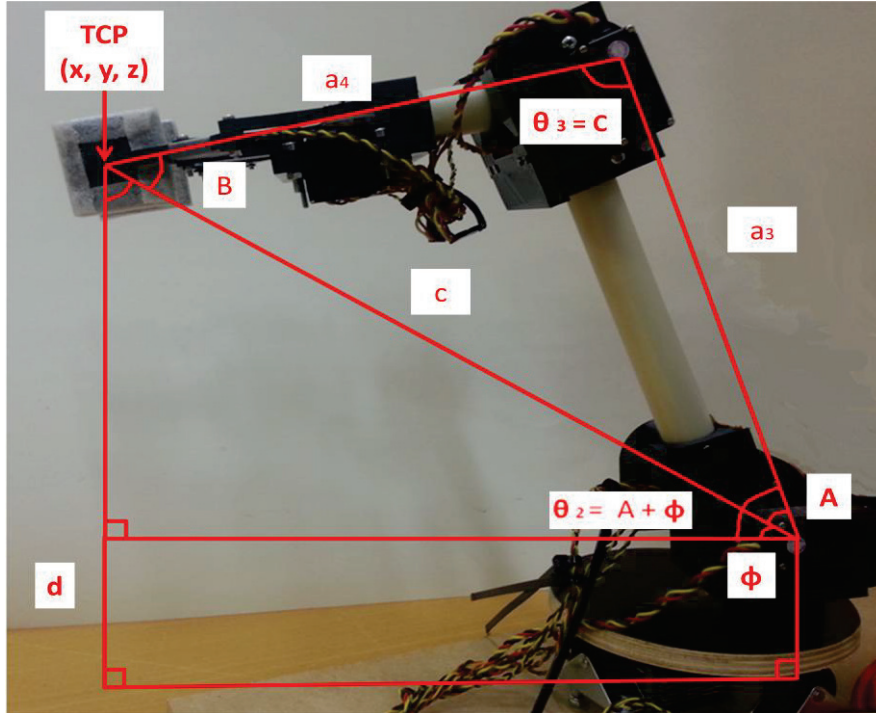


Figur 28 - Khalil och Dombres konvention tillämpad på prototyp.

Tabell 1 - De geometriska parametrarna för en artikulerad robotarm enligt Khalil och Dombres konvention.

i	$\alpha_i$ (rad)	$a_i$ (mm)	$d_i$ (mm)	$\theta_i$ (rad)
1	-	0	73.5	$\theta_1$
2	$-\frac{\pi}{2}$	0	0	$\theta_2$
3	0	225	0	$\theta_3$
4	0	202.6	-	$\theta_4$

Vinklarna för respektive servo tas fram med hjälp av geometriska metoder utifrån Figur 29 och Figur 30 i form av trigonometriska ekvationer, vilka presenteras nedan.



Figur 29 - Geometrisk beskrivning av prototyp. TCP anges utifrån basleden.

Längden från basleden till gripdonets TCP, hypotenusan  $c$ , beräknas enligt ekvation 5.

$$c = \sqrt{x^2 + y^2 + (z - d)^2} \quad (5)$$

Vinkeln mellan de två länkarna uttrycks med hjälp av cosinuslagen, se ekvation 6.

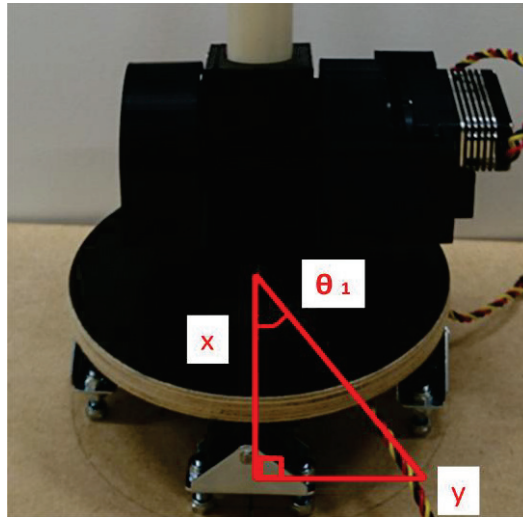
$$\underline{C} = \cos^{-1} \left( \frac{a_4^2 + a_3^2 - c^2}{2a_4a_3} \right) \quad (6)$$

Vinkeln mellan första länken och hypotenusan  $c$  beräknas med hjälp av sinuslagen enligt ekvation 7.

$$A = \sin^{-1} \left( \frac{a_4}{c} \sin \underline{C} \right) \quad (7)$$

Vinkeln  $\varphi$  i den undre triangeln beräknas enligt nedan i ekvation 8.

$$\varphi = \sin^{-1} \left( \frac{z - d}{c} \right) \quad (8)$$



Figur 30 - Geometrisk beskrivning av prototypens basled.

Vinklar för respektive servo i varje led beräknas enligt uttrycken nedan i ekvation 9-11.

$$\theta_1 = \tan \left( \frac{y}{x} \right) \quad (9)$$

$$\theta_2 = A + \varphi \quad (10)$$

$$\theta_3 = \underline{C} \quad (11)$$

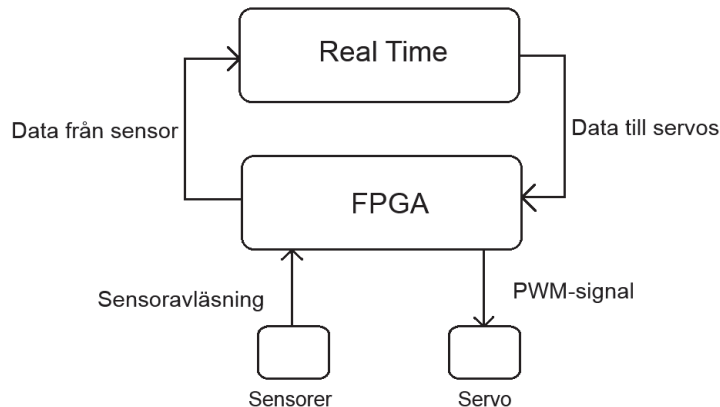
## 4.6 Styrning

Detta avsnitt behandlar programmeringen och reglerdesignen av prototypen.

### 4.6.1 Programstruktur

Robotarmen styrs genom hårdvara och mjukvara från *National Instruments*, beskrivna i avsnitt 3.4.1 respektive avsnitt 3.4.2. Koden för styrning av prototypen är uppdelad i tre stycken VI, två realtids-VI som exekveras på *CompactRIOs* processor och ett VI som kompileras på dess FPGA. VI:en har olika funktioner. I VI:et på FPGA läses informationen

från sensorerna in och PWM-signalerna som styr servona genereras, se Figur 31. De två realtids-VI:en innehåller alla strukturer och beräkningar som krävs för att behandla information från sensorerna samt för att skicka rätt signaler till servona, ett program för armen och ett för gripdonet. I användargränssnittet för realtids-VI:et för armen anges koordinaterna för TCP och i användargränssnittet för gripdonet anges om det ska öppnas eller stängas.



Figur 31 - Schematisk bild av kommunikation.

#### 4.6.2 Styrning av gripdon

Gripdonet kommer att stegas synkront vid stängning tills sensorerna känner av ett föremål. Gripdonet kommer inte att sluta stänga förrän trycket på sensorerna blir såpass högt att föremålet kan gripas. Alltså kommer trycket inte vara tillräckligt högt för att gripdonet ska stanna om den ena sensorn kommer i kontakt med föremålet innan den andra. Så fort detta händer kommer gripdonet att hålla sin position tills användaren anger ett kommando för att släppa föremålet. Gripdonet släpper då föremålet och inväntar nya instruktioner för att gripa ett nytt föremål.

På grund av gränsen för trycket, kommer gripdonet endast att sluta stänga om ett föremål greppas, eller om föremålet är såpass tungt att motstånden från den på den ena sensorn överskrider gränsvärdet.

Spänningen en sensor mäts analogt, och kan beräknas enligt ekvation 12.

$$V_{out} = \frac{R_M * U}{R_M + R_{FSR}} \quad (12)$$

$R_M$  = Resistor som avgör känsligheten av sensorn

$R_{FSR}$  = Resistansen som varierar från trycket

$V_{out}$  = Spänningen som mäts analogt

$U$  = Inmatningsspänning

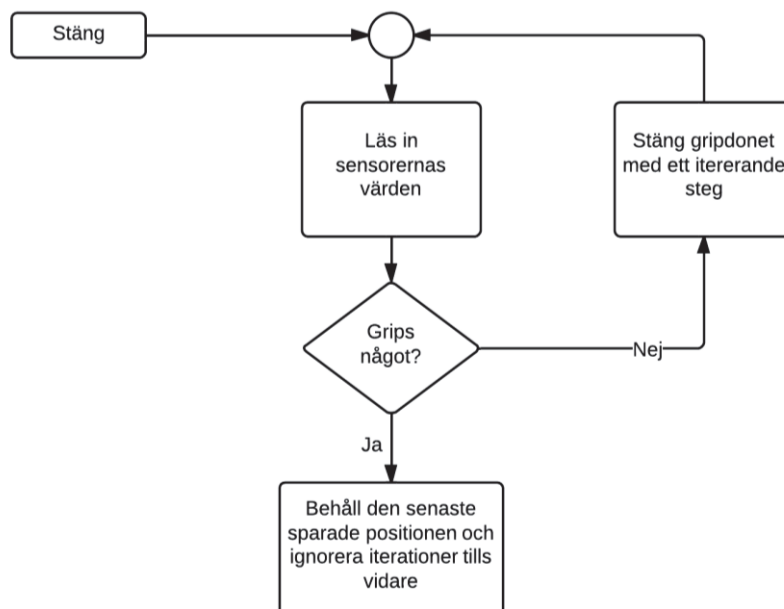
Ett typiskt samband mellan kraft och resistans kan fås från sensorns datablad (32). Om trycket antas motsvara 200 gram på varje sensor, medför detta att resistansen ska motsvara  $R_{FSR} = 4 \text{ k}\Omega$ . I databladet hittades en graf med olika värden för känslighetsresistorn, som visade hur känsligheten varierade för sensorn beroende på denna. En inmatningsspänning på  $U = 9,6 \text{ V}$  i samband med en känslighetsresistor på  $R_M = 100 \text{ k}\Omega$  används, och om dessa värden sätts in i ekvation 12, beräknas utspänningen enligt ekvation 13.

$$V_{out} = 9,23 \text{ V} \quad (13)$$

Ett värde på cirka 23 V blir då lämpligt att ha som gränsvärde för gripdonet, vilket som tidigare sagt mäts in analogt.

Programmet befinner sig i en konstant loop på grund av reglering av gripdonet med hjälp av sensorerna. Programmet har två lägen: öppna och stäng. När gripdonet ska öppna är koden väldigt enkelt, vilket beror på att ingen riktigt återkoppling behövs om en användare bara vill att gripdonet ska öppna.

Då gripdonet ska stänga, finns ett flödesschema över hur koden för detta ser ut i Figur 32, som är en förenkling av den verkliga koden som kan ses i Figur 33. Eftersom programmet befinner sig i en konstant loop så fastnar exekveringen inte i slutet, vilket innebär att programmet helt enkelt körs om när gripdonet griper ett föremål.



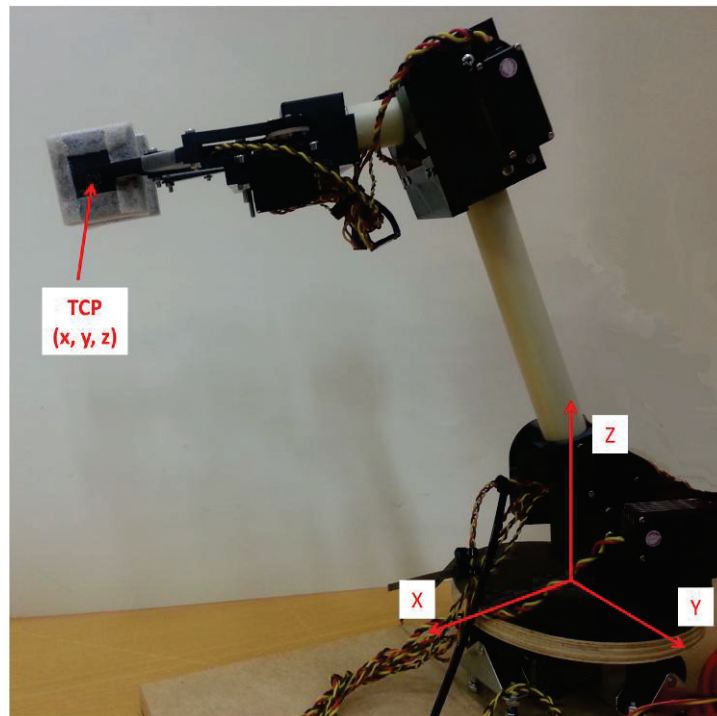
Figur 32 - Flödesschemat för hur gripdonet stänger.

I Figur 33 ses koden för gripdonets stängning. En tydlig återkoppling syns ej. Detta beror på att återkoppling för villkoren sker med hjälp av variabler (de gröna rutorna sensor 1 Stop samt sensor 2 Stop). För villkoren används det tidigare uträknade värdet som avrundades till 9. Tydligheten i figuren däremot är att placeringen av while och for-looparna tydligt visas, samt var dessa är placerade i förhållande till öppna eller stäng kommandot som representeras av en *case-structure*.



#### 4.6.3 Styrning av arm

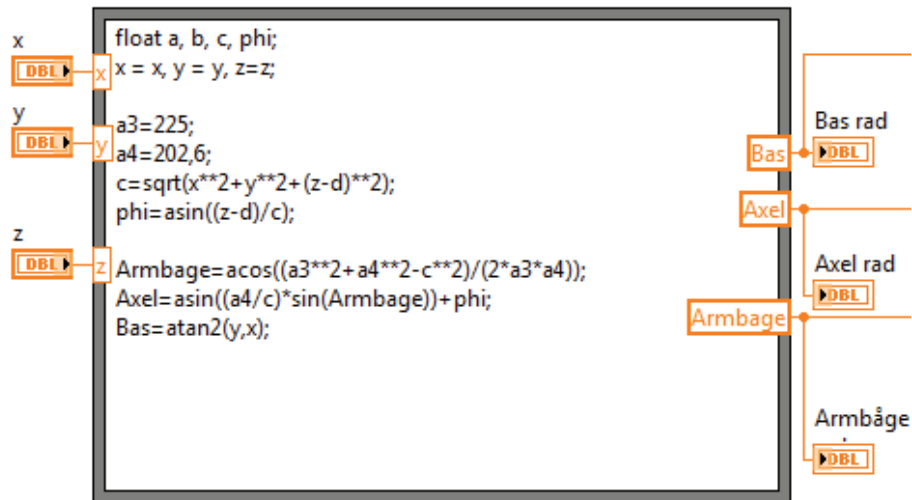
Koden som styr armen återfinns i realtidsprogrammet för robotarmen. Robotarmen styrs genom att användaren anger TCP i millimeter utifrån robotarmens bas genom att dra i reglagen för x, y och z i användargränssnittet, se Figur 34.



Figur 34 - Prototyp med koordinatsystem och TCP.

I koden finns ett block som utvärderar matematiska uttryck, en så kallad *Formula node*, se Figur 35. Koordinaten angiven av användaren är indata och i detta block används invers kinematik för att beräkna vinkeln, alltså positionen för respektive servo, i radianer. Dessa ekvationer härleds och presenteras i avsnitt 4.5.





Figur 35 - Formula node med invers kinematik vilken beräknar vinklar för respektive servo.

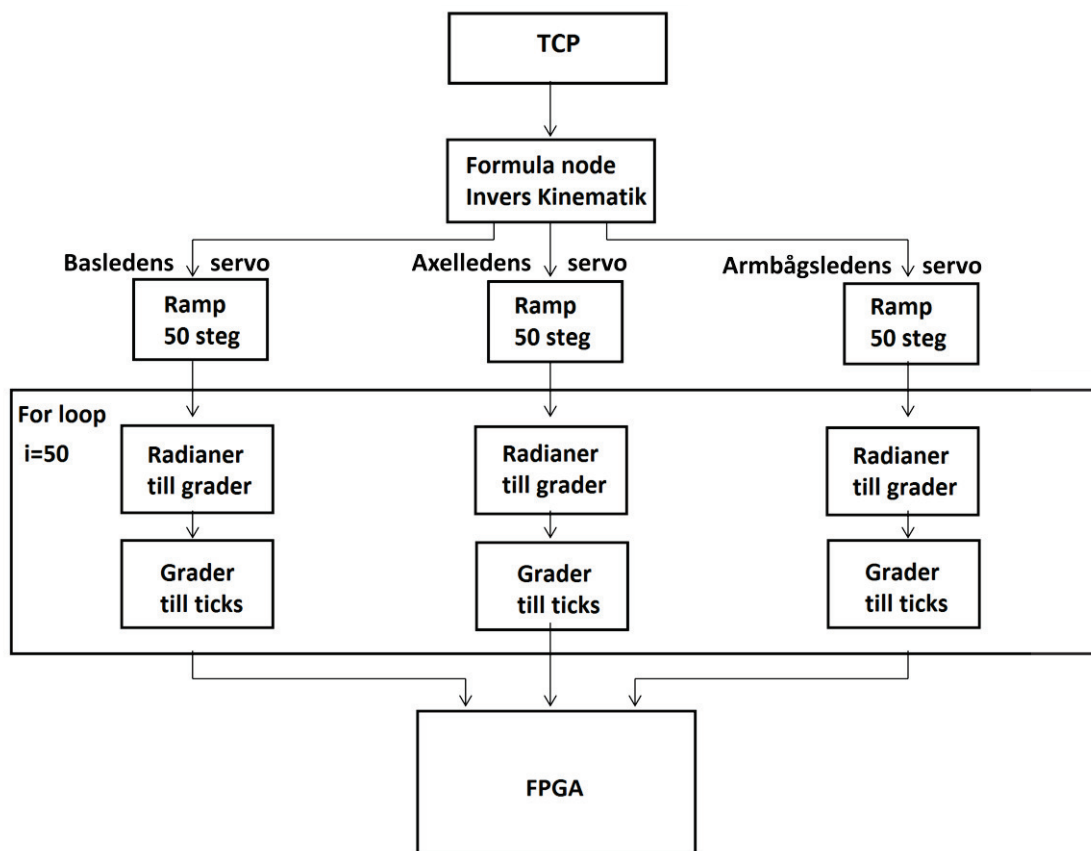
Utdatan behandlas sedan i en ramp, beskrivet i avsnitt 4.6.4, där steget mellan den nuvarande vinkeln och den beräknade delas upp i 50 steg. Varje steg omvandlas till grader med hjälp av ekvation 14.

$$Vinkel \text{ i grader} = \frac{180}{\pi} * (Vinkel \text{ i radianer}) \quad (14)$$

Vidare omvandlas graderna till ticks med hjälp av ekvation 15. Sambandet mellan grader och ticks är linjärt och har tagits fram genom att mäta servots grader i förhållande till givet ticks. 30 000 ticks innebär att servots vinkel är 0° och 88 000 ticks att den är 160°. Ticksen skickas sedan till FPGA för att där generera PWM-signaler, se avsnitt 4.6.5, vilka skickas till respektive servo.

$$Ticks = (Vinkel \text{ i grader}) * 363 + 30\,000 \quad (15)$$

Omvandlingen av vinkeln från radianer till grader samt omvandlingen av vinkeln i grader till ticks återfinns i en for loop i programmet. Loopens funktion är att skicka ett steg åt gången var 100 millisekund till FPGA. En schematisk bild av koden som styr armen kan ses i Figur 36.



Figur 36 - Schematisk bild av styrning av arm.

I Tabell 2 redovisas exempelkörningar av programmet för olika TCP med respektive vinkel och motsvarande tick för lederna.

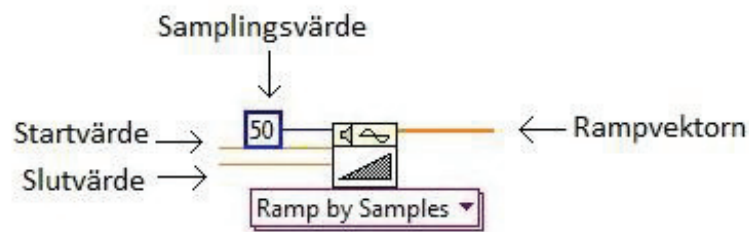
Tabell 2 - Exempelkörning av programmet med olika TCP angivet i millimeter med respektive vinklar i grader och motsvarande ticks för lederna.

TCP (x, y, z) i [mm]	Bas	Axel	Armbåge
(126, 172, 37.5)	$53.7751^{\circ} = 49520$ ticks	$45.0396^{\circ} = 46349$ ticks	$60.4845^{\circ} = 51956$ ticks
(150, 150, 50)	$45.0000^{\circ} = 46335$ ticks	$48.6478^{\circ} = 47659$ ticks	$59.6114^{\circ} = 51639$ ticks
(100, 100, 100)	$45.0000^{\circ} = 46335$ ticks	$72.7299^{\circ} = 56401$ ticks	$38.8829^{\circ} = 44114$ ticks

#### 4.6.4 Reglering av servorörelse

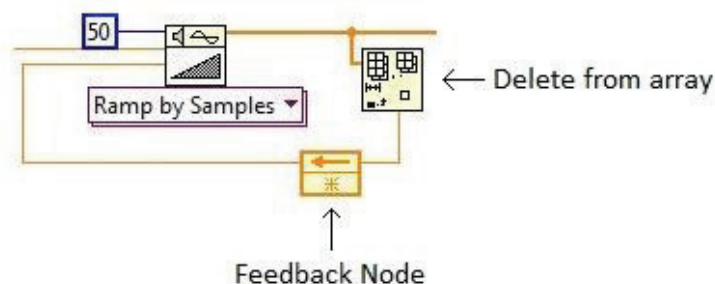
I och med att servona har inbyggda regulatorer har ingen extern reglering av position implementerats, men däremot styrs servona med en styrsignal som följer en rörelseprofil vilket gör så att hastigheten på robotens rörelser kontrolleras. Rörelseprofilen som används är i form av en ramp som delar upp vinkelskillnaden mellan den önskade positionen och den nuvarande positionen i mindre steg. I *LabVIEW* görs detta med en

funktion som heter *Ramp Pattern*, som kan ses i Figur 37. I prototypens fall används rampblocket i realtidsprogrammet för att dela upp vinkelförändringen i 50 steg som ett i taget skickas till FPGA-programmet där PWM-signalen till servona genereras. Till rampblockets ingångar skrivs ett startvärde (den nuvarande positionen), ett slutvärde (den önskade positionen) och ett samplingsvärde som säger hur många steg intervallet ska delas upp i. I programmet är samplingsvärdet satt till 50 steg. Som utsignal ger blocket ut en vektor som innehåller de steg som servot ska ta.



Figur 37 - Visar rampfunktionen med dess insignaler: start- slut- och samplingsvärde, samt vektorn som rampen ger som utsignal.

Rampens startvärde utgörs som tidigare nämnt av servots nuvarande position vilket kräver återkoppling, alltså att programmet håller reda på den senaste signalen som skickats till servot. I *LabVIEW* finns ett funktionsblock som heter *Delete from array* som används för att spara just den signalen. Det som funktionen gör är att den hämtar det sista elementet i vektorn som rampen gav ut, det vill säga målpositionen för servot och sparar det i en återkopplingsfunktion, *Feedback Node*, som sparar värdet till nästa iteration. Vid nästa iteration, det vill säga när en ny koordinat för TCP har angetts, kommer återkopplingsfunktionen att ange det gamla slutvärdet som startvärde för nästa ramp. Hur återkopplingen går till visas i Figur 38.



Figur 38 - Figuren visar de två blocken som krävs för att återkoppla en iterations slutvärde till nästa iterations startvärde.

Utifrån rampblockets indata returnerar rampblocket en vektor som innehåller 50 värden. För att alla värden i vektorn ska skickas till FPGA-programmet ett i taget krävs det att

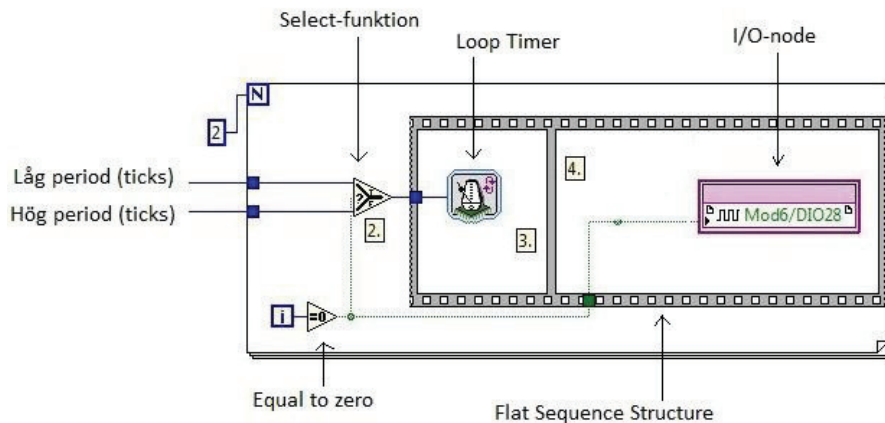
vektorn stegas igenom från början till slut vilket görs med en for-loop där varje iteration skickar ett värde till FPGA-programmet. I for-loopen finns en timer som bestämmer tiden mellan varje iteration, som i detta fall är satt till 100 ms. Anledningen till att det krävs en timer är för att signalerna inte får skickas med mindre intervall än PWM-signalens periodtid, det vill säga summan av den låga och den höga periodens tid. Om signalerna hade skickats utan tidsfördröjning hade robotens rörelser blivit hackiga, då alla steg i förflyttningen inte kommer att komma med i PWM-signalen.

#### 4.6.5 Generering av PWM-signal

I FPGA-programmet skapas den PWM-signal som styr ett servo. De båda servotyperna i roboten styrs av samma typ av PWM-signal; en signal med låga perioder som har fast längd och höga perioder vars längd varierar. Längden på den höga perioden bestämmer hur lång förflyttning servot gör. Den låga perioden är kortare för gripdonets servon, 150 000 ticks vilket motsvarar 3,75 millisekunder enligt ekvation 16, medan armens servon har längre låg period, 250 000 ticks vilket motsvarar 6,25 millisekunder enligt samma ekvation. I och med att dessa värden är fasta anges de i FPGA-programmet som konstanter medan den höga perioden läses in som insignal till programmet. Givet längden på de låga perioderna så tillåts den höga perioden att variera i längd mellan 30 000 och 80 000 ticks för de båda servotyperna.

$$Tid\ i\ sekunder = \frac{Ticks}{Frekvens\ i\ hertz} \quad (16)$$

FPGA skickar och tar emot signaler med en frekvens på 40 MHz. Med denna frekvens motsvarar ett tick 0,25 nanosekunder (39). Som nämdes i avsnitt 0 skickas ett värde från rampvektorn i armens realtidsprogram till FPGA-programmet var 100 millisekund. Det värdet motsvarar längden på den höga perioden i PWM-signalen. Utifrån konstanten och insignalen, det vill säga den låga och den höga perioden, skapas PWM-signalen med hjälp av LabVIEW:s struktur *Flat Sequence Structure* och ett antal funktionsblock som kan ses i Figur 39.



Figur 39 - LabVIEW-kod som genererar en PWM-signal.

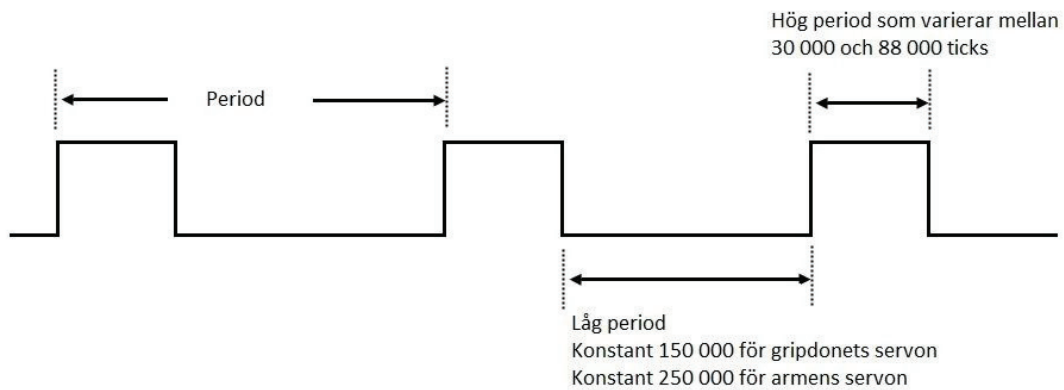
Det block som benämns *I/O-node* i Figur 39 är en nod som är kopplad till I/O-modulen NI 9403 som sitter i *CompactRIO*. Det är den modulen som läser av noden och skickar PWM-signalen till Sidecar där servona är inkopplade. Till denna nod skrivs antingen hög eller låg signal beroende på om for-loopen är i sin första eller andra iteration. För att avgöra vilken iteration som exekveras används funktionen *Equal to zero* som testar om index *i* är lika med noll eller inte. Vid första iterationen kommer index *i* att vara lika med noll vilket innebär att blocket *Equal to zero* skickar ut en hög signal till noden. Vid andra iterationen kommer index *i* att vara lika med ett, vilket innebär att blocket skickar en låg signal till noden.

Hur länge signalen till noden ska vara hög eller låg avgörs av två funktioner: *Loop Timer* och *Select* som båda är utmarkerade i Figur 39. *Select*-blocket tar tre insignaler: antal ticks för hög period, antalet ticks för låg period och en boolesk insignal som avgör vilket ticks-antal som ska skickas som utsignal. Här skickas resultatet från *Equal to Zero*-testet till den booleska ingången. Om denna är sann kommer blocket att returnera antalet ticks för en hög period och om den är falsk returneras antalet ticks för en låg period.

Det antal ticks som *Select*-blocket returnerar blir insignal till *Loop Timer*-blocket, som bestämmer hur länge for-loopen ska exekvera, det vill säga den bestämmer hur lång varje iteration ska vara och därmed hur länge hög respektive låg signal skrivs till modulen, baserat på det antalet ticks som ges som insignal. Det innebär alltså att varannan iteration ger låg signal, som varar i 150 000 respektive 250 000 ticks (alltså 2,75 respektive 6,25 millisekunder), och varannan iteration ger hög signal vars längd bestäms av det beräknade antalet ticks för hög signal som skickats från realtidprogrammet.

I och med att *Loop Timer* och *I/O-node* är placerade i en *Flat Sequence Structure* kommer signalen att skickas till noden under exakt så lång tid som *Loop Timer* bestämmer.

Resultatet av denna kodsnuitt blir en PWM-signal med varierande frekvens, det vill säga olika långa perioder. Signalen har konstant låg period och varierande hög signal, vilket ses i Figur 40.



Figur 40 - PWM-signal som presenterar cykeltiden samt den varierande höga perioden och den konstanta låga perioden.

Figur 40 är summan av tiden för en hög och en låg period, och det är denna tid som begränsar hur ofta ett nytt ticksantal får skickas från rampen i realtidsprogrammet. Den längsta periodtiden som kan uppstå är alltså då hög signal varar under 88 000 ticks, vilket motsvarar 2 millisekunder och ger då en periodtid på 5,75 millisekunder för gripdonets servon och på 8,25 millisekunder för armens. I och med att det i dagsläget är 100 millisekunder mellan varje iteration kommer rampformen att hållas i och med att varje värde i rampen kommer med i PWM-signalen. PWM-signalen skickas från modulen till *Sidecar* som har en spänningsmatning på 6 V. Detta medför att om en PWM-signal till armens servo skickas med exempelvis 50 000 ticks på den höga perioden så blir duty cycle:

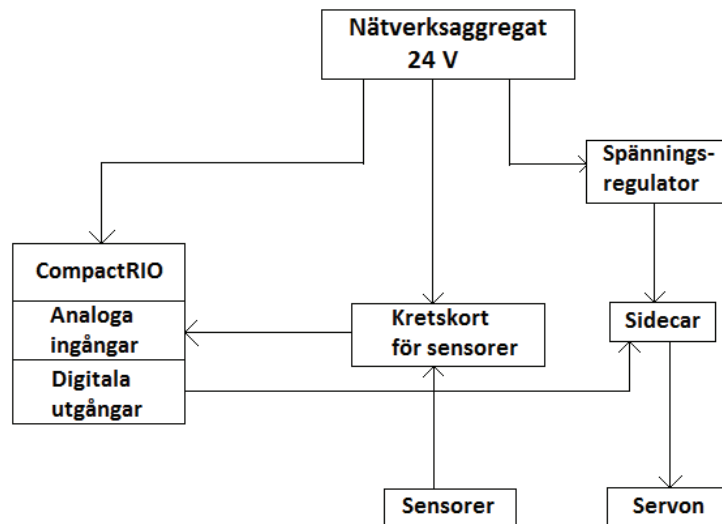
$$\frac{50000}{50000 + 250000} = 0,1666 = 16,7 \%$$

Om duty cycle är 16,7 % kommer servot att matas med:

$$Duty Cycle * Spänning = 0,167 * 6 = 1 V$$

## 4.7 Elektronik

Det finns tre delar i prototypen vilka kräver ström: *CompactRIO*, den digitala sidecar och ett kretskort för sensorerna. En schematisk bild av hur dessa delar kopplas visas i Figur 41. Alla delarna drivs av ett nätverksaggregat vilket ger ut 24 V.



Figur 41 - Schematisk bild av elektroniken.

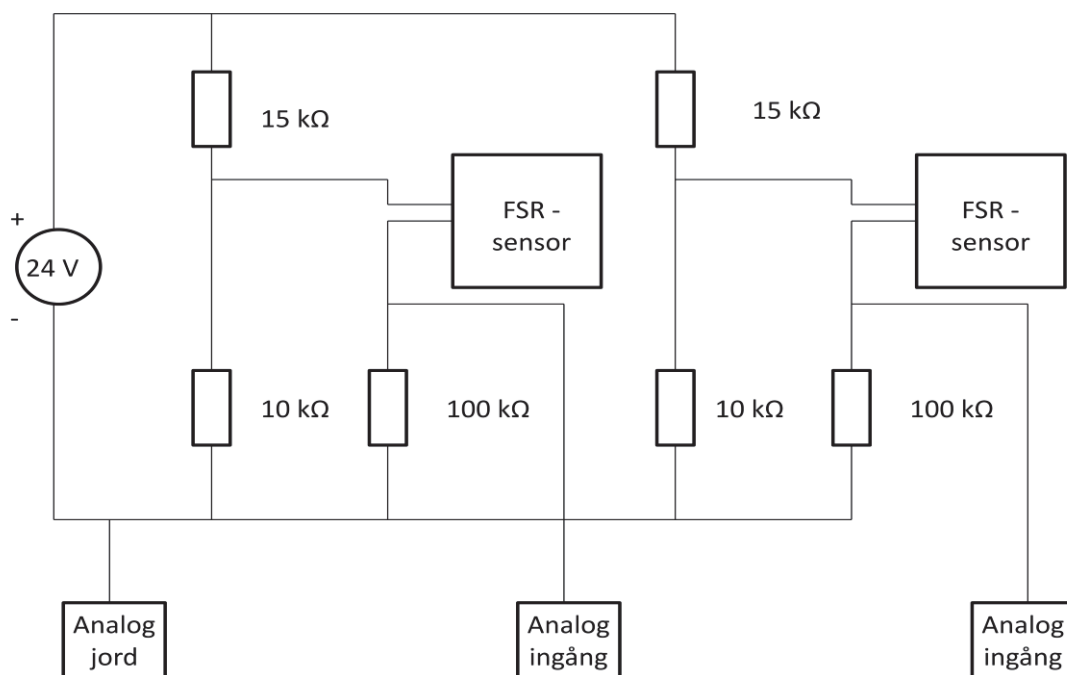
*CompactRIO* kräver 24 V och därav kopplas den direkt till nätverksaggregatet. I prototypens fall driver modulerna inte servona eller sensorerna utan de endast skickar och tar emot signaler.

Sidecar kräver 12 V och därför regleras spänningen från nätverksaggregatet ner till rätt nivå med hjälp av en spänningsregulator. Det är den digitala sidecar som driver servona i gripdon och arm, se Figur 41, via dess PWM-utgångar. PWM-utgångarna ger ut en spänning på 6 V och en maxström på 3 A till varje servo (28). PWM-signalerna genereras i FPGA-programmet och skickas sedan ut till PWM-utgångarna på sidecar via den digitala modulen NI 9403, se Figur 42.



Figur 42 - Sidecar kopplad till den digitala modulen NI 9403.

Till kretskortet är sensorerna kopplade, se Figur 43 för en schematisk bild. Kretskortet är tillverkat på ett sådant sätt att spänningen över seriekopplingen av FSR-sensorerna och mättningsresistorn,  $100\text{ k}\Omega$  i Figur 43, är  $9,6\text{ V}$ . Resistansen i denna seriekoppling är hög i förhållande till den parallellkopplade  $10\text{ k}\Omega$ -resistansen och därför går det knappt någon ström igenom FSR-sensorn. En liten ström genom sensorn är fördelaktigt då den är känslig för stora strömmar (28).



Figur 43 - Schematisk bild över kretskort samt kopplingar till den analoga modulen.



Då FSR-sensors resistans varierar, varierar även spänningen över mätresistorn. Spänningen över mätresistorerna mäts med hjälp av den analoga modulen NI 9201. Jord från kretskortet kopplas till den analoga modulens ingång för jord och kablarna för respektive sensor och mätresistor kopplas till två olika analoga ingångar, se Figur 44. Modulen läser sedan av spänningen mellan jord och respektive analoga ingång för att programmet ska kunna avgöra vilken last respektive sensor utsätts för.



Figur 44 - Analoga modulen NI 9201 med jord och kablarna för respektive sensor inkopplade.

#### 4.8 Dimensionering av drivsystem

Det servo som kommer att utsättas för störst belastning är det som höjer och sänker armen i basen, alltså det i robotens axel. Detta dimensioneras därför för att kunna driva vid extremfallet när robotarmen är helt utsträckt, det vill säga när lastmomentet är som störst. På grund av robotarmens komplexa geometri kan dess tröghetsmoment vara komplicerat att beräkna. Givet komplexiteten antogs ideala geometrier vid beräkningen, och därav infördes en felmarginal för att minimera risken för underdimensionering.

Under dimensioneringsprocessen är materialval, robotarmens volym och länklängder av intresse. Dessa faktorer påverkar robotarmens totala massa och dess hävarm, och därmed det totala moment som drivmedlet belastas med. Beräkningarna utfördes genom att från CAD-mjukvaran importera värden för komponenternas area, volym och massa, till beräkningsmjukvaran *MATLAB*.

Momentet som drivmedlet behöver klarar av delas upp i två olika delar under beräkningsprocessen; jämviktsmoment och det som tröghetsmoment bidrar till.

##### Jämvikt

Momentet för att drivmedlet ska klara av att hålla lasten i jämvikt räknas ut genom att multiplicera massan av en komponent med gravitationsaccelerationen och avståndet från dess tyngdpunkt till rotationscentrum för momentet. Eftersom flera olika komponenter

finns med, kommer dessa att summeras. Det totala jämviktsmomentet kan beskrivas enligt ekvation 17.

$$M_j = \sum m \cdot g \cdot d \quad (17)$$

$M_j$  = Momentjämvikt

$m$  = Massa

$g$  = gravitationskraften,  $9,82 \text{ m/s}^2$

$d$  = Avståndet mellan momentpunkten och tyngdpunkten för föremålet

### Tröghetsmoment

Hela armen antas ha den ideala formen av en stång, och antas befinna sig i extrempunktsläget då den är fullt utsträckt. Stångformen innebär att tröghetsmomentet, beräknas enligt ekvation 18.

$$I = \frac{m \cdot l^2}{3} \quad (18)$$

$I$  = Masströghetsmoment

$m$  = Massa

$l$  = Längd

Tröghetsmomentet har inte samma enhet som moment, vilket är varför värdet får multipliceras med vinkelfrekvensaccelerationen och beskrivs enligt ekvation 19.

$$M_I = I \cdot \dot{\omega} \quad (19)$$

$M_I$  = Moment från tröghetsmoment

$I$  = Tröghetsmoment

$\dot{\omega}$  = Vinkelfrekvensacceleration

Vinkelfrekvensaccelerationen beräknas genom att dela accelerationen med avståndet till tyngdpunkten av föremålet. På grund av ett flertal komponenter, beräknas en gemensam tyngdpunkt av dessa med hjälp av Steiners sats enligt ekvation 20.

$$tp_{tot} = \sum \frac{A_i \cdot tp_i}{A_{tot}} \quad (20)$$

$tp_{tot}$  = Den gemensamma tyngdpunkten

$A_{tot}$  = Areal för alla komponenter

$tp_i$  = tyngdpunkten för en komponent

$A_i$  = Area för en komponent

Med hjälp av den gemensamma tyngdpunkten, beräknas vinkelfrekvensaccelerationen enligt ekvation 21.

$$\dot{\omega} = \frac{acc_{tot}}{tp_{tot}} \quad (21)$$

$acc_{tot}$  = Den totala accelerationen som består av gravitationsaccelerationen och ytterligare en önskad acceleration

Efter att ha räknat ut jämviktsmomentet och bidrag från tröghetsmomentet, adderas dessa enligt ekvation 22.

$$M_{tot} = M_j + M_I \quad (22)$$

$M_{tot}$  = Totalt moment

$M_j$  = Momentjämvikt

$M_I$  = Moment från tröghetsmoment

Alltså fås det totala momentet som drivmedlet behöver klara av genom resultatet från ekvation 22.

Tabell 3 - Värden för delar i konstruktionen, samt hur långt deras tyngdpunkt ligger från rotationspunkten.

Delar i konstruktion	Massa [kg]	Avstånd mellan rotationspunkt och tyngdpunkt [m]
Länk	0,074	0,150
Led	0,177	0,225
Gripdon	0,280	0,375
Föremål	0,070	0,450

Genom att sätta in alla värden från Tabell 3 in i ekvation 17 fås jämviktsmomentet, vilket blir:

$$M_j = \sum m \cdot g \cdot d = 1,842 \text{ [Nm]} \quad (23)$$

Vid uträkning av tröghetsmomentet används den totala massan i samband med längden på armen. Dessa sätts in i ekvation 18, och ger:

$$I = \frac{m_{tot} \cdot l^2}{3} = 0,041 \text{ [kgm}^2\text{]} \quad (24)$$

För att räkna ut momentet av tröghetsmomentet, behöver först ekvation 21 räknas ut, som i sin tur beror av ekvation 20. Tabell 4 visar arean för alla olika komponenter som

drivsystemet kommer att belastas utav. Dessa värden är beräknade som värsta fall, för att säkerställa att det dimensionerade drivsystemet inte blir för svagt.

Tabell 4 - Värden för arean av alla olika komponenter.

Delar i konstruktion	Area [m <sup>2</sup> ]
Länk	0.0072
Led	0.0070
Gripdon	0.0580
Föremål	0.0023

Genom en insättning av alla värden från Tabell 4 in i ekvation 20, fås den gemensamma tyngdpunkten som:

$$tp_{tot} = \sum \frac{A_i * tp_i}{A_{tot}} = 0,342 \text{ [m]} \quad (25)$$

Drivsystemet antas bidra med en acceleration på  $3 \text{ m/s}^2$ , vilket ger en total acceleration på:

$$acc_{tot} = g + a = 9,82 + 3 = 12,82 \text{ [m/s}^2\text{]} \quad (26)$$

Genom att sätta in värden från ekvation 25 och 26 in i ekvation 21, fås vinkelfrekvensaccelerationen enligt:

$$\dot{\omega} = \frac{acc_{tot}}{tp_{tot}} = 37,543 \text{ [rad/s}^2\text{]} \quad (27)$$

De nödvändiga beräkningar som krävs för att ta reda på momentet som tröghetsmomentet bidrar med har nu utförts. Genom en insättning av ekvation 24 och 27 in i ekvation 19, fås momentet som:

$$M_I = I \cdot \dot{\omega} = 1,525 \text{ [Nm]} \quad (28)$$

Det totala momentet räknas ut genom en insättning av ekvation 23 och 28 in i ekvation 22, vilket ger:

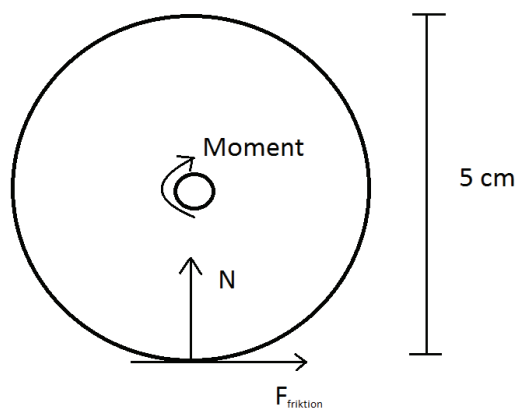
$$M_{tot1} = M_j + M_I = 3,367 \text{ [Nm]} \quad (29)$$

Alltså behöver ett drivsystem klara av ett moment på ungefär 3,37 Nm för att kunna lyfta hela robotarmen. Beräkningen utfördes med antaganden om värsta fall, vilket innebär att ett drivsystem som klarar av ett lite mindre moment också hade fungerat. Dessa beräkningar tar däremot inte hänsyn till friktionen som uppstår i axelleden. Om radien på leden antas vara 2,5 cm, skulle detta innebära att friktionskraften skulle befinna sig 2,5 cm från momentpunkten. Figur 45 visar hur krafterna och momentet påverkar leden. Friktionskraften bidrar med ett moment som kan beräknas enligt ekvation 30.

$$M_{friktion} = 0,025 * F_{friktion} \quad (30)$$

Ekvationen för att räkna ut kraften från friktionen kan räknas ut enligt ekvation 31.

$$F_{friktion} = \mu * N \quad (31)$$



Figur 45 - Hur leden för drivsystemet antas se ut.

För att räkna ut friktionen i leden räknas först normalkraften ut enligt ekvation 32.

$$N = \frac{M_{tot}}{distans} = 134,684 \text{ [N]} \quad (32)$$

Distans = 2,5 cm (radien för leden)

Friktionskoefficienten i lederna antas vara 0.1, vilket. Friktionskraften fås så genom en insättning av ekvation 32 in i ekvation 31, vilket ger:

$$F_{friktion} = 13,468 \text{ [N]} \quad (33)$$

Insättning av ekvation 31 i ekvation 30 ger momentet som friktionen bidrar med, vilket blir:

$$M_{frikction} = 0,337 \text{ [Nm]} \quad (34)$$

Momentet från friktionen kan anses ha en påverkan i extremläget, men på grund av den låga friktionskoefficienten ökar detta inte kraven på drivsystemet särskilt mycket. Det nya momentet för drivsystemet räknas nu enligt:

$$M_{tot} = M_{tot1} + M_{frikction} \quad (35)$$

En insättning av ekvation 29 och 34 in i ekvation 35 ger då det totala momentet på:

$$M_{tot} = 3,704 \text{ [Nm]} \quad (36)$$

Momentet från ekvation 36 är vad drivsystemet måste klara av vid ett icke-ideellt fall. Något som ytterligare kan förändrar momentet däremot är accelerationen. Om till exempel accelerationskraften hade fördubblats från  $3 \text{ m/s}^2$  till  $6 \text{ m/s}^2$ , hade detta medfört en ökning från cirka 3,7 Nm till cirka 4,1 Nm. En acceleration på  $3 \text{ m/s}^2$  som ligger utöver gravitationsaccelerationen anses vara tillräckligt hög för att täcka in en eventuell marginal som kan behövas.

Två val av drivsystem hittades, och ansågs vara lämpliga för robotarmen. En variant som enligt specifikation orkar med en last på cirka 4,3 Nm och en dyrare variant som orkar med en last på cirka 8 Nm. Skillnaden mellan dessa är att styrningen för den svagare servon är något enklare än den kraftigare. På grund av förenklad styrning, och att kravet på servot är 4,1 Nm så valdes den svagare varianten.

## 5 Resultat

Detta kapitel presenterar de utmaningar som stötts på under framtagandet av prototypen.

### 5.1 Konstruktion

Att skriva ut komponenter med hjälp av en 3D-skrivare är en smidig och relativt snabb metod att använda för att realisera en produkt. Genom att rita upp modeller med hjälp av CAD-mjukvara och sedan skriva ut dessa som fysiska komponenter så kan koncept och idéer smidigt testas.

3D-skrivaren som har använts för att skriva ut komponenter med har visat sig ha brister gällande noggrannhet och precision. Realiseringen från CAD-modell till fysisk 3D-komponent medförde dimensionella skillnader just på grund av 3D-skrivarens parametrar och felmarginal samt materialets egenskaper.

3D-skrivarens felmarginal uppmärksammades, av naturliga skäl, tydligast där kravet på dimensionell precision var som högst. Axelledens och armbåglens cirkulära delar, där rotation skulle uppstå, modellerades och dimensionerades initialt i CAD-programvaran för att passa varandra optimalt. Ledernas diameter och hålen på klykornas diameter dimensionerades enligt samma mått för att uppnå rotation i leden utan glapp. 3D-skrivarens felmarginal tillsammans med materialets egenskaper gjorde dock att de utskrivna komponenterna inte passade ihop. Axelns diameter var för stor för att passa i klykan. Utskrifterna hade även tappat form då de delar som i CAD-modellen var runda snarare var något ovala i utskriven form.

I och med att de utskrivna komponenternas mått inte stämde överens med CAD-modellernas så krävdes ett antal iterationer för att anpassa CAD-modellerna efter 3D-skrivarens felmarginal, för att på så sätt uppnå önskad dimensionering av de 3D-utskrivna komponenterna.

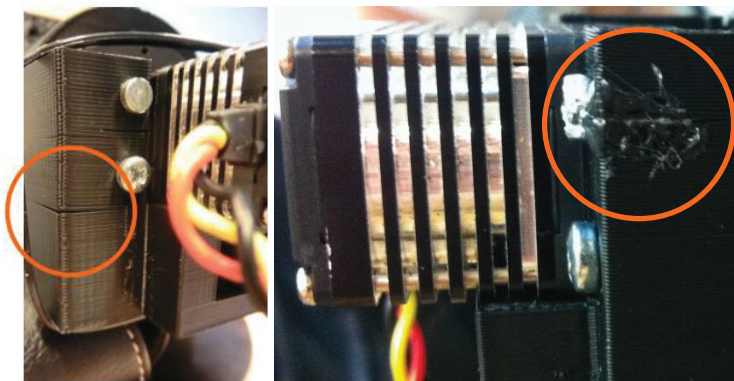
Iterationsarbetet var tidskrävande på många sätt. Inte bara på grund av 3D-skrivarens bearbetnings- och utskrivningstid, utan även med avseende på att styrning och övriga tester av robotarmen inte kunnat utföras fullt ut innan alla komponenter monterats.

Det är en utmaning att få en hög hållfasthet för robotarmen samtidigt som den totala vikten är låg.

De utskrivna detaljerna är inte solida, utan har ett skal och innanför detta en bikakestruktur för att spara vikt. Beroende på faktorer som fyllnadsgrad, utskriftshastighet och orientering så får både skalet och den inre strukturen olika egenskaper. ABS-plasten lider av relativt stor termisk längdutvidgningskoefficient vilket resulterar i spänningar i materialet då komponenterna svalnar. Spänningarna kan i sin tur göra att somliga trådlager delaminerar. (38)

Utöver risk för ökad friktion så påverkar även orienteringen på fibrerna hållfastheten av de 3D-utskrivna komponenterna. Om fibrerna orienteras fel mot de belastande krafterna som komponenten utsätts för så riskerar lagren att dras isär och resultera i sprickor och brott.

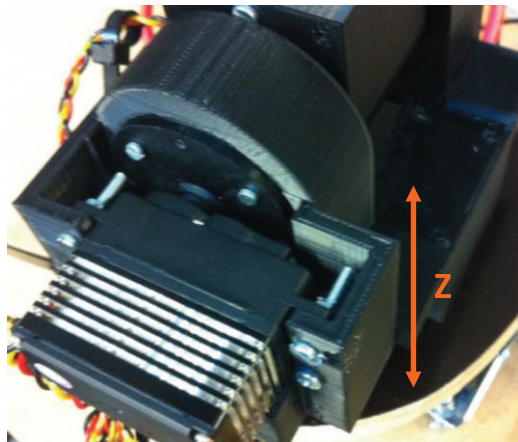
Figur 46 visar robotarmens axelled där sprickor och brott har uppkommit till följd av hög belastning på ett utsatt område. Området är utsatt i den meningen att komponentens godstjocklek är tunnare här än på övriga ställen och trådstrukturen är orienterad så att lagren har dragits isär. I gripdonets fall har utskriften fungerat bättre eftersom belastningen är liten.



Figur 46 - Sprickor i axelled, olimmad till vänster samt limmad till höger.

Belastningen är som högst i axelleden i och med att det är denna del som tar upp störst del av momentet som uppstår när robotarmen är fullt utsträckt. Om robotarmens rörelser dessutom är ryckiga och snabba så riskerar belastningen att bli ännu större. Figur 47 visar hur drivsystemet är monterat mot axeln. Här kan även ses att de största krafterna kommer att uppstå i höjdlid, längs z-axeln, när drivsystemet driver axelleden. Största belastningen blir då som tidigare nämnts mot fiberstrukturens orientering, vilket till slut har bidragit till sprickor och brott.





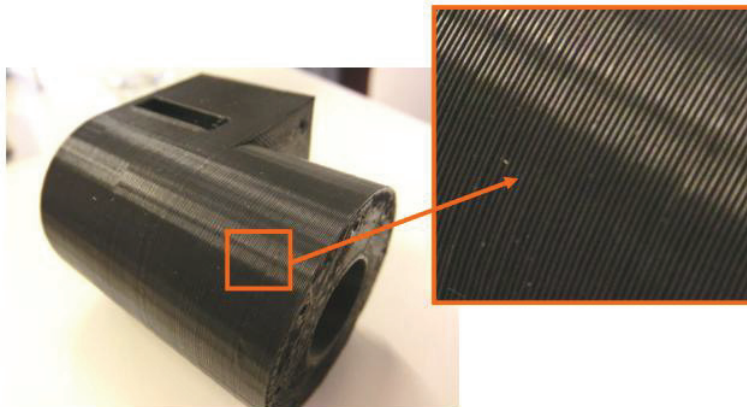
Figur 47 - Montering av drivsystem i axelled.

Under utskrivningsprocessen har heller inte exakt samma skrivarparametrar använts för alla komponenter. Detta tillsammans med övriga presenterade svårigheter gällande materialegenskaper gör att det inte finns något sätt att enhetligt räkna ut hållfastheten för komponenterna. Det enda sättet att få en uppfattning av styrkan hos en komponent är att tillverka flera identiska komponenter och utsätta ett urval av dessa för belastning till dess att brott uppstår. (38)

Friktion är en stor utmaning alla typer av mekaniska system, vilket ger störningar och motstånd i styrning och reglering. Den framtagna prototypens leder har väldigt hög friktion vilket betyder att 3D-skrivaren inte kunde skriva ut med den ytfinitet som var förväntad.

Skrivaren som har använts för att realisera komponenterna med för detta projekt använder FDM-teknik (Fused Deposition Modeling). Komponenterna blir inte homogena i och med att tekniken innebär att en tunn plasttråd smälts fast vid underliggande trådar. Resultatet blir då att den färdiga komponenten har en fiberliknande struktur och komponentens egenskaper beror därför av hur dessa fibrer är orienterade. Orienteringen beror på hur detaljen placeras och växer fram i 3D-skrivaren. (38)

I och med att skrivaren använder FDM-teknik så skrivs komponenterna ut i lager. Dessa blir då räfflade, vilket kan ses i Figur 48. Resultatet kan då bli ökad friktion om komponenterna ska glida mot varandra.



Figur 48 - Vy över 3D-utskriven komponents yta.

Vid realisering av en konstruktion är det en utmaning att designa konstruktionen på ett sådant sätt att montering är möjlig.

Montering av samtliga delar och komponenter är även det en betydande utmaning för en prototyp utskriven i 3D på grund av att utskriftsmaterialet är sprött. Det är svårt att bearbeta materialet i form av nedslipning och uppborrning av hål samt skapa nya profiler. Vid montering kan det även vara svårt att komma åt med monteringsverktyg om detta inte har anpassats i CAD-modellerna.

## 5.2 Drivsystem

Dimensionering av motorer är en utmaning då hänsyn måste tas till flera olika faktorer. Motorerna dimensioneras för att, utifrån position i konstruktionen, klara av att driva robotarmen med den totala lasten. Denna last består dels av den statiska lasten i form av robotens egen vikt inklusive vikten av föremålet som ska greppas, och dels av den dynamiska lasten i form av tröghetsmoment som uppstår vid acceleration eller vid obalans. Motorn i basleden kräver störst dragkraft på grund av att den utsätts för längst hävarm och därmed, i vald konstruktion, även störst belastning. Eftersom alla elektriska komponenter i systemet måste samkoordineras då de drivs av samma nätaggregat är det en utmaning att mata en motor med rätt spänning och ström för att denna ska kunna operera optimalt enligt angiven data.

Servona som driver prototypens arm matas inte med tillräcklig spänning och ström eftersom sidecar endast ger ut 6 V och 3 A. Detta resulterar i att de inte arbetar med det angivna vridmomentet. Detta visades i axelleden då servot inte kunde generera ett tillräckligt stort moment för att dra upp prototypens arm.

### 5.3 Styrning

En stor utmaning vid programmering av robotstyrning är att skapa en struktur i realtidsprogrammet, där många faktorer ska samspela. I den framtagna prototypens fall ska sensordata, kinematikberäkningar och servosignaler kombineras i samma program. Detta kan medföra svårigheter då dessa faktorer är beroende av varandra. Varje process ska fungera i förhållande till flera andra processer och data ska även skickas mellan dessa. Processerna i realtidsprogrammet ska dessutom skicka data till FPGA som kommunicerar med hårdvaran. Detta kan resultera i ett ytterligare kommunikationsproblem som beror på att FPGA och realtidsprogrammet arbetar med olika frekvenser.

Den största utmaningen med styrningen är att få robotarmens rörelser kontrollerade. Genom att skicka signaler till armens servon utifrån den rampformade rörelseprofilen regleras rörelsen, men utmaningen ligger dels i att bestämma hur stora steg servonas vinkelförändring ska delas upp i för att få ett bra flyt och dels i hur ofta ett nytt värde från rampen ska skickas till servona.

Det är även en utmaning att få en kontrollerad greppningsrörelse hos gripdonet. Prototypens gripning kontrolleras genom användning av sensorer och en stegning av styrsignalen. Utmaningen låg därför i att hitta den längd på stegen som möjliggör att kontrollera rörelsen med kontinuerlig avläsning av sensorerna så pass att gripdonet stannar precis då sensorerna registrerar ett tryck. Då servona tar långa steg finns risk att det sista steget klämmer sönder objektet, trots att sensorn reagerar. Om steglängden kortas ned blir rörelsen däremot väldigt långsam vilket inte heller är önskvärt.

Den kinematiska beskrivningen av robotarmen är ytterligare en utmaning då alla leders koordinatsystem matematiskt ska relateras till varandra. Utifrån den kinematiska beskrivningen kan robotarmens referensramar orientera sig i förhållande till varandra, givet TCP, men det kan vara svårt för roboten att orientera sig i omgivningen. Detta problem är speciellt tydligt i den framtagna prototypens fall, där användaren anger koordinater manuellt utifrån robotarmens bas. Föremål kan vara lokaliserade på olika platser i arbetsrummet, vilket kan medföra problem för användaren att avgöra på vilken koordinat detta finns.

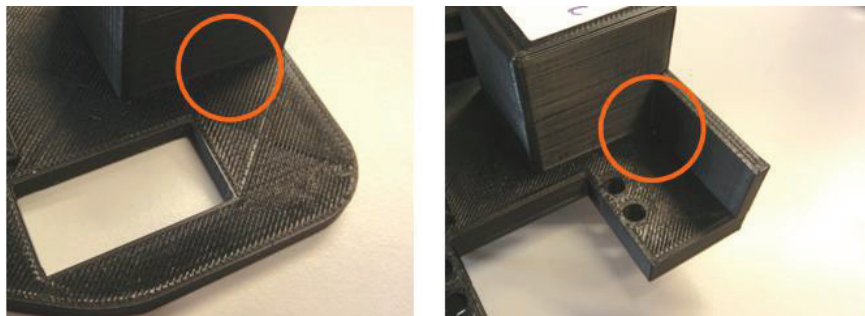
## 6 Diskussion

Att utveckla en robot från grunden är omfattande, komplicerat och kräver tvärvetenskaplig kunskap. Metoden att identifiera vilka utmaningar som upplevs vid robotframtagning genom att undersöka både teoretiska och praktiska aspekter anses vara effektiv. Den praktiska delen, att tillverka prototypen, gav upphov till en iterationsprocess som förbättrade prototypen under projektets gång. Iterationsprocessen kunde dock ha skett i fler steg och ytterligare iterationer, vilket pekar på robotikens stora omfattning och komplexitet.

Alla delar är beroende av varandra och måste därför utvecklas parallellt även om de olika delarna kräver en viss sekvens. Det kan vara en god idé att börja med att definiera krav för roboten i en kravspecifikation, utifrån vilken arbetet blir mätbart samt möjligt att utvärdera och verifiera. En kravspecifikation är även fördelaktigt för att vision och uppfattning av projektets mål ska vara samma för samtliga inblandade i projektet. Därefter bör en lämplig mekanism för robotarm och gripdon bestämmas. Utifrån mekanismerna kan resterande delar såsom val av drivsystem och material samt kinematik och styrning utvecklas parallellt.

### 6.1 Konstruktion

Vid design och konstruktion av komponenter så bör, förutom dimensionering, även tankar om godstjocklek och stödpunkter vara en central del av processen. Delar där komponenten kommer att utsättas för mest krafter, belastning och slitage bör förstärkas med gods utan att för den delen minska eller hindra funktionen som önskas. För att minska risken för sprickor och brott bör även design efter stödpunkter och framförallt stöd i flera dimensioner finnas i åtanke. I Figur 49 visas gripdonets basplatta, dels efter första iterationen (vänster) då markerat område stöddes i 2 dimensioner, och dels efter andra iterationen (höger) då markerat område stöddes i 3 dimensioner. Komponenten har alltså genom iterationer ökat stödet och minskat risken för sprickor och brott utan att minska eller förlora funktionen.



Figur 49 - Utskriven komponent med stöd i 2 (vänster) respektive 3 dimensioner (höger).

Det är även bra att undvika inåtgående hörn samt tänka på att förstärka konstruktionen vid hål för att minska brottrisken på grund av höga spänningskoncentrationer.

Det valda materialen för hela konstruktionen ska vara slitstarka med hög hållfasthet och låg densitet. Konstruktionen bör ha hög hållfasthet och samtidigt vara lätt för att undvika alltför kraftiga och dyra drivsystem.

En rekommendation är att implementera någon typ av lager i lederna, exempelvis glidlager. Konstruktionen blir dock mer komplex och man får då ha i åtanke att det kan krävas flera iterationer för att glidlagrena ska kunna passas in i lederna. I basleden är en lagerliknande lösning implementerad. Friktionen är låg, men i en större robot skulle det vara bättre att använda ett standardiserat lager.

Att skriva ut komponenter med hjälp av en 3D-skrivare är en smidig och relativt snabb metod att använda för att realisera en produkt. Genom att rita upp modeller med hjälp av CAD-mjukvara och sedan skriva ut dessa som fysiska komponenter så kan koncept och idéer smidigt testas. Ett problem med 3D-skrivning är dock att hållfastheten tenderar att bli låg. Eftersom 3D-skrivaren skriver ut i lager är det extra känsligt med belastning som drar isär lagren. 3D-utskrivning är förvisso en smidig tillverkningsmetod, men med den tillgängliga 3D-skrivaren i prototyplabbet på Chalmers Tekniska Högskola, lämpar den sig endast för prototyp tillverkning eller för små robotar.

Om 3D-utskrivning ska användas rekommenderas hög kvalitet på utskriftsmaterialet och skrivare, trots en hög kostnad. Detta för att minimera den friktion som uppstår vid rotation i leder, men även för en ökad precision mellan CAD-modell och utskrift.

Vid design av komponenter är det viktigt att ta hänsyn till hur delarna ska monteras. Vissa delar kan behöva delas upp i flera delar som sedan monteras samman i efterhand. Det är också viktigt att tänka på att det ska vara möjligt att komma åt med verktyg som exempelvis skruvmejslar. En annan viktig aspekt rörande monteringen är att se till att kablarna är placerade och fästa i konstruktionen så att de ej kläms eller förhindrar den önskade rörelsen.

## 6.2 Drivsystem

Om möjligheten finns att välja samma typ av drivsystem till hela roboten bör det göras. Detta underlättar vid såväl programmeringen som vid planering av de elektriska komponenter som krävs för drivsystemen, eftersom alla motorer då kräver samma sorts styrning och elektronik.

För att välja rätt servo, eller annat valt drivsystem, är det viktigt att dimensionera med hänsyn till alla faktorer, som last, tröghetsmoment och friktion. Att använda säkerhetsfaktor är även bra för att få marginal vid dimensionering av drivsystem.

### 6.3 Styrning

Programmen som styr roboten har itererats fram i många olika versioner. Många av dessa iterationer kunde ha undvikits om det från början hade funnits en bättre planering för vad programmet skulle utföra. För att snabbt komma igång med programmering hade det varit hjälpsamt med ett flödesschema som tar upp alla delar av konstruktionen som ska styras. Utifrån detta flödesschema kunde programmets uppgifter ha definierats på ett tydligare sätt. Något som dock har hjälpt oss under programmeringen har varit att undersöka och inspireras av tillgängliga exempelprogram som finns på *National Instruments* hemsida och i *LabVIEW*.

Att ha separata realtidsprogram för arm och gripdon är inte den bästa lösningen. I den framtagna prototypens fall finns tre program; ett realtidsprogram för armen, ett realtidsprogram för gripdonet samt ett gemensamt FPGA-program för PWM-generering och sensoravläsning. Från början fanns en önskan om att göra ett kombinerat realtidsprogram för både gripdon och arm, så att det totalt hade funnits endast två program. För att lösa det hade programmet behövt innehålla sensordata, kinematikberäkningar, beräkning av servosignaler utifrån en rörelseprofil samt ett flertal dataöverföringar till FPGA-programmet, kombinerat i samma program. Då detta testades blev programmet snabbt komplext och för att underlätta gjordes därför ytterligare ett program för gripdonet, vilket gjorde att just kommunikationsproblemen mellan realtidsprogrammen och FPGA-programmet underlättades. Däremot medförde detta istället två andra problem; styrningen blev mer spretig i och med att fler program skulle koordineras samt mindre sammanhållen i och med att gripdonet och armen måste styras var för sig.

En del inom styrningen som kunde ha lösts på bättre sätt är hur armens servon styrs. Hur många steg rampen ska delas upp i samt hur ofta ett värde från rampen ska skickas till servona skulle behöva justeras kraftigt. I dagsläget delas servots vinkelförändring upp i 50 steg oavsett hur stor förändring som önskas. Detta kommer att medföra att prototypens rörelse blir ryckig till följd av stora steg om förändringen är stor, medan stegen blir mycket små om endast en lite förändring önskas. De 50 värdena från rampen skickas sen till servona med 100 millisekunders fördröjning vilket gör att det alltid tar 5 sekunder för roboten att ställa in sig efter den önskade TCP-koordinaten, oavsett hur stor förändringen är. Hur lång tid en förflyttning tar borde ju vara proportionell mot hur långt roboten ska röra sig, vilket gör detta till ett område som går att utveckla betydligt för prototypen.

En kinematisk beskrivning av en robot blir mest effektiv och strukturerad om den baseras på en utvecklad konvention, exempelvis en som bygger på Denavit-Hartenberg konventionen. Kinematiken bakom en robot är omfattande och av detta skäl bör arbetet inom ämnet påbörjas tidigt i ett projekt.

#### 6.4 Fortsättningsarbete

Utöver de utmaningar som behandlades i projektet finns det ytterligare stora utmaningar inom främst styrning, men även konstruktion. Ett fortsättningsarbete skulle kunna vara att göra robotarmen mer flexibel genom att implementera ytterligare två leder i gripdonet: en led som är parallell med armbågsleden och en led vars axel genomskär och är vinkelrät till föregående. Detta skulle resultera i en typ av vrist likt en mänsklig handled.

Den framtagna prototypen är inte anpassad efter estetik och i ett fortsatt arbete kan konstruktionen anpassas ytterligare på ett sådant sätt att kablar döljs inuti gripdon och leder. Detta skulle även vara fördelaktigt med tanke på den minskade sårbarheten för kablarna.

Inom styrning kan ett utvecklingsområde vara att öka automationsgraden för prototypen. Exempel på områden som kan öka detta kan vara någon typ av miljöavläsning, i form av exempelvis IR- eller ultraljudssensor. Genom en sådan miljöavläsning skulle roboten kunna hitta föremålet utan användarinteraktion. Den framtagna prototypen har ingen förmåga att beräkna den kortaste möjliga vägen till föremålet, utan för att utföra beräkningarna skulle det behövas en banoptimeringsalgoritm. Utmaningen blir då att implementera algoritmen i styrningen så att roboten får snabb respons. När banoptimering implementeras bör även risken för självkollision undersökas.

Det viktigaste fortsättningsarbetet med styrningen handlar om att släppa lösningen med de 50 stegen som rampen just nu skapar. Istället borde rampen dela upp intervallet i ett visst antal steg baserat på hur stor förändringen är, så att alla steg alltid är lika långa. På det sättet skulle roboten få ett jämnare rörelsemönster där tiden som en förändring tar blir proportionell mot förändringens storlek.

Servona som driver armen kan i dagsläget endast rotera 160 grader, detta skulle kunna justeras till 180 grader med hjälp av en servoprogrammerare vilket skulle öka robotens arbetsrum. Vid fortsatt arbete skulle det även vara viktigt att utveckla regleringen av robotens rörelser. Utöver att justera rampens längd kan extra regulatorer implementeras i styrningen. I och med att servona har inbyggda regulatorer prioriterades ej ytterligare reglering trots att den inbyggda regleringen inte alltid är tillräcklig. Genom att addera ytterligare regulatorer i programmeringen kan rörelserna bli mjukare och stabilare.

## 7 Slutsats

Att konstruera en robot är omfattande, komplicerat och tvärvetenskapligt vilket även gör det till ett intressant område. Det finns många utmaningar som måste lösas och inom detta projekt har flera identifierats och undersökts.

Den mest övergripande utmaningen är att planera arbetsprocessen för att konstruera en robot. Alla delar i en robot är beroende av varandra och därav krävs en parallell utvecklingsprocess av vissa delområden.

Det är en utmaning att få en robotarm med lätt vikt med en hög hållfasthet. I projektet tillverkades många komponenter med hjälp av 3D-skrivare som är en smidig och snabb realiseringsmetod. De utskrivna komponenterna var av låg vikt, men hållfastheten visade sig vara låg och oberäknelig. Detta visade sig vid montering och belastning av komponenterna. Denna metod rekommenderas därför främst för prototypstillverkning eller för komponenter utsatta för endast små belastningar. Vid 3D-utskrivning rekommenderas hög kvalitet på 3D-utskrivare och utskriftsmaterial. Det är även viktigt att orientera komponenterna rätt i skrivaren för att generera en så hög hållfasthet som möjligt. För att få en konstruktion med en högre hållfasthet är det bra att ha stöd i tre dimensioner, undvika inåtgående hörn och förstärka konstruktionen vid hål. Friktion är en annan stor utmaning som gör sig tydlig i leder. Det är rekommenderat att implementera lager, exempelvis glidlager, i alla leder för att minimera friktionen.

Utmaningen gällande drivsystem är att dimensionera med tanke på flera faktorer, varav vissa kan vara svåruppskattade. Det är även en utmaning att försörja drivsystemet med tillräckligt drivmedel för att det på så sätt ska prestera maximalt. Det är fördelaktigt att välja samma typ av drivsystem för en robot för att på så sätt få en homogen drivmedelsförsörjning och styrning.

Utmaningarna inom styrningen utgjordes av att kontrollera prototypens rörelser vid rörelse av arm och gripning. Det var även en utmaning att skapa en programstruktur som var överskådlig och effektiv. I *LabVIEW* finns flera exempel som kan fungera som underlag för programmering, om detta programmeringsspråk ska användas.

Det finns ytterligare utmaningsområden att undersöka som exempelvis miljöavläsning och banoptimering. Robotik är som sagt ett stort område med många olika metoder och lösningar som kan implementeras. I detta projekt har endast en bråkdel av dessa undersökts och det finns många ytterligare att upptäcka.



## 8 Litteraturförteckning

1. IFR, International Federation of Robotics. IFR, International Federation of Robotics. [Online].; 2012 [cited 2014 mars 10. Available from: <http://www.ifr.org/robots-create-jobs/>.
2. ISO, International Organization for Standardization. ISO, International Organization for Standardization. [Online].; 2012 [cited 2014 mars 10. Available from: <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>.
3. IFR, International Federation of Robotics. IFR, International Federation of Robotics. [Online].; 2013 [cited 2014 Mars 10. Available from: <http://www.ifr.org/robots-create-jobs/>.
4. Ivancevic DVG, Ivancevic TT. Human-Like Biomechanics. 28th ed. Tzafestas PS, editor. Dordrecht: Springer Netherlands; 2006.
5. Zatsiorsky V, Prilutsky B. Biomechanics of Skeletal Muscles. 1st ed. Robertson LD, editor. Champaign: Human Kinetics; 2012.
6. Prinz FB, Atwood CL, Richard FA, Beaman JJ, Brown RL, Fussell PS, et al. Rapid Prototyping in Europe and Japan. Panelrapport. Baltimore: Loyola College; 1997. Report No.: ISBN 1-883712-44-0.
7. Nationalencyklopedin. Nationalencyklopedin, Mekanism. [Online].; 2014 [cited 2014 april 11. Available from: <http://www.ne.se/lang/mekanism>.
8. C.S. Lee G. Robot Arm Kinematics, Dynamics and Control. Ann Arbor: University of Michigan, Data; 1982.: IEEE.
9. H. Asada H. Robot Mechanisms. <http://ocw.mit.edu/courses/mechanical-engineering/2-12-introduction-to-robotics-fall-2005/lecture-notes/chapter3.pdf>. Cambridge: Massachusetts Institute of Technology, Department of Mechanical Engineering; 2005.
10. McCarthy JM, Soh GS. Geometric Design of Linkages. 2nd ed. Antman SS, Marsden JE, Sirovich L, editors. New York: Springer; 2011.
11. Nice K. How Stuff Works. [Online].; 2000 [cited 2014 maj 02. Available from: <http://science.howstuffworks.com/transport/engines-equipment/bearing1.htm>.
12. Coppelia Robotics. Coppelia Robotics. [Online].; 2014 [cited 2014 maj 02. Available from: <http://www.coppeliarobotics.com/helpFiles/en/jointDescription.htm>.
13. Bera TK, Samantaray AK, Karmakar R. Bond graph modeling of planar prismatic joints. In A. K, editor. Mechanism and Machine Theory. Kharagpur: Indian Institute of Technology; 2011. p. 2-20.

14. Monkman GJ, Hesse S, Steinmann R, Schunk H. Robot Grippers. 1st ed. Hesse S, editor. New Jersey: John Wiley & Sons; 2007.
15. Nationalencyklopedin. Nationalencyklopedin, Kinematik. [Online].; 2014 [cited 2014 maj 19. Available from: <http://www.ne.se/lang/kinematik>.
16. Nationalencyklopedin. Nationalencyklopedin, Dynamik. [Online].; 2014 [cited 2014 maj 19. Available from: <http://www.ne.se/lang/dynamik/157808>.
17. Waldron K, Schmiedeler J. Kinematics. In Siciliano B, Khatib O, editors. Springer Handbook of Robotics. New York: Springer; 2008. p. 23-29.
18. Nationalencyklopedin. Nationalencyklopedin, Stegmotor. [Online].; 2014 [cited 2014 maj 2. Available from: <http://www.ne.se/stegmotor>.
19. National Instruments. National Instruments. [Online].; 2011 [cited 2014 Mars 20. Available from: <http://www.ni.com/white-paper/3656/en/+&cd=1&hl=sv&ct=clnk&gl=se>.
20. Futaba. Futaba. [Online]. [cited 2014 Mars 20. Available from: <http://www.futaba-rc.com/servos/digitalservos.pdf>.
21. Nationalencyklopedin. Nationalencyklopedin, Logaritmisk Potentiometer. [Online].; 2014 [cited 2014 maj 2. Available from: <http://www.ne.se/logaritmisk-potentiometer>.
22. Nationalencyklopedin. Nationalencyklopedin, Potentiometer. [Online].; 2014 [cited 2014 maj 2. Available from: <http://www.ne.se/lang/potentiometer>.
23. Otten G, J.A. de Vries T, van Amerongen J, M. Rankers A, W. Gaal E. Linear Motor Motion Control Using a Learning Feedforward Controller. IEEE/ASME Transactions on Mechatronics. 1997 September; II.
24. Britannica Online Encyclopedia. Encyclopædia Britannica Online. [Online].; 2014 [cited 2014 maj 2. Available from: <http://www.britannica.com/EBchecked/topic/182667/electric-motor/45833/Linear-induction-motors>.
25. National Instruments. National Instruments, What is CompactRIO. [Online].; 2014 [cited 2014 maj 2. Available from: <http://www.ni.com/compactrio/whatis/>.
26. Xilinx. Xilinx, FPGA. [Online].; 2014 [cited 2014 maj 2. Available from: <http://www.xilinx.com/training/fpga/fpga-field-programmable-gate-array.htm>.
27. National Instruments. National Instruments, NI 9403. [Online]. Austin: National Instruments; 2012 [cited 2014 Mars 19. Available from: <http://sine.ni.com/ds/app/doc/p/id/ds-165/lang/sv>.

28. FIRST Robotics Competition. Digital Sidecar Data Sheet 2012 FIRST Robotics Competition. Manchester;; 2012.
29. National Instruments. National Instruments, NI 9201. [Online]. Austin: National Instruments; 2012 [cited 2014 Mars 19. Available from: <http://sine.ni.com/ds/app/doc/p/id/ds-184/lang/sv>.
30. Nationalencyklopedin. Nationalencyklopedin, Sensor. [Online].; 2014 [cited 2014 maj 2. Available from: <http://www.ne.se/lang/sensor>.
31. Howe RD. Tactile Sensing and Control of Robotic Manipulation. Journal of Advanced Robotics. 1994; III(8).
32. Interlink Electronics. FSR Force Sensing Resistor Integration Guide and Evaluation Parts Catalog. 2014..
33. Nationalencyklopedin. Nationalencyklopedin, Trådtöjningsgivare. [Online].; 2014 [cited 2014 maj 2. Available from: <http://www.ne.se/tr%C3%A5dt%C3%B6jningsgivare>.
34. National Instruments. National Instruments, Getting started. [Online].; 2008 [cited 2014 maj 19. Available from: <http://www.ni.com/white-paper/7001/en/>.
35. Sun J. Pulse-Width Modulation. In Vasca F, Lannelli L, editors. Dynamics and Control of Switched Electronic Systems. New York: Springer; 2012. p. 25-61.
36. Barr M. Pulse Width Modulation. Embedded Systems Programming. 2001 September.
37. Lennartson B. Reglerteknikens grunder. 4th ed. Lund: Studentlitteratur AB; 2006.
38. Stigler G. [Mail].; 2014-06-04.
39. National Instruments. National Instruments, Testing and Debugging LabVIEW FPGA Code. [Online].; 2013 [cited 2014 Juni 14. Available from: <http://www.ni.com/tutorial/51862/en/>.