



CHALMERS



# Design och reglering av en inverterad pendel

Demoanläggning som visar ett exempel på reglerteknik  
Examensarbete inom högskoleingenjörsprogrammet Mekatronik

Siavash Khoshaman  
Tommy Gelotte

INSTITUTIONEN FÖR ELEKTROTEKNIK  
CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2021

[www.chalmers.se](http://www.chalmers.se)

# Konstruktion och reglering av en inverterad pendel

**Siavash Khoshaman**  
**Tommy Gelotte**



Institution för Elektroteknik  
Chalmers Tekniska Högskola  
Göteborg, Sverige 2021

Examinator: Veronica Olesen, Institution för Elektroteknik, Chalmers Tekniska Högskola  
Handledare: Emil Andersson, John Tagesson Ritz, Midroc Automation

## Förord

Detta examensarbete utfördes av två mekatronikstudenter från Chalmers tekniska högskola. Arbetet är utfört inom institutionen för Elektroteknik och i samarbete med Midroc Automation. Examensarbetet motsvarar 15 högskolepoäng utav 180 högskolepoäng för högskoleingenjörutbildning. Arbetet utfördes i största del i Midrocs kontor och verkstad i Göteborg och innehåller flera delar av utbildningens innehåll, mestadels kring dynamiska system och reglerteknik.

Vi skulle vilja tacka Midroc Automation och särskilt våra handledare Emil Andersson och John Tagesson Ritz för denna möjlighet och för all stöd och support under projektets gång. Slutligen skulle vi vilja tacka vår examinator Veronica Olesen för all support under hela utbildningen och detta arbete.

Siavash Khoshaman och Tommy Gelotte, Göteborg 2021

# Sammanfattning

Reglering är ett brett begrepp som innefattar många områden inom industri och privatliv. Temperaturer, vattennivåer och mycket annat justeras för att processer, maskiner och inomhusklimat ska fungera. Midroc Automation är ett företag som arbetar med olika typer av regleringar inom industrin. Därför vill Midroc visa ett interaktivt exempel på reglering, som kan tas med och visas runt om i världen på olika mässor. Resultatet av detta är en balanserad inverterad pendel, med tillhörande styrskåp. Även om det inte finns många liknande exempel på inverterade pendlar inom industrierna som Midroc Automation arbetar med, är det ett typexempel på en klassisk variant av reglering som visar upp att hårdvaran i kombination med mjukvaran klarar av att hindra pendeln från att falla.

Denna rapport beskriver det projekt som genomfördes för att konstruera en demoanläggning i form av inverterad pendel för Midroc Automation. Projektet bestod av fyra huvuddelar: beslut om hårdvara, konstruktion av anläggningen, reglering och programmering samt verifiering av slutgiltigt resultat.

Den färdigställda anläggningen består av en linjär bana där pendeln sitter fast på en vagn som förflyttar sig åt sidorna. Styrenheten består av Siemens PLC med tillhörande moduler och motordrivare för kontroll av motorn. Regleringsprocessen kan styras via en HMI-skärm där användaren kan ställa in olika PID-parametrar för att reglera pendeln till upprätt position. Reglering av pendeln fungerar under testning och uppnår givna mål och krav.

Under projektets gång diskuterades olika lösningar kring konstruktion och beslut om hårdvara och dessa beställdes i samråd med Midroc. Slutligen testades anläggningen av framtida användare samt ansvarig personal hos Midroc Automation och de var tillfreds med lösningen som togs fram.

# Abstract

Automatic control is a widely used concept that includes many areas in industry and private life. Temperatures, water levels and much more are adjusted for processes, machines and indoor climate to work. Midroc Automation is a company that works with different types of control applications in industry. Therefore, Midroc wants to show an interactive example of automatic control, which can be transported and shown around the world at various exhibitions. The result of this is a balanced inverted pendulum, with associated control cabinets. Although there are not many similar examples of inverted pendulums in the industries that Midroc Automation works with, it is a typical example of a classic variant of automatic control that shows that the hardware is able to prevent the pendulum from falling.

This report describes the project that was carried out to construct a demo facility in the form of an inverted pendulum for Midroc Automation. The project consisted of four main parts: Decision on hardware, construction of the plant, regulation and programming and verification of final results.

The resulting plant consists of a linear path where the pendulum is attached to a carriage that moves to the sides. The control unit consists of Siemens PLC with associated modules and motor driver for controlling the motor. The control process can be adjusted via an HMI screen where the user can set various PID parameters to control the pendulum to the upright position or make a disturbance to the process. Automatic control of the pendulum works during testing and achieves the given goals and requirements.

During the project, various solutions regarding design and decisions about hardware were discussed and these were ordered in consultation with the supervisors at Midroc. Finally, the equipment was tested by future users and responsible staff at Midroc Automation and they were satisfied with the solution that was developed.

# Innehållsförteckning

<b>1 Inledning</b>	<b>1</b>
1.1 Bakgrund	1
1.2 Syfte	1
1.3 Avgränsningar	1
1.4 Precisering av frågeställningen	2
<b>2 Teoretisk / teknisk bakgrund</b>	<b>3</b>
2.1 Beskrivning av en inverterad pendel	3
2.2 Programmable Logic Controller (PLC)	3
2.3 Human Machine Interface (HMI)	4
2.4 Likströmsmotor (DC motor)	4
2.5 Hall-sensor	4
2.6 Induktiv givare	5
2.7 Tillståndsmodell	5
2.8 PID-reglering	5
2.9 Kaskadreglering	6
2.10 TIA portal	7
2.11 PROFINet och PROFI-safe	7
2.12 IRT-kommunikation	7
<b>3 Metod</b>	<b>8</b>
3.1 Förberedelser och tidsplan	8
3.2 Beräkning	8
3.3 Hårdvara	9
3.4 Programmering	9
3.5 HMI design	9
<b>4 Testning av mekanik i Matlab och Simulink</b>	<b>10</b>
4.1 Beräkning av mekanik	10
4.2 Tillståndsmodell i Matlab	12
4.3 Simulering i Simulink	13
4.4 Första försöket	14
4.5 Andra försöket	17
4.6 Vagnens hastighet	20
<b>5 Beslut om hårdvara</b>	<b>21</b>
5.1 Motor och motordrivare	21
5.2 Linjär bana	24
5.3 Vagnen	25
5.4 PLC enhet	28

5.5 I/O Moduler	28
5.6 HMI enhet	29
5.7 Rotationsgivare	29
5.8 Ändlägesgivare	30
5.9 Nätaggregat	30
5.10 Nätverksswitch	31
<b>6 Programmering</b>	<b>32</b>
6.1 Programmets struktur	32
6.2 Viktiga variabler	33
6.3 Specifika inställningar för motordrivaren	34
6.4 Kalibrering av givare och position	35
6.5 Start av reglering	36
6.6 Programmets viktiga funktioner	36
6.6.1 Motor_Control_FC	36
6.6.2 PID_Control_And_Rotation_Sensor_OB	37
6.7 PID-reglering	37
6.7.1 Kaskadreglering	37
6.7.2 Inställning av PID-parametrar	38
6.7.3 Teoretiskt test av PID-värden	39
<b>7 HMI</b>	<b>42</b>
7.1 Beskrivning av startskärmen	42
7.2 Beskrivning av skärm med PID-inställningar	44
7.3 Beskrivning av skärm för manuell styrning	45
<b>8 Resultat</b>	<b>46</b>
8.1 Svar på frågeställningar	47
<b>9 Diskussion</b>	<b>48</b>
9.1 Planering och arbetsupplägg	48
9.2 Siemens hårdvara	48
9.3 Simulering av pendeln	48
9.4 Den linjära banan	49
9.5 Förbättringsmöjligheter	49
9.6 Etik och hållbarhet	50
<b>Referenslista</b>	<b>51</b>
<b>Bilaga 1</b>	<b>54</b>
Friläggning	54
Linjärisering	56
Rörelseekvationer	56
<b>Bilaga 2</b>	<b>57</b>

# Terminologi / förkortningar

HMI. Human Machine Interface

PLC. Programmable Logic Controller

I/O. In-/utportar

CAD. Computer Aided Design

PID. Proportional Integral Derivative

TIA Portal. Totally Integrated Automation Portal

FBD. Function Block Diagram

LD. Ladder

SCL. Structured Control Language

GRAPH. Sekvensprogrammering

IRT. Isochronous Real Time

# 1 Inledning

## 1.1 Bakgrund

Midroc Automation är en av Sveriges främsta leverantörer av automatiserade system. Företaget driver många stora projekt men även mindre. Midroc automation vill framställa en demoanläggning för att kunna visa upp på mässor. Anläggningen ska uppvisa vilka tekniska metoder och de system och hårdvaror som företaget arbetar med.

Midroc Automation arbetar mycket med att visa upp sig för att hitta nya medarbetare på arbetsmarknaden och är av den anledningen mycket aktiva på olika mässor. Därför behövs en portabel reglerteknisk anläggning som ska kunna transporteras runt för att visa vad Midroc arbetar med. Arbetet utförs i Midroc Automations egna kontor och verkstad.

## 1.2 Syfte

Målet var i första hand att konstruera en mindre anläggning för att visa upp på mässor. Anläggningen demonstrerar ett exempel på reglerteknik i form av balansering av en inverterad pendel längs en räls. Styrningen hanteras av en Siemens PLC och det finns möjlighet för extra kontroll och observering med en interaktiv skärm.

Den färdiga produkten ska kunna stå upp av sig själv, samt så ska den kunna återhämta sig om pendeln påverkas av yttre krafter. Det ska gå att reglera pendelns position på rälsen och den ska kunna förflytta sig utan att falla.

Anläggningen skapades i första hand för att kunna visas upp på mässor och övriga uppvisningstillfällen i Midroc Automations intresse. Den ska visa reglerteknik på ett intressant och interaktivt sätt, samtidigt som den ska kunna flyttas relativt enkelt.

## 1.3 Avgränsningar

Projektets fokus kommer att ligga på dimensionering och reglering av den fysiska maskinen. Övriga moment som inte handlar om detta kommer inte att undersökas lika mycket. Även om de olika mekaniska delarnas hållfasthet tas i beaktande kommer det inte att gås in på detaljer om det. Denna rapport kommer inte heller att behandla bygget av styrskåp, installation eller transportlösning.

## 1.4 Precisering av frågeställningen

För att få ett fungerande system som ska kunna visas upp på mässor finns det en del frågor på vägen.

- Kommer PLC:n att hinna reglera processen, eller behövs något snabbare?
- Är det möjligt att skapa en modell i Matlab eller liknande program för att förutspå vilka reglerparametrar som passar till den fysiska maskinen?
- Går det att reglera en inverterad pendel längs en linjär bana med den hårdvara som Midroc Automation jobbar med?

## 2 Teoretisk / teknisk bakgrund

I detta kapitel beskrivs olika tekniska koncept som används i projektet. Se kapitel 5 för en mer detaljerad beskrivning av besluten kring vilken hårdvara som valdes för projektet.

### 2.1 Beskrivning av en inverterad pendel

Det finns olika varianter av reglerade inverterade pendlar inom reglertekniken, men alla bygger på samma princip: Att balansera en stående stav som annars skulle ramla av sig själv. I detta projekt handlar det om en inverterad pendel som är fäst i en vagn som kan åka längs en rak bana. För att ge en bild av detta kan det jämföras med en Segway. Figur 2.1 demonstrerar en Segway.



Figur 2.1. Segway x2 och i2. Från [1], CC-BY

En Segway, som visas i figur 2.1, fungerar på en liknande princip som en inverterad pendel på rörlig vagn, då den använder sina hjul för att kompensera så att den håller sig upprätt.

### 2.2 Programmable Logic Controller (PLC)

PLC är en programmerbar styrenhet som används främst inom automationsindustrin för att reglera och styra olika slags processer och maskiner. För att användaren lättare ska kunna styra processen kopplas PLC oftast till något slags operatörspanel som exempelvis en HMI-panel. Processen börjar med att tillståndet på ingångarna läses av och sedan jämförs med det program som är inskrivet i PLC:n. Sedan ställs utgångarnas tillstånd så att de överensstämmer med det inskrivna programmet och på detta sätt styrs processen på önskad sätt [2]. I projektet har Siemens PLC SIMATIC S7-1500 använts med syfte att kunna visa fördelarna med denna enhet för olika företag intresserade av styrenheter.

## 2.3 Human Machine Interface (HMI)

HMI är en slags instrumentpanel som gör det möjligt för användaren att kunna kommunicera och styra enheten som detta är kopplat till. HMI paneler används brett inom olika delar av industrin som exempelvis fordonsindustrin, mat och medicintillverkning. Syftet med HMI panel är att visa parametrar i processen i realtid. Genom en HMI skärm kan användaren ta sig in i en process och ändra och justera olika parametrar för bättre resultat i systemet [3]. I projektet användes Siemens Simatic HMI Unified comfort panel för att göra det möjligt för användaren att kunna styra processen med pendelns position på banan och även en del andra möjligheter.

## 2.4 Likströmsmotor (DC motor)

Likströmsmotor är den vanligaste typen av elektriska motorer. Arbetssättet går till så att motor matas med elektrisk ström som omvandlas till rörelseenergi [4]. Detta åstadkoms genom att lindningar av koppartråd (även kallade spolar) skapar magnetfält när ström går genom dem. Dessa magnetfält går emot andra magnetfält, ofta permanentmagneter, vilket orsakar krafter som påverkar rotorn, som i sin tur börjar snurra.

Det finns olika varianter av likströmsmotorer men ofta delas de in i två grupper, borstade och borstlösa motorer. Borstade motorer har strömledande borstar som glider mot en yta som kallas kommutatorn, som har i uppgift att leda strömmen till rätt spolar [4]. Detta gör så att rätt spole skapar magnetfält som då får motorn att snurra. Borstlösa motorer är elmotorer som inte behöver några borstar för att fungera, vilket ofta leder till att behovet av underhåll på motorn minskar.

I detta projekt används en borstlös 24V DC motor med intern rotor från EBM-papst, med tillhörande växellåda för att uppnå tillräckligt med moment och passande hastighet. Den styrs av en motordrivare från Siemens.

## 2.5 Hall-sensor

En hall-sensor är en sensor som reagerar på magnetfält och med denna funktion gör det möjligt att ha uppsikt över positionen eller vinkeln av den del av system som den är kopplad till [5]. Detta betyder att det går att skapa en rotationsgivare som baseras på samma princip. Genom en sådan rotationsgivare kan man kontrollera vinkeln på en mekanisk detalj då den omvandlar rotationsrörelse till digitala signaler. Rotationsgivare används i många olika produkter som exempelvis hushållsprodukter men även inom elektronik och fordonsindustrin. I detta projekt används en rotationsgivare på pendelns rotationsaxel för att kunna känna av axelns vinkel.

## 2.6 Induktiv givare

En induktiv givare är uppbyggd av en spole och kondensator som bildar ett elektromagnetisk fält vid framsidan av givaren. När ett metalliskt föremål närmar sig en induktiv givare upptäcks föremålet av en elektrisk krets som finns inbyggt i givaren och det i sin tur ändrar utsignalen från givare [6]. Induktiva givare används brett inom dagens elektronik och har många användningsområden på grund av sina fördelar. Eftersom induktiva givare saknar rörliga delar så har dessa lång livslängd och även hög precision. I detta projekt används induktiva givare som ändlägesgivare på vardera sida av den linjära enheten.

## 2.7 Tillståndsmodell

En tillståndsmodell (kallad “state-space model” på engelska) är ett sätt att förutspå framtida värden av en samling av variabler, med hjälp av deras tidigare värden [7]. Detta gör det möjligt att bevaka systemets olika delar, som sedan kan återkopplas för reglering. För att skapa en tillståndsmodell behövs de differentialekvationer som beskriver systemet. Dessa stoppas sedan in i konstanta matriser A, B, C och D. Ekvationer (1) och (2) visar hur modellen ser ut.

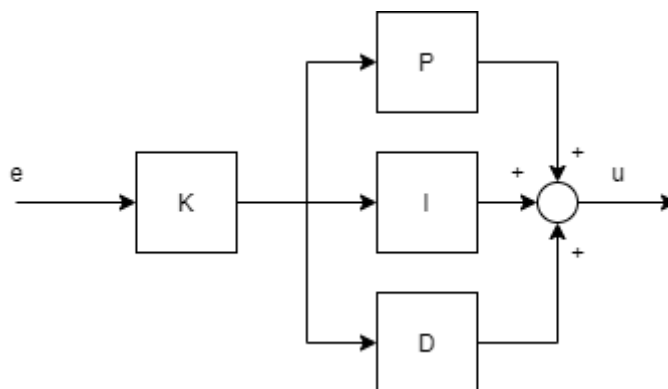
$$\frac{dx}{dt} = Ax + Bu \quad (1)$$

$$y = Cx + Du \quad (2)$$

Där  $x$  är en vektor som innehåller systemets tillståndsvariabler (kallas för tillståndsvektorn), och  $u$  är systemets styrsignaler.

## 2.8 PID-reglering

PID-regulator är den vanligast förekommande regulatorn inom industrin. Parametrarna PID står för proportionell, integrerande och deriverande verkan. Blockschemat till en PID regulator visas i figur 2.2. Sambandet mellan insignal och utsignalen i en PID-regulator beskrivs i ekvation (3).



Figur 2.2. Blockschemat för PID-regulator. Från [8]

$$u(t) = K \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \cdot e'(t) \right] \quad (3)$$

PID-regulatorn tar in signalen  $e(t)$  som är skillnaden mellan börvärdet (det önskade värdet) och ärvärdet (det faktiska värdet), och skickar ut en styrsignal  $u(t)$ . PID-regulatorn består av en P-del, en I-del och en D-del. För att justera dessa finns det tre parametrar:  $K$ ,  $T_i$  och  $T_d$ .

Tre parametrar finns att justera för att få en passande reglering:

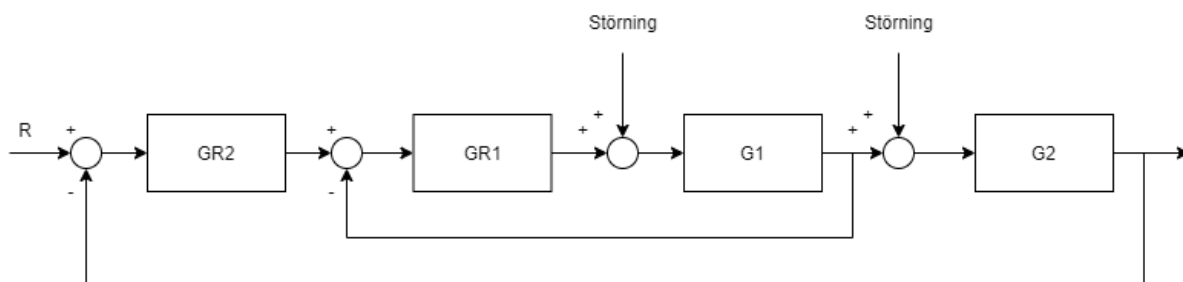
P-delen av regulatorn kan också kallas för förstärkningen (gain på engelska). Denna delen tar felet,  $e(t)$ , och multiplicerar det med  $k$ -värdet. Detta ger en enkel form av reglering då styrsignalen,  $u(t)$ , helt enkelt får ett värde som är proportionellt med felet och  $k$ -värdet.

I-delen av regulatorn får den multiplicerade signalen från P-delen och tar integralen från den. Detta betyder att I-delen ger större signaler ju längre tid börvärdet är annorlunda från ärvärdet. För att justera styrkan på detta så används parametern  $T_i$ . Eftersom  $T_i$  är inverterad ökar effekten från I-delen om  $T_i$  minskar.

D-delen av regulatorn är inte alltid nödvändig, eller passande. Den tar derivatan av den multiplicerade signalen från P-delen och ger ett bidrag till utsignalen. Detta betyder att ju snabbare insignalen ändras, desto större effekt har D-delen. Den går att justera med parametern  $T_d$ .

## 2.9 Kaskadreglering

Kaskadreglering är en regleringsmetod som är vanligt förekommande inom industrin. Kaskadreglering kan tillämpas till de processer som kan delas upp i två seriekopplade delprocesser. Vid kaskadreglering använder man oftast två PID-regulatorer och metoden går ut på att utsignalen från ena PID-regulatorn sätts som börvärde för den andra regulatorn. Med genom kaskadreglering kan man förhindra störningar i ett tidigt skede i processen då det används en extra återkoppling jämfört med andra typer av reglering [9]. Ett typiskt användningsområde för kaskadreglering är nivåreglering av en vattentank. Blockschemat i kaskadreglering är uppbyggd av en inre loop och en yttre loop. Den inre loopen förhindrar och motverkar störningar i ett tidigt skede och sedan skickas signalen till den yttre regulatorn. Se figur 2.3 för ett generellt blockschema för kaskadreglering.



Figur 2.3. Standard blockschema för kaskadreglering. Från [9].

## 2.10 TIA portal

TIA portal är Siemens egna program för programmering av PLC, HMI och andra enheter. I TIA portal skapas funktioner och funktionsblock som utgör de olika delprogrammen. De skapade funktionerna aktiveras i Main-funktionen för att kunna fungera. Utöver detta finns det även datablock som används för lagring av data, samt organisationsblock som fyller mer specifika funktioner som till exempel interrupts och liknande. Funktionerna och funktionsblocken kan programmeras i olika språk som till exempel: FBD, LD, SCL och GRAPH.

## 2.11 PROFINet och PROFI-safe

PROFINet är en ethernet-kommunikationslösning baserad på internationella standarder som används brett inom dagens automationsindustri. Syftet med PROFINet är byte av data mellan styrenheter och komponenter. PROFINet används inom processer som kräver snabb datakommunikation med väldigt kort cykeltid [10]. En av fördelarna med PROFINet är möjligheten att använda säkerhetssystemet PROFI-safe. I praktiken kopplas alla komponenter tillsammans med ethernetkabel, och valet görs i TIA portal att man vill använda PROFINet som kommunikationsprotokoll.

PROFI-safe är en teknologi som används inom automation för säkerhetskommunikation. Via PROFI-safe uppnås en mer säker datakommunikation vilket leder till högre funktionssäkerhet [11].

## 2.12 IRT-kommunikation

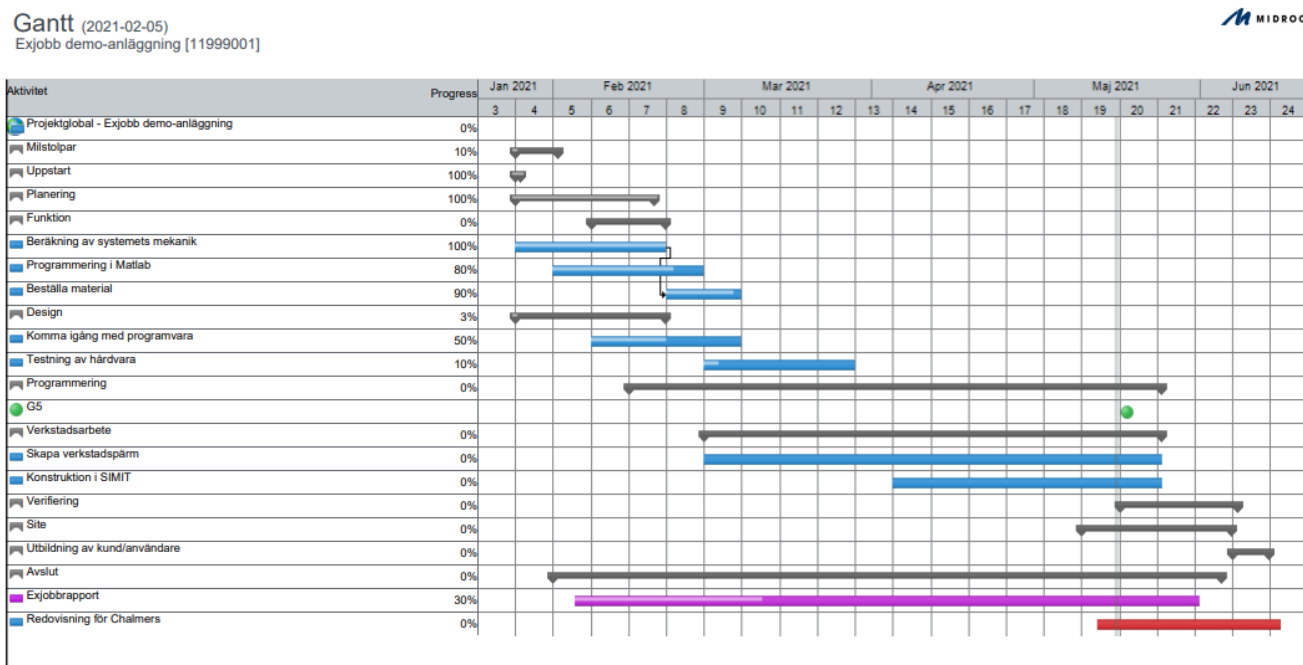
Vanligtvis vid kommunikation med ethernet-kablar skickas informationen med varierande intervall, vilket kan orsaka problem då signaler behöver skickas på ett tillförlitligt sätt. Med IRT-kommunikation går det att skicka information med fasta tidsintervall över ethernet [12]. I detta projektet används IRT för att skicka signaler till motorn med samma intervall, för att säkerställa att motorn uppdateras så snabbt den kan.

### 3 Metod

I detta kapitel kommer arbetets metodik att presenteras. Det följer arbetets gång i samma ordning som följande kapitel, fast med mer fokus på omständigheterna och metoderna än det faktiska resultatet.

#### 3.1 Förberedelser och tidsplan

Programmet Antura användes i projektet för att planera en rimlig tidsplan. I Antura hade man möjlighet att bocka av de delar av arbetet som är färdiga och även tillägga ärenden inom de olika delarna som kommentarer på vad som behöver göras för att en del ska räknas som färdigt. Figur 3.1 visar gantt-schemat som gjordes i Antura som används av Midroc för projektplanering.



Figur 3.1. Gantt-schema i Antura

#### 3.2 Beräkning

Programmet Matlab användes för att analysera olika mekaniska modeller för systemet. Modellerna användes för att dimensionera komponenter till projektet samt för att åstadkomma initialvärden på regulatorparametrar. Flera rapporter studerades där det gjordes jämförelse mellan framtagna ekvationer och värden för att kunna bekräfta att dessa är rimliga.

### 3.3 Hårdvara

Den största och viktigaste delen av projektet var att kartlägga alla delar som skulle användas, både mekaniska och elektriska. Hårdvaror som motordrivare och HMI panel valdes för att under projektets gång utvärdera och analysera dessa inför kommande projekt på Midroc. Andra komponenter så som PLC, I/O-enhet samt nätverksswitch valdes eftersom de normalt används på Midroc inom olika projekt.

För att kunna beställa rätt linjär enhet till projektet diskuterades de krav som linjära enheten borde klara för att vara passande till projektet. Dessa krav var maximal hastighet som vagnen kan komma upp i samt möjlighet till motoranpassning på banan. Genom att besöka flera butiker med automationslösningar och samtal med olika tillverkare av linjära enheter kunde man hitta den mest passande linjära enheten till projektet som klarar av de satta kraven.

Programmet Fusion 360 och en 3D-skrivare användes för att skriva ut en del specifika delar till projektet så som adapter mellan banans vagn och pendelns fäste samt fäste för pendeln. Även fästet för ändlägesgivare och koppling mellan banans fäste och rotationsgivaren skrevs ut via 3D-skrivaren.

Tia Selection Tool som är ett verktyg utvecklat av Siemens för framtagning av hårdvara användes för att kunna bestämma vilken motor som skulle användas till projektet utifrån de satta kraven. Dessa krav är maximal hastighet som motorn kan köra samt moment.

### 3.4 Programmering

För att skapa en plan över hur programmet skulle se ut diskuterades det i början av projektet på ett ungefär hur det skulle fungera. Det gjordes enkla skisser på olika sekvenser och funktioner.

Själva programmeringen gjordes i programmet TIA portal, som är skapat av Siemens. Det är en utvecklingsmiljö som har stort stöd för att programmera Siemens hårdvara, oavsett om det är motordrivare, PLC eller HMI-paneler. Programmet byggdes upp del för del så att alla olika delar kunde testas innan de användes skarpt. Säkerhetsfunktioner relaterade till motorns ändlägen skapades innan den körde på banan av sig själv.

### 3.5 HMI design

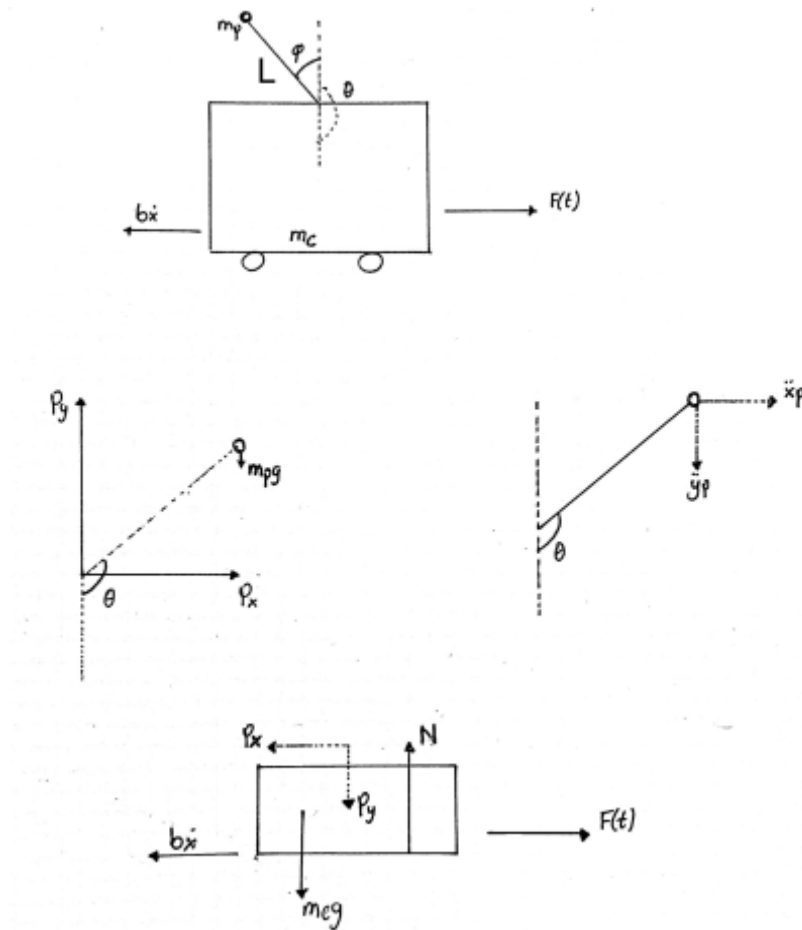
Genom diskussioner med Midroc och framtida användare kom man fram till vilka knappar och funktioner som är nödvändiga att ha med på HMI-skärmen samt var på skärmen dessa knappar ska placeras.

## 4 Testning av mekanik i Matlab och Simulink

Första steget var att beräkna mekaniken kring systemet för att få en djupare uppfattning om hur systemet fungerar samt bestämma vilka delar som kan användas i projektet. Mer specifikt är det vagnens potentiella topphastighet som är viktig att känna till. Med hastigheten går det att bestämma vilken motor som behövs och hur snabbt vagnen på banan behöver köras.

### 4.1 Beräkning av mekanik

I detta kapitel kommer inte alla steg av den mekaniska beräkningen att redovisas. För hela beräkningsgången, se bilaga 1. Beräkningarna skapades med hjälp av tidigare rapporter som bygger på samma principer som detta projektet [13][14]. Figur 4.1 visar friläggningarna av det mekaniska systemet, och variablerna som används i beräkningarna listas i tabell 4.1.



Figur 4.1. Friläggning av det mekaniska systemet.

Tabell 4.1. Lista på variabler som används i beräkningarna.

Variabel	Beskrivning
$x$	Vagnens position
$x'$	Vagnens hastighet
$x''$	Vagnens acceleration
$\varphi$	Pendelns vinkel, från stående position
$\varphi'$	Pendelns vinkelhastighet
$\varphi''$	Pendelns vinkelacceleration
$l$	Längd från pendelns rotationspunkt till dess tyngdpunkt
$g$	Jordens gravitationskonstant
$m_p$	Pendelns massa
$m_c$	Vagnens massa
$b$	Friktionskoefficienten
$I$	Pendelns tröghetsmoment
$F(t)$	Kraften som påverkar pendeln

Beräkningarna började med friläggning av det mekaniska systemet, som visas i figur 4.1. Via newtons lagar togs det fram rörelseekvationer, som sedan linjäriserades runt  $\varphi = 0$ . Efter linjäriseringen bryts  $x''$  och  $\varphi''$  ut. Rörelseekvationerna som till slut fås visas som ekvationer 4 och 5.

$$\varphi'' = \frac{m_p l(F(t) - bx') + m_p g l \varphi (m_c + m_p)}{m_c I + m_c m_p l^2 + m_p I} \quad (4)$$

$$x'' = \frac{(F(t) - bx')(I + m_p l^2) + m_p^2 g l^2 \varphi}{m_c I + m_c m_p l^2 + m_p I} \quad (5)$$

## 4.2 Tillståndsmodell i Matlab

Första steget med Matlab är att ställa upp rörelseekvationerna (4) och (5) i tillståndsmodellens fyra matriser. Dessa matriser används sedan för att beräkna alla intressanta parametrar. Det är i detta steget som de olika värdena på variablerna stoppas in, såsom längden på pendeln eller dess vikt. Detta ger då ut matriserna i numeriska former som sedan stoppas in i Simulink manuellt.

I detta projektet kommer tillståndsvektorn  $x$ , från tillståndsmodellen, att kallas för  $q$  för att inte blanda ihop denna med variabeln  $x$ . Vektorn  $u$  från modellen kommer här representera kraften på vagnen, så den innehåller bara en variabel. Ekvationer 6 och 7 visar tillståndsmodellen som används.

$$\frac{dq}{dt} = Aq + Bu \quad (6)$$

$$y = Cq + Du \quad (7)$$

Vektorn  $q$ :

$$\begin{bmatrix} x \\ \theta \\ x' \\ \theta' \end{bmatrix}$$

Matris A:

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{m_p^2 g l^2}{I(m_c + m_p) + m_c m_p l^2} & -\frac{b(I + m_p l^2)}{I(m_c + m_p) + m_c m_p l^2} & 0 \\ 0 & \frac{m_p g l (m_c + m_p)}{I(m_c + m_p) + m_c m_p l^2} & -\frac{m_p l b}{I(m_c + m_p) + m_c m_p l^2} & 0 \end{bmatrix}$$

Matris B:

$$\begin{bmatrix} 0 \\ 0 \\ \frac{I+m_p l^2}{I(m_c+m_p)+m_c m_p l^2} \\ \frac{m_p l}{I(m_c+m_p)+m_c m_p l^2} \end{bmatrix}$$

Matris C:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Då en tillståndsmodell skulle användas för att testa systemet så valdes alla fyra tillståndsvariabler som utsignaler genom att sätta 1:or i separata kolumner i C-matrisen som motsvarar varje tillståndsvariabel i vektorn q.

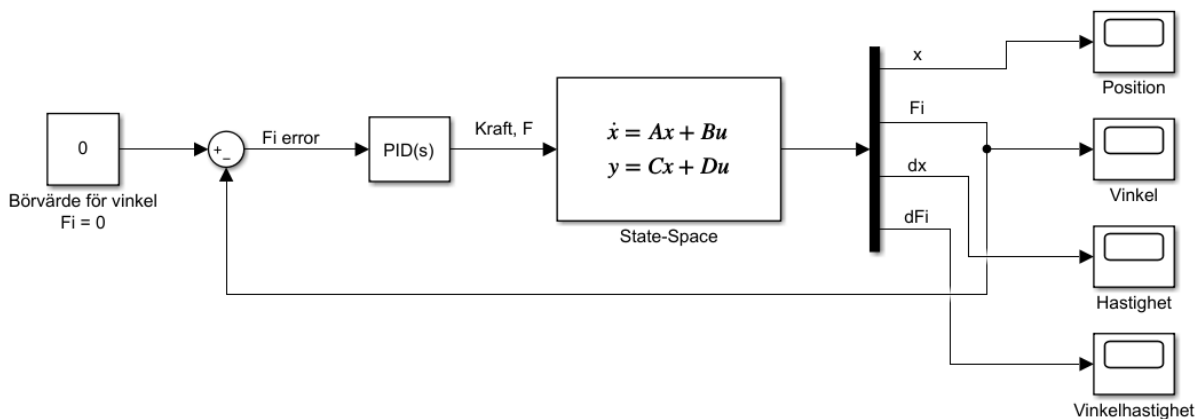
Matris D:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

### 4.3 Simulering i Simulink

Då det har gjorts en linjärisering av systemet är det viktigt att reflektera över vad som är rimliga antaganden och vilka slutsatser man kan dra från dem. Enligt [14] är en maximal vinkel på  $20^\circ$  ett rimligt antagande för den linjäriserade modellen. Det betyder att pendeln maximalt kan skilja sig  $20^\circ$  från stående position innan den teoretiska modellen inte längre går att lita på. Detta ansågs även vara rimligt för den verkliga pendeln, så vissa val så som maximal styrvinkel utgår från samma maxvinkel. Av samma anledning valdes det då att simuleringen går ut på att pendeln startar i ett läge då den ligger  $20^\circ$  från stående position, och målet är att vinkeln ska bli 0.

Det skapades ett program för att testa systemet i det fallet då pendeln ska återhämta sig från den antagna maxvinkeln. Detta gjordes genom att använda matriserna i Simulinks block för tillståndsmodeller för att räkna ut position, vinkel, hastighet och vinkelhastighet utifrån kraften på vagnen. För att få en reglering återkopplades pendelns vinkel och subtraheras från vinkelns börvärde. En PID-regulator kopplas in för att möjliggöra reglering av systemet. Se figur 4.2 för en bild från Simulink. PID-regulatorns parametrar togs fram genom iterativa tester.



Figur 4.2. Reglering av pendelns vinkel i Simulink.

Det är viktigt att notera att detta är en väldigt enkel reglering av systemet. Vagnens position tas inte i beaktning, och många delar av det verkliga systemet finns inte med. I verkligheten kommer en PID-regulator skicka en signal till systemet, som i sin tur skickar en signal till motorn som då påverkar vagnen med en kraft. Detta betyder att PID-regulatorn i den teoretiska modellen egentligen motsvarar flera processövergångar, och kan därför inte användas senare i projektet då en verklig PID-regulator ska skapas.

#### 4.4 Första försöket

Tidigt i arbetet gjordes uppskattningar om hur mycket alla delar skulle väga, och hur lång pendeln skulle vara. Dessa uppskattningar ligger till grund för kommande resultat. Tabeller 4.2 och 4.3 visar uppskattningen av variabler och vilka PID-parametrar som användes i simuleringen.

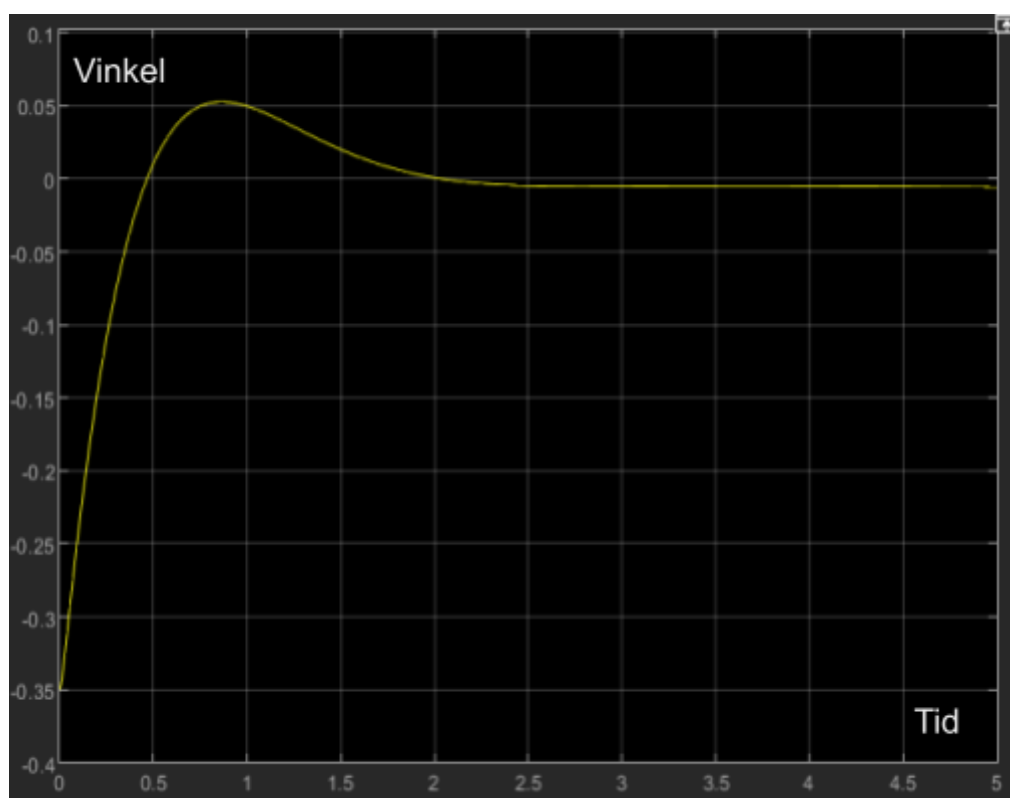
Tabell 4.2. Första uppskattningen av variabler

Massa på vagnen	0.4 kg
Massa på pendeln	0.2 kg
Längd till pendelns tyngdpunkt	0.5 m
Friktionskonstant	0.1

Tabell 4.3. PID-parametrar.

P	9
I	0.05
D	0.2

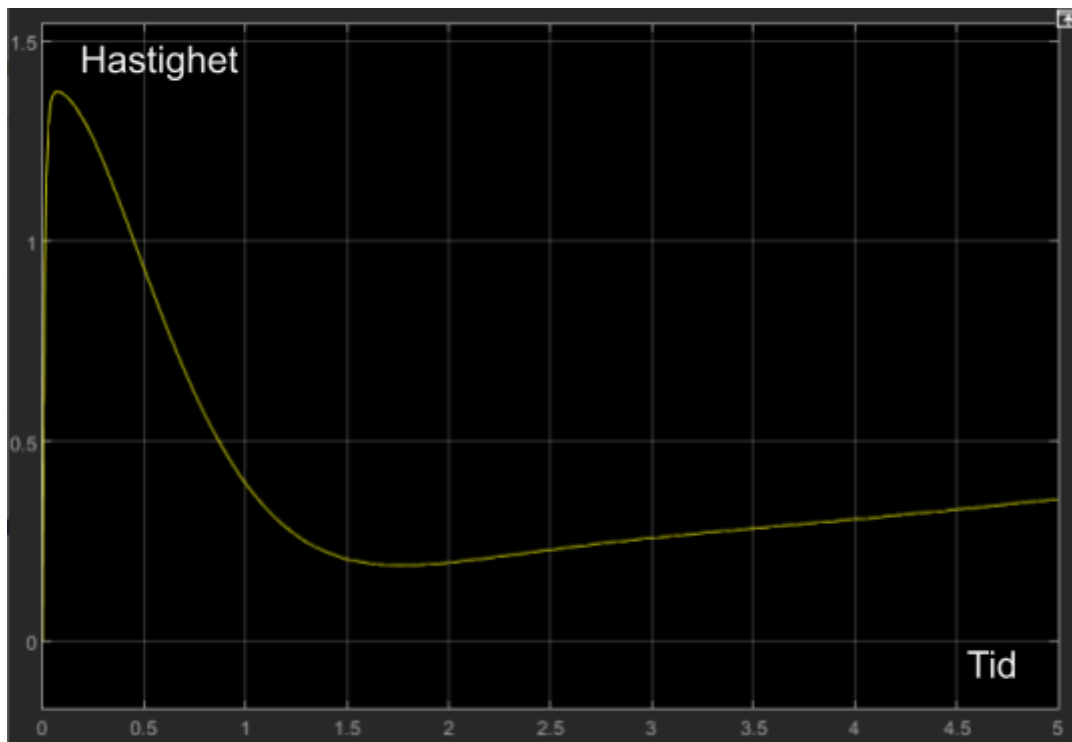
Efter insättning av uträknade matriser in till Simulink fås ett antal grafer. Dessa visas i figurer 4.3, 4.4 och 4.5. I figur 4.3 visas pendelns vinkel över tiden. Då det endast är vinkeln som regleras är det denna som undersöks för att justera regleringen.



Figur 4.3. Vinkel på pendel vid första försöket

Det kan noteras att kurvan i figur 4.3 gör en översväng innan pendeln ställer sig rakt upp. Detta betyder att vinkeln under en kort tid lägger sig på andra sidan av rakt stående position. I detta fallet var kurvans översväng välkomnat, för att få en mer passande rörelse på den verkliga pendeln. Det gjordes ett antagande tidigt att vinkeln bör göra en liten översväng för att få en snabb insvängning, då det viktigaste med denna simuleringen var att undersöka vagnens topphastighet.

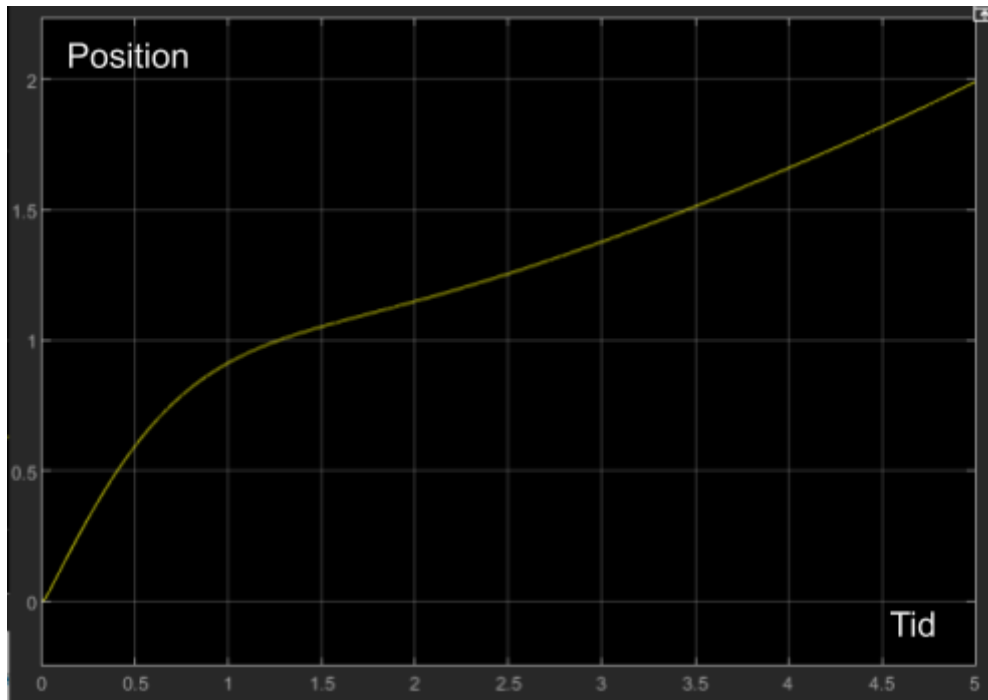
Den viktigaste datan från simuleringen visas i figur 4.4, nämligen hastigheten. Den hämtas samtidigt som vinkeln och alla andra värden från tillståndsmodellen.



Figur 4.4. Hastighet på vagn vid första försöket

I figur 4.4 går det att se att vagnen når en topphastighet på ca 1,4 m/s. Grafens riktning efter 2 sekunders-gränsen kan ignoreras då det egentligen bara visar på att positionen inte regleras. Med andra ord fortsätter vagnen att åka vidare efter att vinkeln har reglerats.

Till sist dokumenterades vagnens position, som visas i figur 4.5. Positionen är inte lika viktig som hastigheten och vinkeln, men kan vara intressant för att få en bild på hur systemet beter sig.



Figur 4.5. Position på vagn vid första försöket.

Figur 4.5 visar på samma sak som i figur 4.4, alltså att vagnen fortsätter att åka vidare på banan då den inte regleras.

## 4.5 Andra försöket

Senare i projektet kunde en bättre uppskattning av pendelns och vagnens vikt göras. Därför gjordes en ny simulering i Matlab och Simulink för att testa om resultatet påverkas. Tabell 4.4 visar de nya värdena. Detta betyder att simulationens PID-värden behövde ändras igen, då dessa anpassas efter systemet för att få en så bra reglering som möjligt. Nya PID-parametrar presenteras i tabell 4.5.

Denna simulering gjordes efter att många beslut om hårdvara hade tagits. Den gjordes för att undersöka ifall det teoretiska resultatet blev ogiltigt då annorlunda värden på massa och avstånd till tyngdpunkt används.

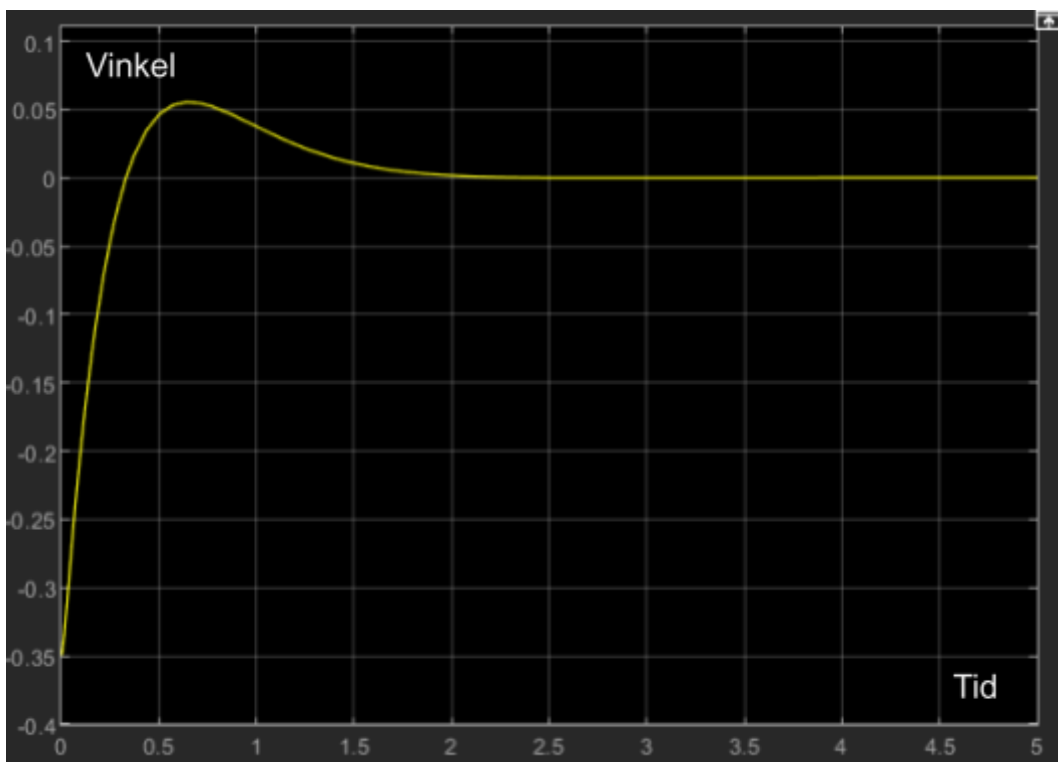
Tabell 4.4. Andra uppskattning av variabler

Massa på vagnen	0.3
Massa på pendeln	0.06
Längd till pendelns tyngdpunkt	0.3
Friktionskonstant	0.1

Tabell 4.5. PID-parametrar.

P	18
I	0.05
D	0.2

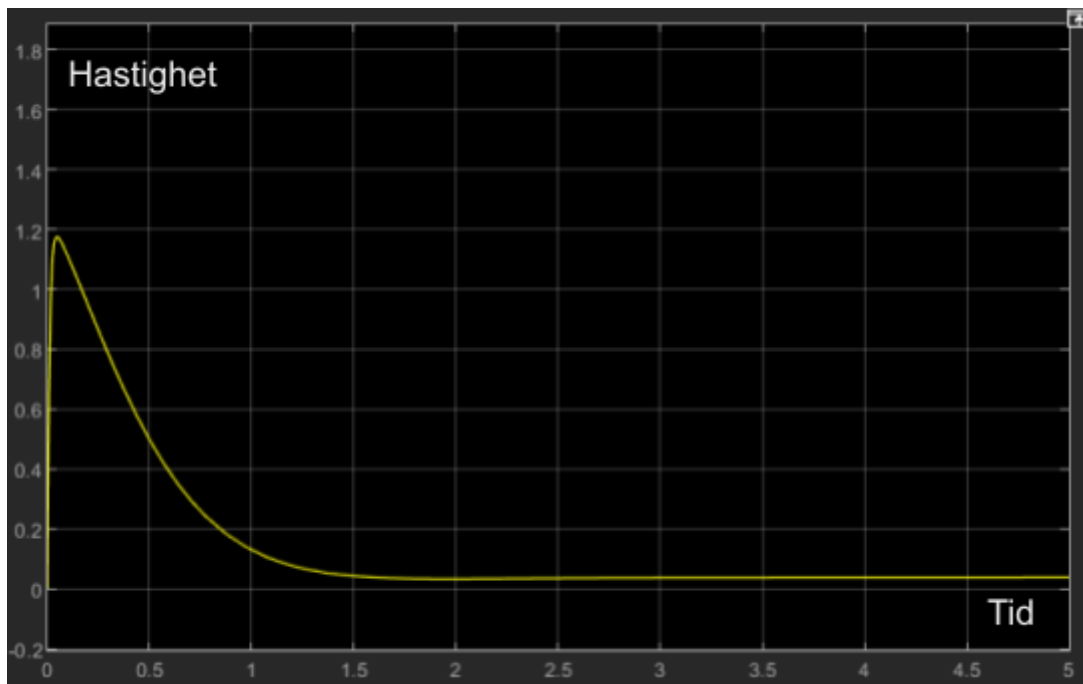
PID-parametrarna i tabell 4.4 justerades så att vinkelns kurva på ett ungefär motsvarar den i figur 4.3, från första försöket. Den nya vinkelkurvan visas i grafen i figur 4.6.



Figur 4.6. Vinkel på pendel vid andra försöket.

Kurvan i figur 4.6 ser ungefär likadan ut som den i figur 4.2, på grund av val av PID-parametrar. Detta var som sagt ett medvetet val då målet var att skapa samma scenario fast med andra fysiska värden.

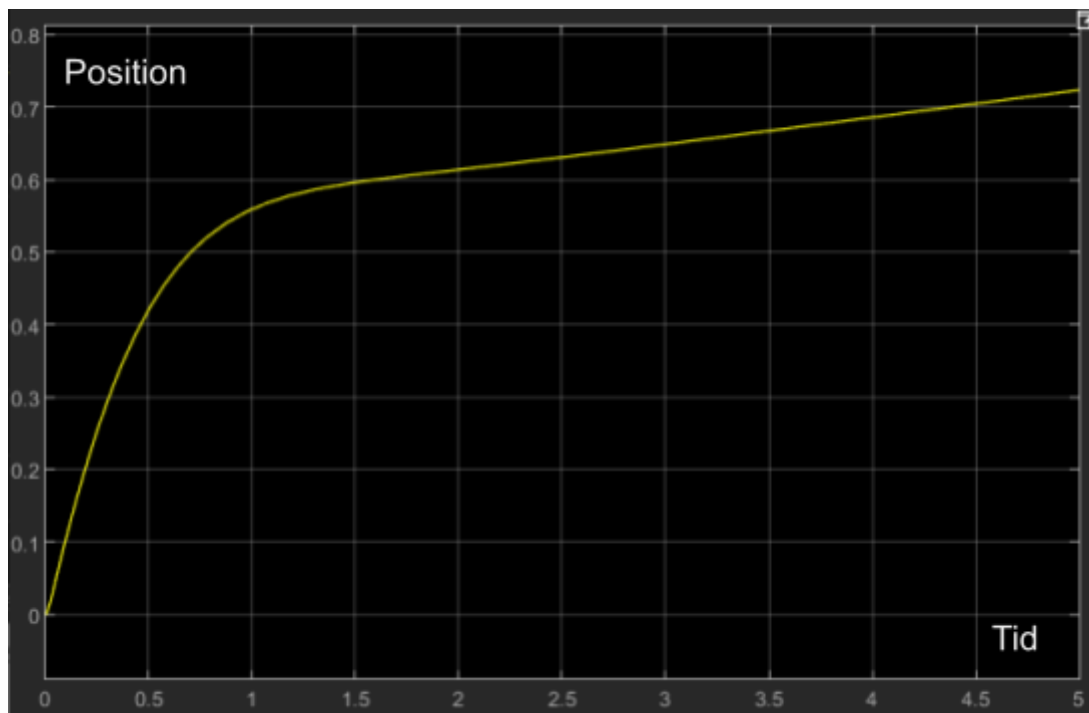
I figur 4.7 visas simuleringens mest intressanta värde, hastigheten.



Figur 4.7. Hastighet på vagn vid andra försöket.

För ungefär samma översväng på vinkeln så visar figur 4.7 att hastigheten blir något lägre med de nya värdena. Detta är ett positivt tecken då det betyder att det sätter mindre krav på motorn och banan, och regleringen bör vara enklare att åstadkomma.

Till sist visas vagnens position i figur 4.8.



Figur 4.8. Position på vagn vid andra försöket.

I figur 4.8 visas positionen vid andra försöket, vilket ser väldigt lik ut positionen i figur 4.5. Alltså är det samma här att vagnen fortsätter åka vidare på banan utan reglering. I stort visar de nya graferna att systemet med uppdaterade fysiska värden inte gör situationen värre, och att regleringen bör vara något enklare.

## 4.6 Vagnens hastighet

Efter båda försöken med olika värden på vagnen och pendelns massa ligger den nödvändiga topphastigheten för att kunna hålla pendeln i upprätt position runt 1,2 till 1,4 m/s. Men ifall verkligheten är mindre förlåtande än simuleringen sattes den förväntade nödvändiga topphastigheten till 2 m/s samt även för att möjligtvis kunna återhämta sig från vinklar större än  $20^\circ$ . Dessutom kommer andra beslut som baseras på vagnen och pendelns massa använda sig av de nyare och exakta värdena som användes i det andra försöket.

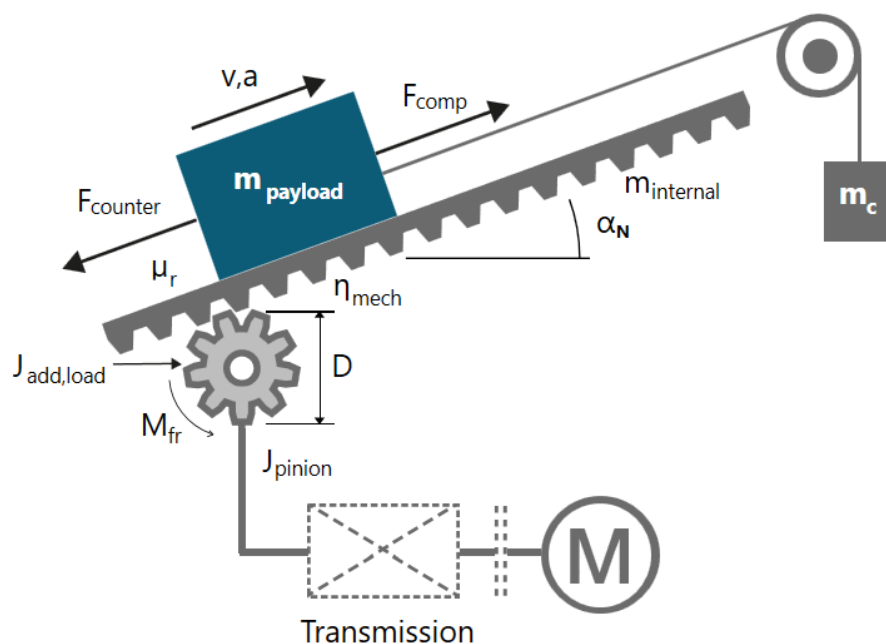
Det är också värt att påpeka att simuleringarna som gjordes är väldigt flexibla. Under utveckling av PID-parametrarna noterades det att olika värden gav ungefär samma kurva för vinkeln, men väldigt olika topphastigheter. Detta är en annan anledning till att den önskade topphastigheten på 2 m/s valdes, för att få marginal.

## 5 Beslut om hårdvara

Då projektet går ut på att bygga ett robust system som ska användas i verkligheten gavs det mycket fokus till att beställa rätt hårdvara. Nedan följer beskrivningar på hur besluten togs. För en lista på all hårdvara som används i projektet, se bilaga 2.

### 5.1 Motor och motordrivare

För att kunna balansera pendeln och hålla detta i upprätt position behövs det en motor som står för kraften som förflyttar vagnen på banan. Det är väsentligt att motorn i detta fall klarar av vissa krav från systemet, sådant som hastighet och moment. För att kunna välja rätt motor med de framtagna parametrarna kring hastighet, moment samt vikt användes Siemens TIA Selection Tool. Programmet gör det möjligt för användaren att få fram produkter från Siemens som klarar av de inmatade kraven i form av vikt, moment och hastighet. Se figur 5.1 för en bild på de variabler som man kan mata in i TIA Selection Tool.



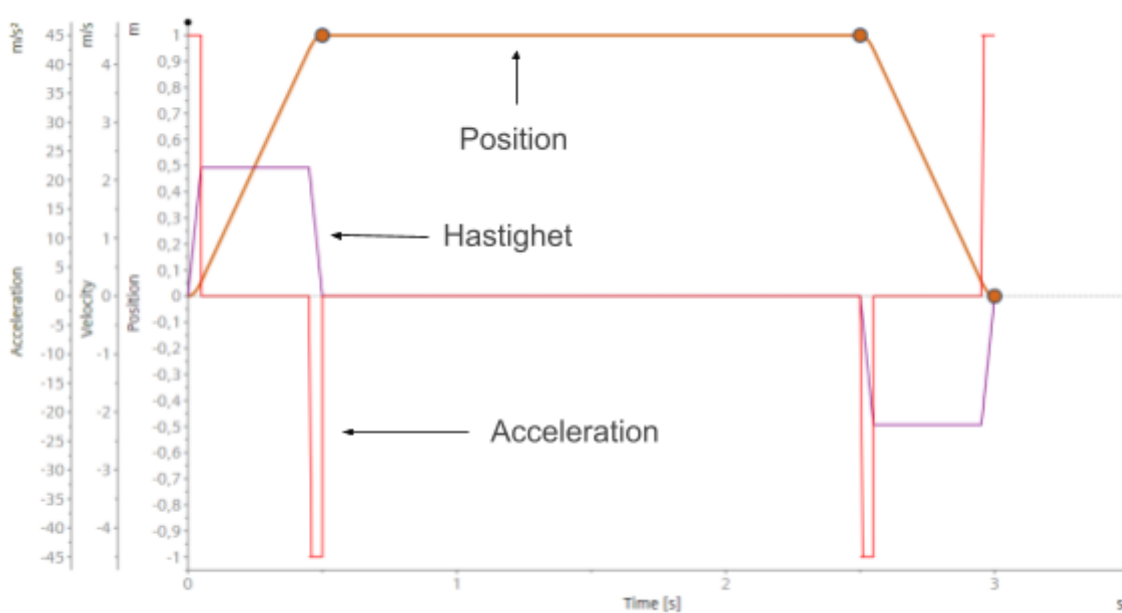
Figur 5.1. Bild på variabler för beräkning av motor, hämtad i TIA Selection Tool

För att få diametern på kugghjulet, dess tröghetsmoment samt vikten på olika rörliga delar i den linjära banan kontaktades tillverkaren. Dessa värden tillsammans med uppmätta massor på vagn och pendel skrevs in i Tia Selection tool. Valda värden visas i figur 5.2.

Mass			
m internal	Additional mass of the mechanics *	1.56	kg
m payload	Constant useful load	1.16	kg
m c	Counterweight	0.0	kg
Moment of inertia			
J pinion	Pinion moment of inertia	0.000238	kgm <sup>2</sup>
J add,load	Additional moment of inertia related to load	0.0	kgm <sup>2</sup>
Mechanic			
D	Diameter of the drive pinion *	57.3	mm
$\alpha$	Angle of inclination	0.0	°
Forces			
F counter	Constant additional force	0.0	N
F comp	Weighth-compensation	0.0	N
Friction			
$\mu_r,gleit$	Dynamic friction coefficient	0.03	
$\mu_r,haft$	Static friction coefficient	0.0	
M fr,gleit	Additional dynamic friction torque	0.0	Nm
M fr,haft	Additional static friction torque	0.0	Nm

Figur 5.2. Värden för beräkning av motor, hämtad i TIA Selection Tool

Vidare användes TIA Selection Tool för att fylla i en tidslinje med lastens rörelse. Då den potentiella topphastigheten valdes till ungefär 2 m/s skapades en kurva som motsvarar att vagnen åker en meter på en halv sekund, vilket visas i figur 5.3.



Figur 5.3. Bild på banans resekurva, hämtad i TIA Selection Tool

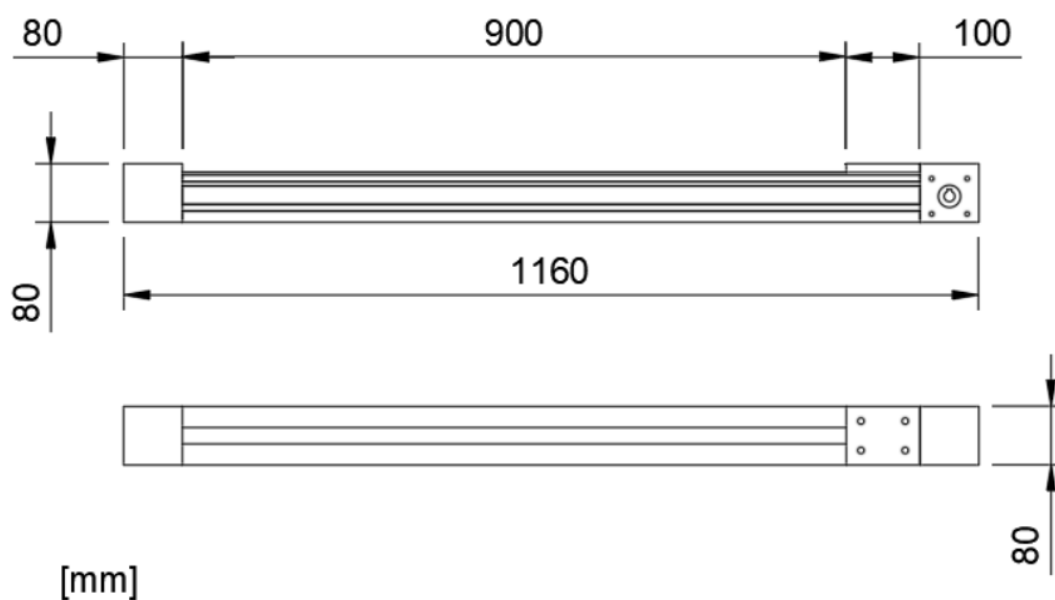
Efter att dessa parametrar har ställts in så fanns det endast en motor och en motordrivare som passade: motorn ECI 63.40 24V och växellådan PerformaxPlus 63.1 5:1 från tillverkaren ebm-pabst. Samtidigt valde TIA Selection Tool en passande motordrivare: PDC 600F SIMATIC MICRO-DRIVE, som visas i figur 5.4.



Figur 5.4. PDC600F med tillhörande kabel

## 5.2 Linjär bana

Den linjära enheten som användes i projektet beställdes från företaget ROLLCO som arbetar med automationslösningar. En rimlig slaglängd för den linjära enheten ansågs vara 900 mm, för att få marginal åt vardera kant på banan. Även den linjära enheten behövde klara vissa krav från processen, som exempelvis den maximala hastigheten som vagnen kan komma upp till på banan. Vidare diskuterades projektet med tekniska avdelningen på ROLLCO kring detaljer på enheten som skulle anpassas efter processen. För att lättare kunna koppla ihop motorn och banan beställdes motoranpassning till vald motor och den geometriska informationen om motorn skickades till ROLLCO som en CAD ritning för att visa mått på axel och resten av motorn. Se figur 5.5 för en ritning på den linjära enheten som användes i projektet. CAD-modellen på banan delades ut av Rollco, som senare behandlades i Fusion 360 för att få ritningen.

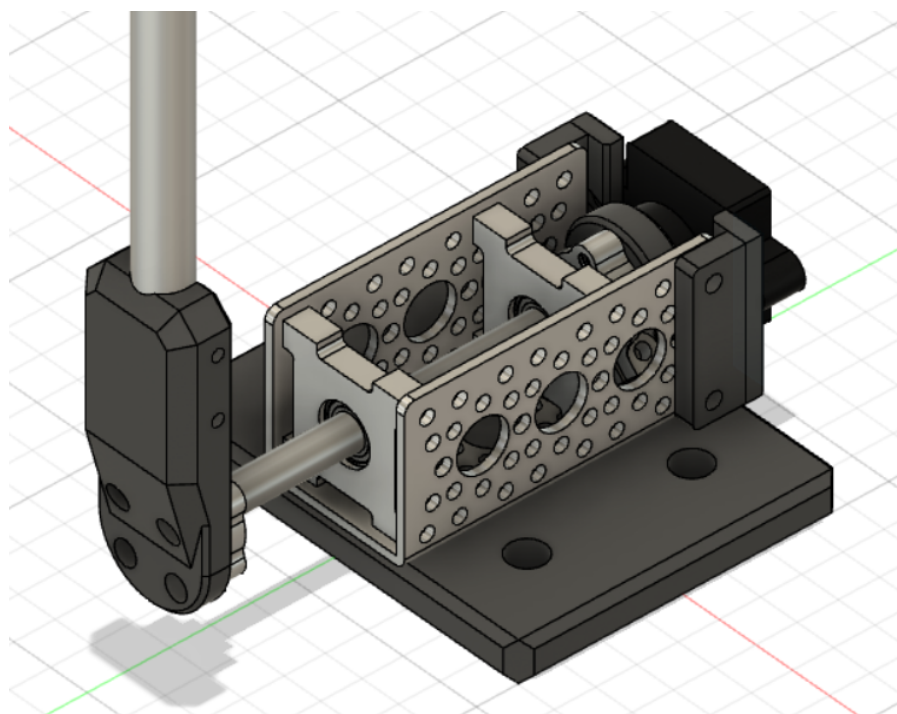


Figur 5.5. CAD ritning av linjär enhet

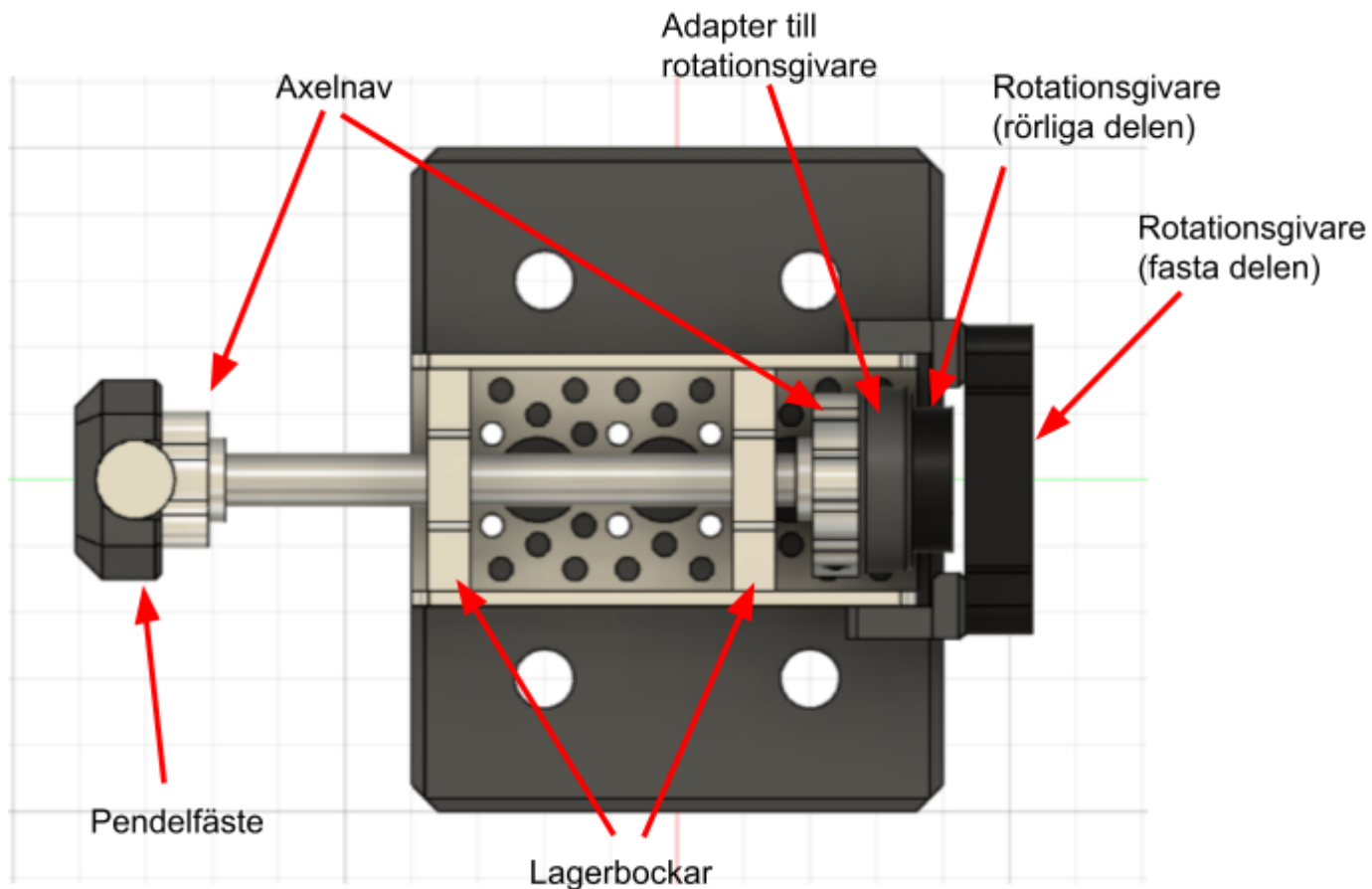
### 5.3 Vagnen

Inledningsvis undersöktes olika alternativ till hur designen av vagnen ska se ut. Flera affärer besöktes för att få inspiration till en lösning på designen av vagn samt vilka material som skulle användas. Ett aluminiumrör valdes som material till själva pendeln då det klarar kraven både viktmässigt och även ur hållfasthetsperspektiv för konstruktionen. En rimlig längd för aluminiumröret ansågs vara ca 60 cm så att pendeln kan röra sig fritt utan att slå i marken.

Kravspecifikationerna på vagnen är följande: Rotationsgivaren ska kunna sitta fast på axeln och allt ska kunna rotera obegränsat med minimal friktion, samt att hela vagnen ska vara så lätt som möjligt. För att förenkla utvecklandet av vagnen valdes det att använda delar från Actobotics sortiment av maskindelar. Då det fanns 3D-filer att hämta så laddades dessa ner, då det gav möjlighet att sätta ihop allt i Fusion 360. Detta gav den bild av hur allt skulle passa, samt gjorde det enklare att utveckla vissa delar för 3D-utskrivning. Se figur 5.6 och 5.7 för CAD-ritningen av vagnen.

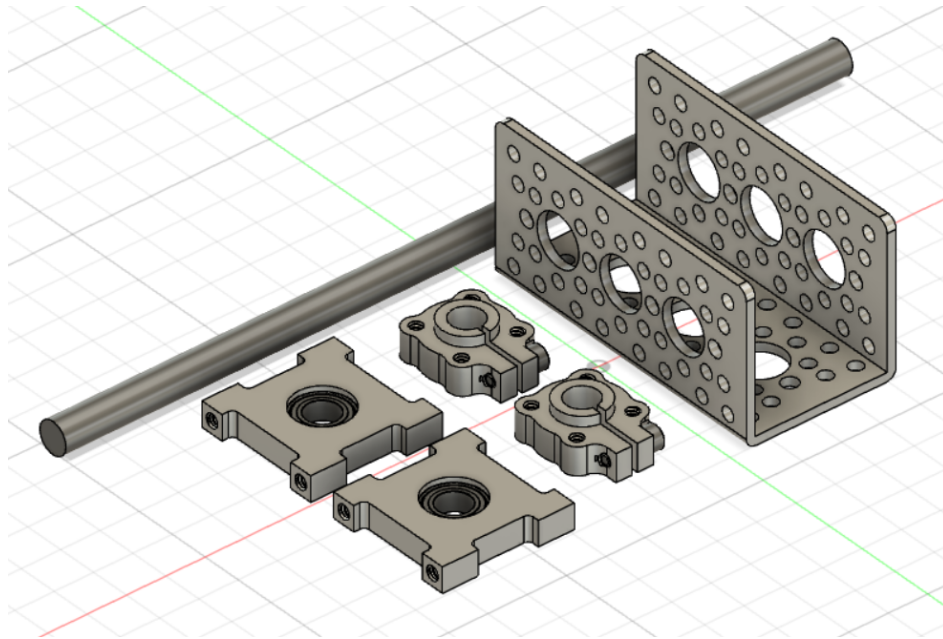


Figur 5.6. CAD-ritning av vagnen med alla delar monterade

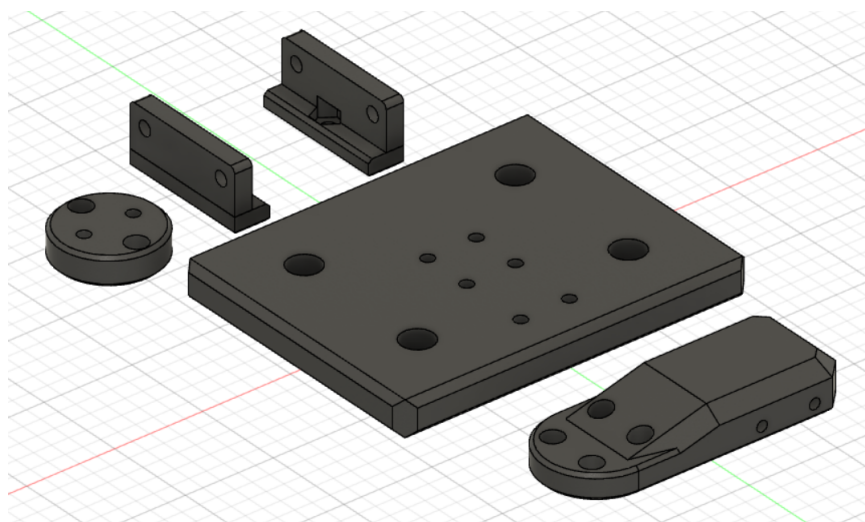


Figur 5.7. CAD-ritning på vagnen ovanifrån, med beskrivande pilar.

Av Actobotics delar så valdes det långa aluminiumchassit och två lagerbockar för att kunna hålla uppe pendeln även om axeln sticker ut en bit från vagnen enligt figur 5.8. För att enkelt kunna ta loss delarna från axeln används nav med klämfäste och skruvar. Övriga delar, såsom pendelfäste, givarfästen och plattan mellan banan och vagnen är modellerade i Fusion 360 och 3D-utskrivna för att kunna sätta ihop saker som egentligen inte passar varandra. Se figur 5.9 för CAD-ritning av de utskrivna delarna. För att minimera risken för slitage på delarna som är utskrivna anpassades de så att allting sitter fast med metallmuttrar.



Figur 5.8. CAD-ritningar av delar från Actobotics.



Figur 5.9. CAD-ritningar av 3D-utskrivna delar.

## 5.4 PLC enhet

PLC-systemet med tillhörande moduler har i första hand valts i syfte att visa vad Siemens produkter kan göra, och hur man arbetar med dem. Med denna begränsning valdes PLC:n SIMATIC S7-1500 som styrenhet.

## 5.5 I/O Moduler

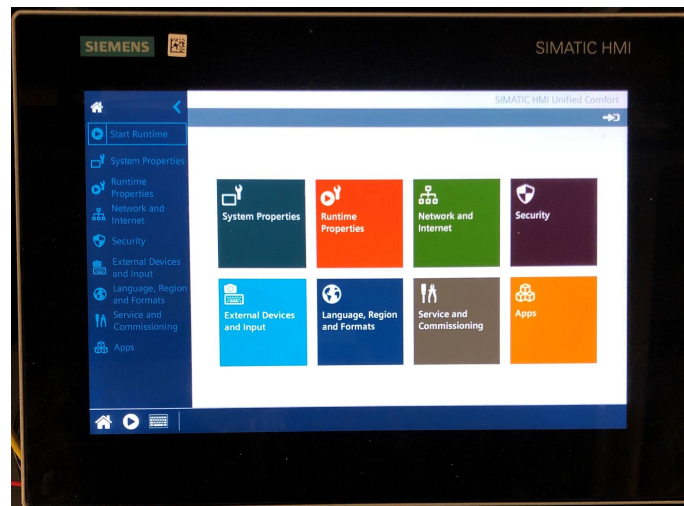
Vidare gjordes det en grundläggande lista på I/O-portar som behövdes för systemet för att ta reda på hur många och vilka typer av signaler som ska gå till och från PLC:n, och med denna listan beställdes passande moduler. Modulerna som beställdes möjliggör kontakt med olika givare och sensorer som används i projektet. I/O-modulerna som används i projektet består av 2 digitala ingångar där ena modulen har stöd för profisafe, en digital utgång och en analog ingång. Se figur 5.10 för en bild på PLC-enheten med tillhörande kort samt I/O modul.



Figur 5.10. SIMATIC S7-1500 med I/O modul och tillhörande kort

## 5.6 HMI enhet

I projektet valdes Siemens nyutkomna Simatic Unified Comfort Panel som skärm. Valet gjordes i syfte att testa den nya skärmen inför användning i framtida projekt. Det ansågs vara gynnsamt att kunna styra processen via en skärm då anläggningen ska visas upp på mässor. Huvudmenyn på HMI-skärmen visas i figur 5.11.



Figur 5.11. Huvudmeny på SIMATIC HMI

## 5.7 Rotationsgivare

Vidare i projektet diskuterades val av en rotationsgivare för att kunna mäta vilken vinkel som pendeln har. Efter att ha undersökt flera hemsidor som säljer elektronikdelar ansågs en 360 graders rotationsgivare från ELFA vara rätt val till arbetet då den klarar av att ange vinkeln med hög noggrannhet samt har lång livslängd och rimligt pris. Rotationsgivaren har en utgångstyp mellan 4 - 20 mA vilket passade utmärkt med resten av utrustningen.



Figur 5.12. Rotationsgivare

## 5.8 Ändlägesgivare

För att kunna kalibrera position och varna för riskabel position användes två ändlägesgivare. Dessa valdes till att vara induktiva givare. En anledning till detta är att Midroc Automation använder oftast induktiva givare i sina projekt och det ansågs vara viktigt att använda denna typ av givare då det kommer att visas upp i mässor. Dessa givare klarar även av projektets tekniska krav.



Figur 5.13. Ändlägesgivare

## 5.9 Nätaggregat

För att förse elektroniken i projektet med ström var det nödvändigt att beställa ett nätaggregat. Nätaggregatet som användes i projektet beställdes i samråd med handledarna på Midroc Automation. Det maximala strömmen som utrustningen kan komma att behöva är 30 Ampere och nätaggregatet som används klarar av 40 Ampere men eftersom det inte fanns en enhet med 30 Ampere som maxgräns beställdes 40 Ampere nätaggregat. Se figur 5.14 för en bild på nätaggregat som används i projektet.



Figur 5.14. Nätaggregat 40A

## 5.10 Nätverksswitch

I projektet användes flera nätverkskablar mellan de olika enheterna. I samråd med handledarna beställdes en nätverksswitch för att koppla samman alla nätverkskablar. Nätverkskabeln mellan PLC enheten och motordrivaren valdes att vara direktkopplat för snabbare kommunikation då nätverksswitchen saknar stöd för IRT.



Figur 5.15. Nätverksswitch med inkopplade Ethernetkablar

## 6 Programmering

Detta kapitel handlar om projektets programmering, samt hur regleringen av pendeln fungerar. All logik programmeras i Siemens TIA portal.

### 6.1 Programmets struktur

Detta projektet har delats upp i olika funktioner. Dessa funktioner sammanfattas i tabell 6.1. Funktionsblocken skapar automatiskt egna datablock vid skapande, men dessa tas inte med i tabellen.

Tabell 6.1. Lista på delprogram i TIA portal.

Namn	Typ	Språk	Beskrivning
Main	Organisation Block (Main)	FBD	Huvudprogrammet. Alla andra funktioner läggs in här.
Motor_Control_FC	Function	FBD	Kontroll av motor och banans hemposition.
Calibration_Process_FB	Function block	GRAPH	Innehåller sekvensen som utgör kalibrering av vinkel och position.
Angle_Calibration_FB	Function block	SCL	Kalibrerar vinkel vid signal.
Motor_Velocity_Chooser_FB	Function block	SCL	Tar in signaler och motorhastigheter från olika källor och skickar vidare det som ska användas.
PID_Control_And_Rotation_Sensor_OB	Organisation block (Cyclic interrupt)	FBD	Inställd på att uppdateras var 2:a millisekund. Innehåller alla PID-regulatorer och beräknar alla styrsignaler. Dessutom behandlas datan från rotationsgivaren här.
PID_Percentage_To_Control_Signal_FB	Function block	SCL	Omvandlar PID-regulatorernas utsignal till en passande form.
Motor_Control_DB	Data block	-	Innehåller variabler som tillhör Motor_Control_FC
Rotation_Sensor_DB	Data block	-	Innehåller variabler som tillhör PID_Control_And_Rotation_Sensor_OB som har med rotationssensorn att göra
PID_Control_DB	Data block	-	Innehåller variabler som tillhör PID_Control_And_Rotation_Sensor_OB som har med PID-regleringen att göra

Utöver funktionerna och funktionsblocken så skapades det även teknologiobjekt. Dessa är färdiga objekt som finns att välja i TIA portal, och kan hjälpa till med specifika användningsområden. Efter att teknologiobjekten har skapats används de via färdiga funktionsblock, och det finns menyer för inställningar av parametrar i objektet. Se tabell 6.2 nedan för en lista på vilka objekt som används i projektet.

Tabell 6.2. Lista på teknologiobjekt

Namn	Användning
Positioning_Axis	Utför beräkningar kring den linjära banan och kontrollerar motorn.
PID_Compact_Angle	PID-Regulator för reglering av pendelns vinkel.
PID_Compact_Position	PID-Regulator för reglering av vagnens position.

## 6.2 Viktiga variabler

Viktiga variabler och signaler sammanfattas här. De viktigaste värdena som är relevanta i detta projektet är pendelns vinkel, vagnens position och ändlägesgivarna. Se tabell 6.3 för mer detaljer. Dessa värden kan kallas för de “yttre” variablerna, då de är värden som kan observeras i verkligheten. Till skillnad från de ovannämnda variablerna finns det “inre” variabler, som då istället används i programmet och måste visas via datorn eller HMI:t.

Tabell 6.3. Lista på viktiga variabler.

	Värde från	Värde till
Rotationsgivare (analog ingång)	0	27648
Pindelns vinkel	0°	360°
Vagnens position	50 mm	950 mm
Induktiv ändlägesgivare NO (vänster)	0 (false)	1 (true)
Induktiv ändlägesgivare NO (höger)	0 (false)	1 (true)

För att förtydliga informationen i tabell 6.1 så ger rotationsgivaren en analog insignal i form av ett heltal. Signalen har då i programvaran ett värde mellan 0 och 27648 som då motsvarar ett helt varv för givaren. Värdet omvandlas till grader som beskriver pendelns vinkel genom multiplikation med en faktor. För att förenkla regleringen valdes det att pendelns vinkel ska vara 0° då den hänger rakt ner mot golvet, och 180° då den står rakt upp. Anledningen till det gjorda valet är för att skillnaden mellan ärvärdet och börvärdet ska vara enkel att använda. Om det är 0° vid stående position så kommer felet att vara 1 vid medsols vridning men 359 vid motsols vridning. Men istället kommer det i detta projektet bli fel som går från 0 åt båda hållen, alltså 1 vid medsols vridning och -1 vid motsols vridning.

Vagnens position beskriver var någonstans vagnen ligger längs med banan, räknat i millimeter från den vänstra kanten. Anledningen till att värdet går från 50 mm till 950 mm är för att vagnen är 100 mm lång, och positionen räknas från vagnens mittpunkt. För att få fram detta används det ett teknologiobjekt i TIA portal som kallas "Positioning Axis". I "positioning axis" går det att välja vilken typ av axel som motorn är kopplad till, och vilken motor/drivare som ska styras. För att kunna få ett pålitligt värde på positionen ställs det in hur långt vagnen färdas på ett varv med drivaxeln. Efter detta går det att hämta variabeln "Actual\_Position" som (efter inställning) ger det aktuella värdet i mm. När teknologiobjektet är inställt går motorn att styra med färdiga funktionsblock som kopplas till Position\_Axis. Dessa funktionsblock kan köra motorn åt båda hållen, ställa in hempositionen och köra vagnen till en viss position med mera. Fördelen med att använda teknologiobjektet är att det går att skicka in ett önskat värde på vagnens linjära hastighet, så extra beräkningar för att ta reda på nödvändig rotationshastighet på motorn för att få en viss linjär hastighet behövs inte.

Ändlägesgivarna har främst två uppgifter: Att hjälpa till med kalibrering av vagnens position och att stoppa vagnen ifall den åker för långt åt något håll. Ifall en ändlägesgivare aktiveras stoppas motorn och användaren måste kvittera ett fel i HMI:t för att kunna starta motorn igen.

### 6.3 Specifika inställningar för motordrivaren

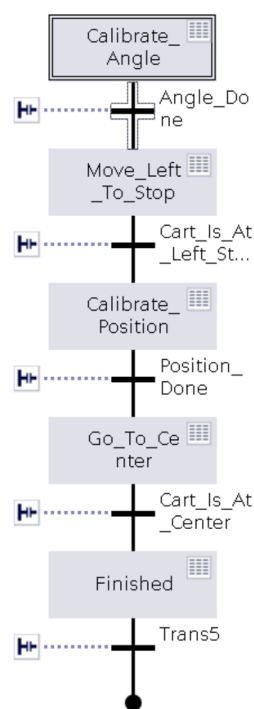
För att möjliggöra reglering av pendeln behöver systemets olika parametrar ställas in. För fallet med en balanserande pendel så behöver regleringsprocessen vara så snabb som möjligt. En av processens största "flaskhalsar", som påverkar dödtiden i regleringen, visade sig vara motorns begränsningar på acceleration samt dess uppdateringsfrekvens.

Via kontakt med Siemens support ställdes motorns uppdateringshastighet in med IRT-koppling. Detta möjliggör en konstant uppdateringsfrekvens så att det går att lita på att motorn får nya signaler med ett bestämt tidsintervall. För att åstadkomma detta sitter motordrivaren direkt kopplad med nätverkskabel till PLC:n, detta för att nätverksswitchen som används inte klarar av kommunikation med IRT. Uppdateringshastigheten för motorn som används i projektet är ställd till 4 millisekunder, vilket är den minsta uppdateringshastigheten som motorn ska klara av enligt supporten. Men även om motorn uppdateras med en hög hastighet kommer den inte garanterat att hinna upp i önskad hastighet.

För att åtgärda felet ökades accelerationen och decelerationen i inställningarna inuti teknologiojektet Positioning Axis som styr motorn. Detta visade sig att vara den allra största begränsningen på systemet, då vagnen sällan kom upp i hastigheter höga nog att reglera pendeln med standardaccelerationen.

## 6.4 Kalibrering av givare och position

Innan automatisk reglering behöver rotationsgivaren och vagnens position kalibreras, för att mätvärdena ska vara pålitliga. För att åtgärda detta skapades det ett funktionsblock med en sekvens (GRAPH i TIA portal) som aktiveras från HMI:t. Sekvensen börjar med att kalibrera rotationsgivaren, och sedan positionen. Se figur 6.1 för visuell förklaring.



Figur 6.1. Flödesschema för kalibreringen

Eftersom rotationsgivaren inte kan garanteras att den sitter på en perfekt position på axeln så valdes det att den ska kalibrera innan användning. Detta åstadkoms genom att ta det aktuella värdet som givaren ger när pendeln hänger stilla rakt mot golvet, och subtrahera givarsignalen med det. Förutom detta så behövs det justeras för negativa tal, ifall givarsignalen minus det korrigerande värdet blir mindre än 0. Vid det fallet adderas det negativa värdet med signalens maxvärde (se pseudokod nedan för hur det fungerar).

**Om** (givarsignal - korrigerande värde) är mindre än 0:  
*Kalibrerad signal = givarsignal - korrigerande värde + 27648*

**Annars:**  
*Kalibrerad signal = givarsignal - korrigerande värde*

För att kalibrera vagnens position körs den till vänster i en långsam hastighet tills den aktiverar den vänstra ändlägesgivaren. Vid denna punkten aktiveras en ingång i ett funktionsblock för motordrivaren som sätter vagnens position till ett förinställt värde som baseras på givarens position. Detta värdet måste ställas in manuellt då givaren kan justeras.

## 6.5 Start av reglering

Enligt målet ska pendeln börja reglera sin position efter att den har fått hjälp att börja stå upprätt. Detta åstadkoms genom att ha ett antal villkor för att starta reglering, samt andra villkor för att avbryta den. För att starta regleringen ska först knappen tryckas in på HMI:t, efter att kalibrering har avslutats. Sedan ska pendeln manuellt lyftas upp så att den ligger inom 2° från rakt stående position. Detta aktiverar en vipa som då startar regleringen av pendeln. Ifall pendeln ramlar ner över/under en viss vinkel så kommer vippan att nollställas och regleringen avslutas. Det går även att avbryta regleringen med en knapp på HMI:t.

## 6.6 Programmets viktiga funktioner

I denna delen kommer de viktigaste och mest relevanta delarna av programmet att beskrivas.

### 6.6.1 Motor\_Control\_FC

Funktionen Motor\_Control\_FC styr allt som har med motorn att göra. Motorn kontrolleras av SIMATIC motor drive, som i sin tur styrs av teknologiobjektet Positioning\_Axis, som får signaler från färdiga funktionsblock i TIA portal. Funktionsblocken är färdiga och ligger under "Motion Control" och kopplas till relevant teknologiobjekt i koden. Alla dessa funktionsblock börjar med "MC\_" och beskrivs kort nedan.

MC\_POWER är ett funktionsblock som startar och stoppar motordriften om ingången "Enable" är aktiv. Då motordriften är på kan de andra funktionsblocken ge signaler till motordrivaren för att köra motorn. Om driften är aktiv kommer även motorn att försöka behålla sin plats på banan, vilket betyder att den inte går att flytta för hand. Ifall säkerhetsstoppet aktiveras, alltså om vagnen befinner sig för långt ut och har aktiverat en ändlägesgivare, så är det här motorn stängs av.

MC\_RESET nollställer felkoder som skickas från de olika MC-blocken. Den vanligaste källan till felkoder är att motorns maximala moment har överskridits.

MC\_MOVEJOG är det viktigaste funktionsblocket för styrningen av motorn. Detta har två ingångar som avgör om motorn ska åka framåt eller bakåt på banan och en ingång där önskad linjär hastighet skickas in. Då motorn kan ta emot hastigheter och färdriktningar från flera olika ställen, som till exempel reglerad drift eller kalibrering, finns det ett egenskapat funktionsblock Motor\_Velocity\_Chooser\_FB som skickar vidare önskad hastighet från rätt källa. Dessutom skickar det ut rätt signal för motorns riktning baserat på om hastigheten är

negativ eller positiv. Signalerna som väljer motorns riktning kan stoppas ifall vagnen befinner sig över eller under definierade gränser som sätts i programmet. Alltså så stoppas motorn från att köra vagnen åt höger ifall motorn är över gränsen nära banans slut, likaså när motorn kör åt vänster vid banans start.

MC\_HOME är ett funktionsblock som kan återställa vagnens position till ett önskat värde. Detta används i kalibreringsfunktionen då värdet sätts till den positionen som den induktiva givaren sitter på.

### 6.6.2 PID\_Control\_And\_Rotation\_Sensor\_OB

Funktionen PID\_Control\_And\_Rotation\_Sensor\_OB innehåller all logik som har med PID-kontrollerna och rotationssensorn att göra. Anledningen till att organisationsblocket "Cyclic Interrupt" används är för att det går att ställa in önskat tidsintervall som blocket körs. Även om programmets totala cykeltid inte nödvändigtvis är för lång så används Cyclic interrupt för att det ska kunna garanteras att koden inuti exekveras med jämna mellanrum. Detta är varför både PID-regulatorerna och rotationssensorn finns inne i samma funktion. Datan från rotationssensorn ska kunna behandlas i samma vända som den används i regulatorn.

I programmet används ett tidsintervall för organisationsblocket på 2 millisekunder. Det inlagda tidsintervallet på 2 millisekunder är för att det ska köras dubbelt så många gånger som PID-regulatorernas samplingstid, som ligger på 4 millisekunder [15].

## 6.7 PID-reglering

Regulatorerna som används i projektet är skapade från en typ av teknologiobjekt i TIA portal som kallas PID\_Compact. I teknologiojektet PID\_Compact går det att ställa in önskade insignaler, utsignaler och parametrar.

### 6.7.1 Kaskadreglering

För att reglera pendeln enligt målet kommer det att behövas två regulatorer, en för vinkeln och en för positionen. Regulatorer för vinkel och position valdes att kopplas ihop tillsammans till en kaskadkontroller [16]. I projektet skapas två PID-regulatorer: en för reglering av vinkel och en för reglering av position.

För att få kaskadregleringen att fungera skapades först regulatorn för vinkeln. Regulatorn tar in vinkeln och vinkelns börvärde och skapar en signal mellan -100% och 100%. Utsignalen omvandlas sedan till ett värde mellan -2000 mm/s och 2000 mm/s via ett funktionsblock. hastigheten går sedan till motorn som då arbetar för att uppnå den. Efter att blocket skapades och justerades, då det var det enda som var inkopplat, arbetade motorn så att pendeln hela tiden stod rakt upp. Det innebär i sin tur att om en yttre kraft påverkade pendeln åt ett håll så kompenserade vagnen så att den inte ramlar, men den fortsätter att åka åt samma håll tills

något påverkar den åt andra hållet. Samma effekt kan observeras i kapitel 4 (figurer 4.4 och 4.7) då vinkeln även där är det enda som regleras.

Nästa steg för att skapa kaskadregleringen är att implementera reglering av vagnens position. Regleringen görs genom att skapa en PID-regulator som tar in vagnens position tillsammans med önskat börvärde för positionen, som sedan skickar ut en signal på -100% och 100% som omvandlas till ett börvärde för vinkeln mellan  $-160^\circ$  och  $200^\circ$ . Vinkeln valdes baserat på den teoretiska max-vinkeln som definierades i början av projektet i beräkningsdelen. Detta är då börvärdet som går in i PID-regulatorn för vinkeln. Sammanfattningsvis så kommer då felvärdet för positionen att omvandlas till en önskad vinkel, som sedan omvandlas till en nödvändig hastighet.

Det går även att stänga av kaskadregleringen i programmet genom en knapp i HMI:t. Då är det ett konstant börvärde för vinkeln på  $180^\circ$  som matas in i PID för vinkeln, och positionen har inte längre någon påverkan.



Figur 6.2. Förenklad modell av kaskadkontroll i projektet.

### 6.7.2 Inställning av PID-parametrar

När alla dödtider och felmarginaler hade minskats genom justering av motorns acceleration och minimering av samplingstider så att systemet kunde reagera snabbt nog börjades justeringen av PID-parametrarna. Som nämnt i tidigare delkapitlet om kaskadreglering skapades först regleringen av vinkeln. Dessa första värden justerades på ett experimentellt och iterativt sätt, då först P-delen ökades med stadiga intervall. Det visade sig då att pendeln kan stå upp med endast P-värde på 10, om än lite ostadigt. P-delen sänktes då lite och I-tiden aktiverades med ett värde på 20. Redigeringen av PID-värden gjorde att pendeln reagerade mycket renare och inte gjorde mycket översväng, men det märktes att det gick att sänka I-tiden för att få mer effekt. Även D-delen kopplades in, med låga värden, men denna gjorde processen oftast ojämn. Eftersom även låga värden på D-delen orsakade våldsamma rörelser valdes det att inte aktivera den delen, så värdet ställdes till 0.

När regleringen för vinkeln var inställd så var det dags för positionen. Då dess utsignal har ett mycket lägre spann med värden jämfört med vinkeln regulator så antogs det att det skulle behövas mycket mindre värden, speciellt för P-delen. När positionen skulle regleras så var det lättare att se hur vagnen betedde sig med översväng och förändringar på börvärde. Ungefär samma procedur användes för att få fram PID-parametrarna för denna delen, med skillnaden att vikten för D-delen vid börvärdesändringar avaktiverades.

Framtagna PID-värden visas i tabell 6.4.

Tabell 6.4. Värden på PID-parametrar.

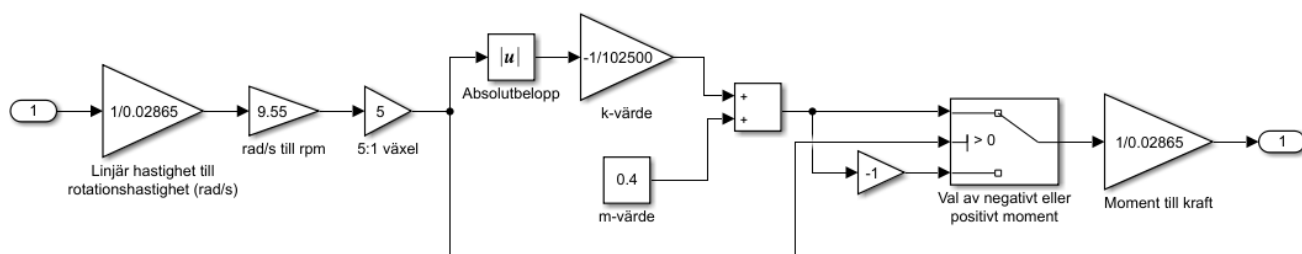
Regulator	P	I	D
PID_Compact_Angle	6	10	0
PID_Compact_Position	0,1	2	0,8

### 6.7.3 Teoretiskt test av PID-värden

Efter de praktiska värdena tagits fram gjordes en undersökning i Simulink om det går att simulera systemet och hitta optimeringar. För detta används samma tillståndsmo- dell som användes vid beräkningen tidigare i projektet. I TIA portal så använder sig Positioning\_Axis och motordrivaren sig av egna regulatorer för att köra motorn i rätt hastighet.

Positioning\_Axis har en begränsning på maximal acceleration och motordrivaren har egna inställningar på hur snabbt motorn reagerar på signaler. För att åstadkomma simuleringen antogs det att motordrivaren och Positioning\_Axis egna regulatorer perfekta. Detta betyder att den önskade rotationshastigheten, som baseras på den önskade linjära hastigheten, är lika med motorns faktiska rotationshastighet. Anledningen till detta antagandet är för att det inte finns tillräckligt med information om hur regulatorerna i TIA portal som hanterar motorn fungerar. Riskerna för att återskapa dessa på ett dåligt sätt bedömdes vara större än antagandet att de är perfekta.

Eftersom tillståndsmo- dellens använder kraften på vagnen som input så behöver vagnens hastighet omvandlas. För detta syftet skapas det ett subsystem i Simulink för att dela upp systemet. Se figur 6.3 för bild på subsystemet. En sammanfattning av vad som händer i systemet är: linjär hastighet (m/s) omvandlas till rotationshastighet (rad/s) som omvandlas till ett varvtal (rpm) [17] som sedan multipliceras med 5 på grund av växeln. Det resulterande varvtalet, som är motorns egna varvtal, omvandlas till ett moment via en formel som tas fram nedan. Till sist omvandlas detta till en kraft som går in i tillståndsmo- dellens.



Figur 6.3. Bild på subsystemet som omvandlar linjär hastighet till kraft. Skapad i Simulink.

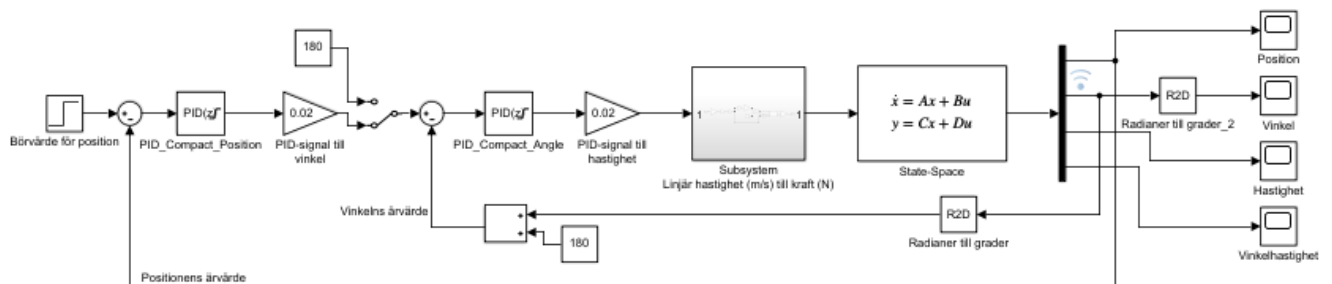
För att förtydliga i figur 6.3: insignalen, som kommer från resten av systemet, går in i 1 på vänster sida. Utsignalen går sedan efter alla beräkningar ut från 1 på höger sida och är den signalen som går vidare till resten av systemet.

Beräkningen från varvtal till kraften baseras på ett diagram i ett datablad om motorn [18]. Diagrammet, som visar motorns relation mellan varvtal och moment, har analyserats och en funktion har skapats baserat på provisoriska mätningar. Ekvation 8 visar resultatet. Funktionen följer den räta linjens ekvation  $y = kx + m$ , där  $x$  är varvtalet (rpm) och  $y$  är momentet (Nm).

$$y = 0,4 \cdot - \frac{1}{102500} \cdot x \quad (8)$$

Praktiskt sett så är inte ekvationen definierad till värden under 0 rpm, så det skapades delar i modellen som åtgärdar detta. I figur 6.3 visas det att  $k$ -värdet använder sig av absolutbeloppet av varvtalet. Sedan är det en omkoppling som inverterar resultatet ifall varvtalet är negativt.

Det hela systemet skapades sedan i Simulink, och visas i figur 6.4. För att kunna likna de verkliga PID-parametrarna med de teoretiska så anpassades signalerna från regulatorerna mellan -100% och 100%, precis som i TIA portal. Efter regulatorerna multipliceras sedan signalerna med konstanter så att de passar det relevanta värdet, alltså  $\pm 20^\circ$  för vinkeln och  $\pm 2 \text{ m/s}$  för hastigheten. Dessutom har PID-regulatorernas samplingstid ställts in till samma som används i TIA portal.



Figur 6.4. Teoretisk modell av kaskadregleringen, skapad i Simulink.

I figur 6.4 går det att se subsystemet som visas i figur 6.3, fast som endast ett block med ingång och utgång. Det kan även noteras att det går att koppla om signalen som går in i PID\_Compact\_Angle så att börvärdet är 180 konstant. Detta för att kunna simulera att kaskadkontrollen stängs av.

Målet med att skapa denna modellen var att kunna optimera PID-parametrarna som används i verkligheten, men för att kontrollera ifall modellen går att lita på ställdes de teoretiska PID-parametrarna in på samma värden som i verkligheten. Eftersom pendeln kan stå upp i verkligheten borde då modellen också uppnå balans med samma värden. Men det visade sig

att samma värden inte gav samma stabila system i simuleringen. Positionens börvärde ställdes in att vara 0, alltså helt stilla, för att ge regulatorn en chans till att stå stilla. Resultatet visas i figur 6.5.



Figur 6.5. Plot på pendelns simulerade vinkel, mätt i grader.

I figur 6.5 går det att se att vinkeln faller mot oändligheten och därmed ej längre står upp. Det är orealistiskt att den faller mot en oändligt låg vinkel, men systemet är fortfarande linjäriserat kring balanserad position så den slutar att vara definierad vid större avvikelser. Faktumet att den teoretiska pendeln inte kan stå upp, även om den startar vid balanserad position, är ett tecken på att det inte kommer att gå att använda modellen för optimering av det verkliga systemet. Även om parametrarna i regulatorerna justeras så balans nås så kommer det inte att gå att tillämpa i verkligheten.

## 7 HMI

Detta kapitel beskriver de olika skärmarna i HMI-panelen som har designats i projektet. Via diskussioner med framtida användare skapades det olika versioner av skärmarna, som till slut fick sina slutgiltiga versioner med hjälp av användarnas feedback. I detta kapitlet visas endast de färdiga versionerna som togs fram.

### 7.1 Beskrivning av startskärmen

I detta kapitel beskrivs startskärmens design och funktioner i HMI-skärmen. Se figur 7.1 för en bild på startskärmen.



Figur 7.1 Startskärm i HMI-panelen

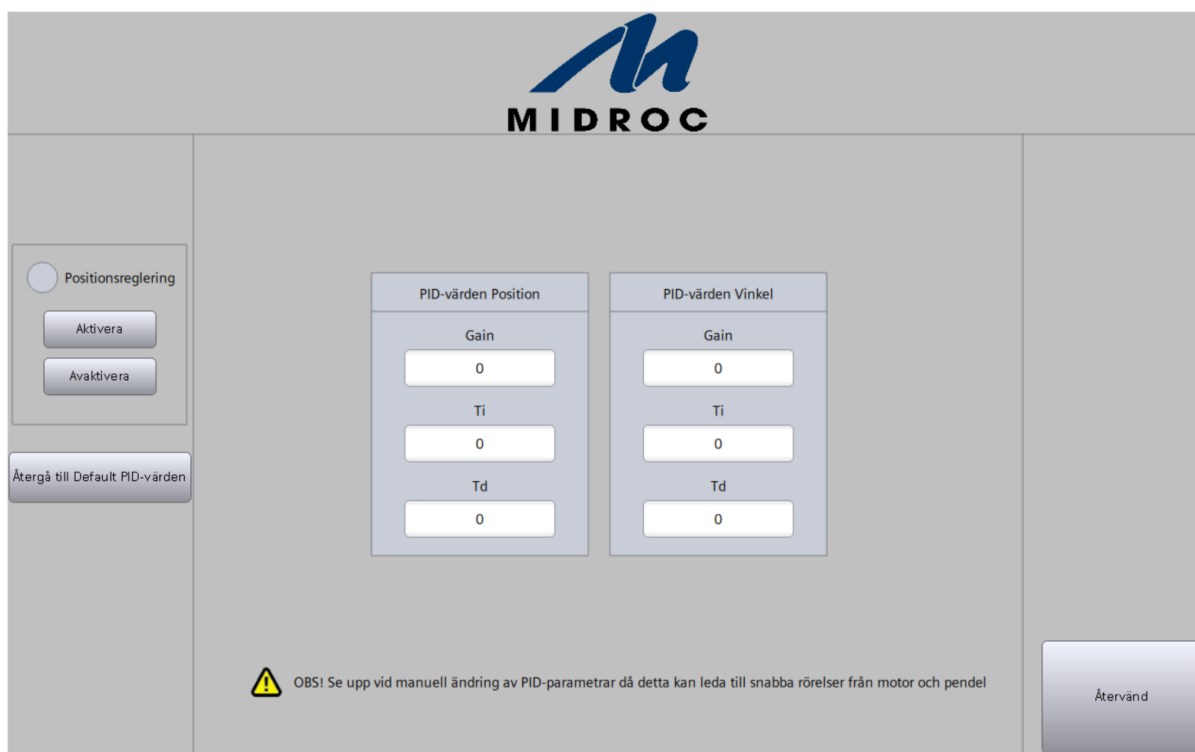
Strukturen i panelens startskärm, som visas i figur 7.1, är tänkt att vara så enkel som möjligt samtidigt som den visar all relevant info för att kunna använda maskinen. På vänster sida finns knapparna som kontrollerar kalibrering och start/stopp av reglering, och på höger sida finns mer allmänna menyer för vidare inställningar eller material. I mitten syns viktiga variabler, såsom positionen, vinkeln och hastigheten, samt så visas även slidern som kontrollerar vagnens position. Vid ändarna på slidern finns det två lampor som indikerar om ändlägesgivarna är aktiverade eller inte. Under slidern syns alla lampor som indikerar maskinens tillstånd och eventuella larm, samt tillhörande knappar som återställer dem.

Knapparnas funktioner i startskärmen:

- **Starta Kalibrering:** Genom att trycka på denna knapp startar man kalibreringsprocessen.
- **Förbered reglering:** Denna knapp startar regleringen, vilket innebär att lampan “väntar på upprätt position” tänds och användaren får lyfta pendeln till upprätt position som då triggar regleringen
- **Stoppa reglering:** Denna knapp stoppar regleringen.
- **Manuell Kontroll:** Genom att trycka på denna knapp öppnar man en ny sida där vagnen manuellt via knapptryckning “medsols” och “motsols” kan köras fram och tillbaka.
- **PID-inställningar:** Denna knapp öppnar en ny sida för användaren där man kan ställa in andra PID-värden för att testa regleringen. Se figur 7.2 för skärm med PID-inställningar.
- **Extramaterial:** Denna sidan är tom, men den finns för att kunna få plats med framtida material som antingen är bra info till användaren, eller intressant info till gäster som kollar på mässorna.
- **Återställ säkerhetsstopp:** Denna knapp återställer säkerhetsstoppet som tänds då vagnen når någon av ändlägesgivarna.
- **Acknowledge Errors:** Återställer eventuella errors i motorobjektet.

## 7.2 Beskrivning av skärm med PID-inställningar

Detta kapitel beskriver skärmen med PID-inställningar. Se figur 7.2 för en bild på skärmen.



Figur 7.2 Skärm med PID-inställningar

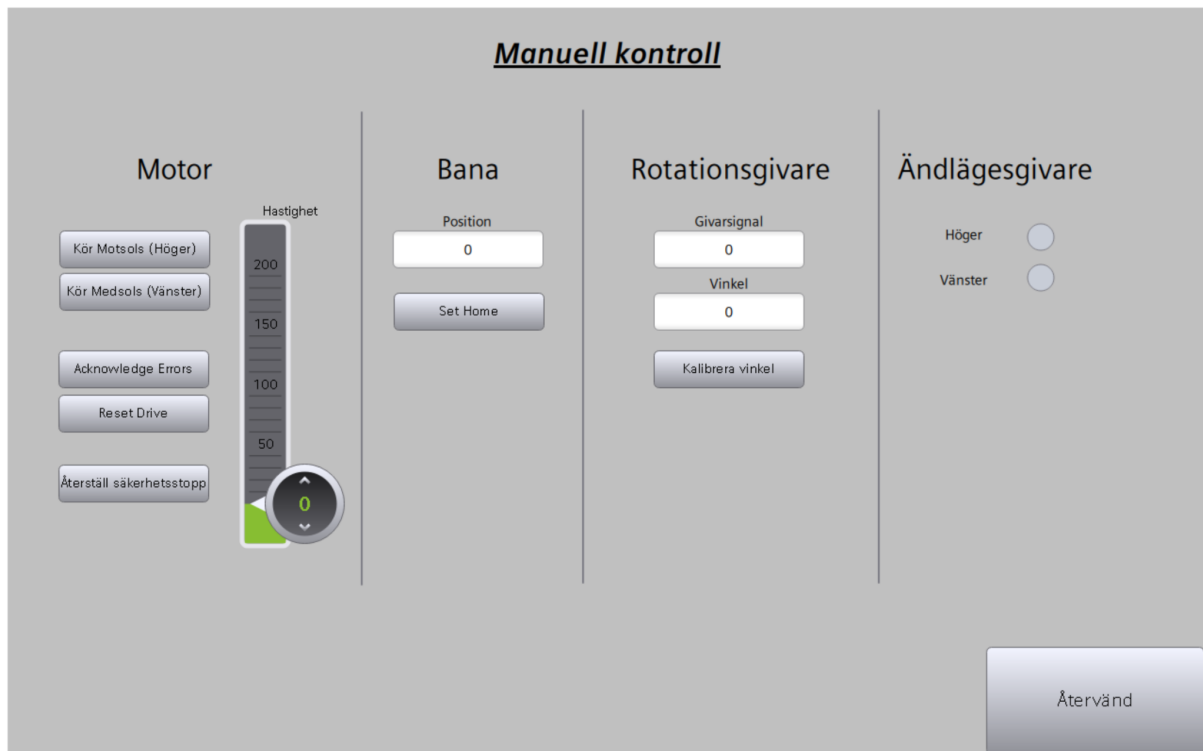
Valet gjordes att flytta allt som har med inställningarna kring PID-reglering till en annan sida för att hålla startsidan relativt enkel. Skärmen för PID-inställningar, som visas i figur 7.2, följer ungefär samma övergripande design på skärmen som startskärmen för att göra användandet lite enklare. Framför allt är det loggan och strecken som följer samma mått. Det går att ändra på PID-inställningarna genom att trycka på rutorna, vilket ger möjligheten att ändra på parametrarna medans pendeln håller på att balanseras. Då ändringen av PID-parametrarna kan innebära en del risker, eftersom vagnen kan komma upp i höga hastigheter och göra hårda rörelser, så lades en varning in.

Knapparnas funktioner i skärmen med PID-inställningar:

- **Positionsreglering aktivera:** Denna knapp startar kaskadregleringen som innebär att positionen regleras och pendeln kan stå i upprätt position.
- **Positionsreglering avaktivera:** Denna knapp avslutar positionsregleringen.
- **Återgå till default PID-värden:** Används för att återställa PID-värden till standardvärden för en fungerande reglering.

## 7.3 Beskrivning av skärm för manuell styrning

Detta kapitel går igenom skärmen för manuell kontroll, som visas i figur 7.3.



Figur 7.3. Skärm för manuell kontroll

Skärmen för manuell kontroll, som visas i figur 7.3, var egentligen den första skärmen som skapades. Detta gav möjligheten till att testa motorn och se alla variabler. När resten av systemet hade byggts upp gjordes valet att behålla skärmen, eftersom det kan vara praktiskt att kunna köra motorn manuellt.

Knapparnas funktioner i skärmen för manuell kontroll:

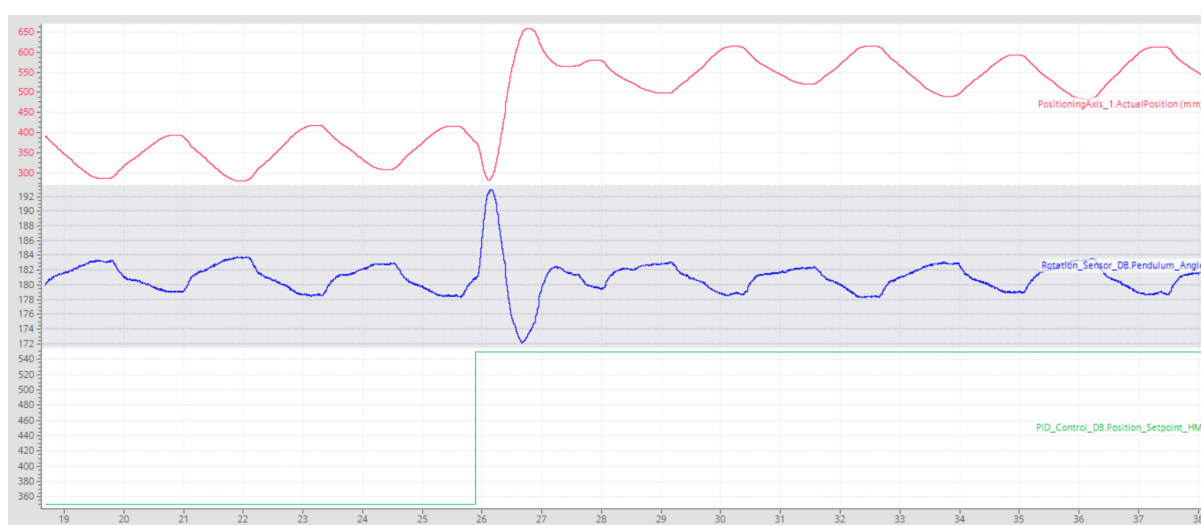
- **Kör Motsols (Höger):** Kör vagnen åt höger.
- **Kör Medsols (Vänster):** Kör vagnen åt vänster.
- **Hastighet (slider):** Ställer in motorns manuella hastighet.
- **Acknowledge Errors:** Återställer eventuella errors i motorobjektet.
- **Reset Drive:** Startar om motordrivaren.
- **Återställ säkerhetsstopp:** Denna knapp återställer säkerhetsstoppet som tänds då vagnen når någon av ändlägesgivarna.
- **Set Home:** Sätter vagnens aktuella position till 0.
- **Kalibrera vinkel:** Sätter pendelns aktuella vinkel till 0.

## 8 Resultat

Den slutgiltiga produkten har reglerats framgångsrikt och pendeln kan stå i upprätt position genom kaskadreglering. Positionen kan ändras via HMI-skärmen och vagnen förflyttar sig åt sidorna utan att pendeln faller vilket var det huvudsakliga målet med projektet.

Kommunikationen mellan hårdvaran var snabb nog för att hinna med processen och den hårdvara som valdes klarade av processens krav utan större hinder.

För att presentera resultatet av pendeln gjordes det en mätning i TIA portal med en inbyggd övervakningsfunktion som kallas Trace. För demonstration visas först några sekunder av positionshållning, och sedan en börvärdesändring av positionen. Kurvorna visas i figur 8.1 nedan.



Figur 8.1. Skärmdump på mätvärden från TIA portal.

I figur 8.1 motsvarar röd kurva vagnens position, blå kurva pendelns vinkel och grön kurva positionens börvärde. Det går att se att vid tidpunkten av börvärdesändringar så åker vagnen lite åt “fel” håll, för att sedan snabbt åka åt “rätt” håll. Detta är troligtvis en effekt av hur kaskadregleringen fungerar, alltså att den ger en önskad vinkel för att kunna åka mot sitt mål. När den sedan närmar sig sin destination gör den ett lite extra stort översväng som hjälper till att bromsa ner pendeln så att den inte faller.

## 8.1 Svar på frågeställningar

### **Kommer PLC:n att hinna reglera processen, eller behövs något snabbare?**

Det huvudsakliga målet med projektet var att undersöka om det är möjligt att reglera en instabil process som inverterad pendulum med hjälp av PLC. Programmet som gjordes i PLC:n och PID-parametrar som togs fram testades genom styrning från HMI-skärmen. Resultatet blev att kommunikationen mellan PLC och motordrivaren som användes var snabb nog och den inverterade pendeln kunde regleras till stående position genom förflyttning av vagnen. Det visades att vagnens position kan ändras utan att pendeln faller.

### **Går det att reglera en inverterad pendel längs en linjär bana med den hårdvara som Midroc Automation jobbar med?**

Målet med projektet var att konstruera en demoanläggning i form av inverterad pendulum åt företaget Midroc Automation och reglera processen med flera nya hårdvaror som företaget använder inom olika projekt. Anläggningen ska visas upp på flertal mässor och man ville visa att hårdvaran har kapacitet och kvaliteten som krävs för att reglera en sådan instabil process. Efter flera tester visade sig att hårdvara och program som Midroc Automation använder är snabba nog för att kunna reglera processen. Den färdiga anläggningen fungerar felfritt och kan visas upp på mässor där andra företag kan observera vilka hårdvaror som används i processen.

### **Är det möjligt att skapa en modell i Matlab eller liknande program för att förutspå vilka reglerparametrar som passar till den fysiska maskinen?**

Flera modeller gjordes i Matlab för att kunna ta fram PID-parametrar som fungerar för reglering av pendeln. Dessa parametrar testades sedan på den riktiga modellen men utan någon framgång och pendeln kunde inte stå i upprätt position.

## 9 Diskussion

Här kommer resultatet att diskuteras, i form av tankar, idéer och förbättringsmöjligheter.

### 9.1 Planering och arbetsupplägg

Flera av prioriteringar som gjordes under projektet kunde vi gjort på ett annorlunda sätt. Beställning av hårdvaror samt den linjära enheten borde ha prioriterats och beställts i ett tidigare skede än vad vi gjorde eftersom vissa delar har längre leveranstid då många företag är underbemannade under pandemin.

### 9.2 Siemens hårdvara

Några av hårdvaror som användes under projektet hade inte använts tidigare av Midroc som exempelvis Simatic Micro Drive och HMI Unified Comfort Panel. Bristen på erfarenhet av dessa komponenter ledde till att vi fick felsöka under uppkopplingen av HMI-skärmen och det visade sig att det var nya uppdateringar i Tia Portal som krävdes för att kunna använda skärmen. Då våra handledare inte hade arbetat med motordrivaren som används i projektet fick vi kontakta Siemens Support för att få stöd. Under tiden vi arbetade med motordrivaren upptäckte vi flera mjukvaruproblem då enheten är ny och fortfarande under utveckling.

### 9.3 Simulering av pendeln

Även om simuleringen av systemet inte fungerade så det är viktigt att reflektera över vad som gick fel. För det första gjordes det ett antagande att ignorera en del av processen kring hastigheten och motorkontrollen. Detta har som sagt oklara effekter på resultatet, men kan spela stor roll. Processerna som ignorerades är, enligt teorier, ett antal interna regulatorer (baserade på inställningar av acceleration med mera) och själva motorns egna fysiska överföring av signal till effekt. Antalet och komplexiteten av dessa tillsammans kan vara varför det inte fungerade i deras frånvaro, men av samma anledning valdes det att de inte skulle simuleras.

Det går också att diskutera omvandlingen av linjär hastighet till kraft. Även här gjordes det en del antaganden om hur allt hängde ihop. Hela det subsystemet bygger på ekvation 6.1 som härleddes från en bild, vilket i sig kan ge en stor felmarginal. Dessutom noterades det i TIA portal att det aktuella momentet som motorn använder i normal balanserande drift ligger på max 0.2 Nm ungefär. Detta moment ger stort tvivel på det teoretiska värdet som endast kan ligga mellan 0.36 Nm och 0.4 Nm rent matematiskt, inom gränserna för ekvationen som används. Anledningen till att omvandla hastigheten till momentet var för att kunna använda samma tillståndsmodell som i början av projektet. Denna ligger väl inom området för tvivel kring om det passar i detta syftet.

Vidare finns det inställningar i TIA portal som begränsar motorns moment. Detta betyder att motorn har ännu fler sofistikerade regulatorer som gör att det blir svårt att förutspå hur den ska bete sig baserat på ett ungefärligt diagram.

## 9.4 Den linjära banan

Designen av banan var nog en av projektets mest föränderliga koncept. I början var det tänkt att vi skulle bygga upp banan på egen hand, men det visade sig att det var en bra idé att beställa en färdig. Beställning av en färdig bana sparade oss mycket tid då vi troligtvis inte behövde felsöka potentiella mekaniska fel.

För det mesta har banan fungerat felfritt. Motorn passade den specialtillverkade adaptorn utan problem och inget gick sönder. Med det sagt så har det noterats en del effekter som inte är optimalt för projektet. Ett svagt gnisslande ljud kan höras när banan kör åt vänster i låga hastigheter, och det gnisslande ljudet har visat sig bero på att pendelns vagn belastar banan snett på grund av att pendeln hänger utanför. Detta pekar på att banan egentligen är designad för belastningar som ligger rakt ovanpå vagnen. I vårt fall ser det inte ut som att det är ett stort problem, då den totala massan av pendeln och vagnen är relativt små jämfört med vad som vanligtvis skulle sitta på banan.

Dessutom visade det sig att banan inte fungerar perfekt i små, precisa hastigheter. Friktionen verkar vara relativt hög, vilket gör att vagnen inte vänder och rör sig perfekt, utan "hackar" sig fram vid start. Detta problem har justerats i TIA portal genom att justera motorns acceleration och moment, men det kvarstår lite ändå. Detta är en av anledningarna till att vi accepterar att vagnen gör lite översväng kring den önskade positionen. Det ger ett "mjukare" intryck av regleringen.

## 9.5 Förbättringsmöjligheter

En tydlig utvecklingsområde för demoanläggningen är förbättring av PID-parametrar för att kunna få en ännu mjukare reglering, så långt som banan tillåter det. Mjukare reglering av pendeln kan öppna upp nya möjligheter som exempelvis reglering av pendel med ett föremål som läggs på pendeln. En annan möjlighet som kräver vidare konstruktion och kan vara mer utmanande är reglering av en dubbel inverterad pendel. Då denna process är ännu mer instabil kan det ställas hårdare krav på utrustningen som användes under detta projekt.

## 9.6 Etik och hållbarhet

Den färdiga anläggningen som redovisas på mässor kommer att visa potentialen och fördelarna med komponenterna som användes i projektet. Många företag och privatpersoner från hela landet kommer att delta i dessa mässor och observera vilka komponenter som används och vad fördelen med hårdvaran som används i projektet är. Att olika företag blir inspirerade och bestämmer sig för att använda hårdvaran som användes under projektet kan i sin tur leda till en bättre infrastruktur samt processförbättringar inom de industriområden som företagen arbetar med.

Ur ett hållbarhetsperspektiv är anläggningen som konstruerades under projektet en robust konstruktion som är designad för att fungera i många år framåt. Exempelvis har induktiva givare lång livstid då givare av denna typ inte behöver vidrörelse för att registrera kontakt. Dessutom är 3D-delarna designade på ett sådant sätt att minska slitage, genom att fästa alla delar med skruv och metallmutter. En nackdel ur hållbarhetsperspektiv är materialet som används för 3D-utskrift av vissa delar som användes under projektet, men 3D-utskriften är relativt små och används inte i större skala av Midroc. Fördelen med 3D-utskrivna delar även om det skrivs ut många exemplar är att dessa delar kan anpassas efter resten av utrustningen och komma till användning. I jämförelse med inköp av färdiga delar som troligtvis inte passar ihop med resten av utrustningen och behöver därmed slängas är 3D-utskrivna delar en mer hållbar lösning.

# Referenslista

- [1] aleehk82, "Black x2 and white i2" 2010. [Elektronisk Bild]. Tillgänglig: [https://en.wikipedia.org/wiki/File:Black\\_x2\\_and\\_white\\_i2.jpg](https://en.wikipedia.org/wiki/File:Black_x2_and_white_i2.jpg) (hämtad: 2021-04-20)
- [2] K.Erickson, "Programmable Logic Controllers," IEEE Potentials, vol 15, no.1, pp. 14-17, 1996. [Online]. Tillgänglig: [https://scholarsmine.mst.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1684&context=ele\\_comeng\\_facwork](https://scholarsmine.mst.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1684&context=ele_comeng_facwork) (hämtad: 2021-04-20)
- [3] Inductive Automation, "What Is HMI?," 2018. [Online]. Tillgänglig: [What is HMI? | Inductive Automation](https://www.inductiveautomation.com/what-is-hmi/) (hämtad: 2021-05-29)
- [4] Energihandbok, "Elmotorer, Teknik och funktion,," [Online]. Tillgänglig: <https://www.energihandbok.se/elmotorer-teknik-och-funktion> (hämtad: 2021-05-29)
- [5] RS components, "What Is A Hall Effect Sensor?," [Online]. Tillgänglig: <https://se.rs-online.com/web/generalDisplay.html?id=ideas-and-advice/hall-effect-sensors-guide> (hämtad: 2021-05-29)
- [6] Carlo Gavazzi, " Så Fungerar En Induktiv Givare,," 2020. [Online]. Tillgänglig: [https://gavazzi.se/kunskapscenter/sa-fungerar-en-induktiv-givare/?gclid=Cj0KCOjwzYGGBhCTARIsAHdMTQwX2CpGYD\\_5\\_D1n\\_4XdnVZCfyBRKKn7MiLAbsu62jIWvzTYszbPZpg aAmMuEALw\\_wcB](https://gavazzi.se/kunskapscenter/sa-fungerar-en-induktiv-givare/?gclid=Cj0KCOjwzYGGBhCTARIsAHdMTQwX2CpGYD_5_D1n_4XdnVZCfyBRKKn7MiLAbsu62jIWvzTYszbPZpg aAmMuEALw_wcB) (hämtad: 2021-06-09)
- [7] K.Johan. Åhström, R .M. Murray, Feedback Systems: System Modelling, 2.uppl., Princeton, New Jersey, USA: Princeton University Press, 2009. [Online]. Tillgänglig: [https://www.cds.caltech.edu/~murray/books/AM05/pdf/am08-complete\\_22Feb09.pdf](https://www.cds.caltech.edu/~murray/books/AM05/pdf/am08-complete_22Feb09.pdf) (Hämtad: 2021-04-21)
- [8] Bertil. Thomas, "PID-reglering", I Modern Reglerteknik: 5. uppl, Stockholm, Sverige: Liber, 2016. kap. 4.7, ss. 65-66.

[9] Bertil. Thomas, "Kaskadreglering", i Modern Reglerteknik: 5. uppl, Stockholm, Sverige: Liber, 2016. kap. 12.5 , ss. 245.

[10] PROFIBUS and PROFINET international, " PROFINET systembeskrivning," 2014.[Online]. Tillgänglig:  
<https://se.profibus.com/index.php?eID=dumpFile&t=f&f=105419&token=3a6730ee99e822d3e2636e46ab2da6421a707e13>  
(hämtad: 2021-05-29)

[11] PROFIBUS and PROFINET international, " PROFIsafe systembeskrivning," 2008.[Online]. Tillgänglig:  
<https://se.profibus.com/index.php?eID=dumpFile&t=f&f=105414&token=68837f102e57ebf27904948e49f181bd4873ed66>  
(hämtad: 2021-05-29)

[12] Profinet University, "Isochronous Real-Time (IRT) Communication", 2021 [Online]. Tillgänglig:  
<https://profinetuniversity.com/profinet-basics/isochronous-real-time-irt-communication/>  
(hämtad: 2021-06-24)

[13] Vlada Sonic, "Inverted Pendulum Demonstrator," Bachelor Thesis,, University Of Groningen, Faculty Of Science and Engineering. Groningen, Nederländerna, 2019. [Online]. Tillgänglig: [Inverted Pendulum Demonstrator - Student Theses Faculty of Science and Engineering \(rug.nl\)](https://theses.rug.nl/)  
(hämtad: 2021-02-15)

[14] University Of Michigan, Carnegie Mellon University, University Of Detroit Mercy, "Inverted Pendulum: System Modeling", 1996. [Online]. Tillgänglig:[Control Tutorials for MATLAB and Simulink - Inverted Pendulum: System Modeling \(umich.edu\)](https://www.umich.edu/~robotics/control_tutorials/inverted_pendulum/)  
(hämtad: 2021-02-17)

[15] M.Pontin, "Virtual commissioning of a pneumatic servosystem with a PLC in MiL, SiL, and HiL," Master Thesis, Department Of Control And Computer Engineering, Politecnico Di Torino, Torino, Italien, 2018. [Online]. Tillgänglig:  
<https://webthesis.biblio.polito.it/11663/1/tesi.pdf>  
(hämtad: 2021-05-20)

[16] Siemens, "Configuring of Cascade Control", 2020. [Online]. Tillgänglig:  
[https://cache.industry.siemens.com/dl/files/319/43033319/att\\_11736/v1/43033319\\_cascadectrl\\_en.pdf](https://cache.industry.siemens.com/dl/files/319/43033319/att_11736/v1/43033319_cascadectrl_en.pdf)  
(hämtad: 2021-05-25)

[17] enheter.info, “Räkna om rad/s till Varv per minut”, 2021. [Online]. Tillgänglig: <https://www.enheter.info/Raekna+om+rad+s+till+Varv+per+minut.php> (hämtad: 2021-06-02)

[18] *ECI motors*, Georgen, Tyskland: ebm-papst. [Online]. Tillgänglig: [https://www.ebmpapst.us/media/content/products/downloads/Motors\\_ECI\\_EN.pdf](https://www.ebmpapst.us/media/content/products/downloads/Motors_ECI_EN.pdf) Hämtad: 2021-06-02

[19] William Moebs, Samuel J. Ling, Jeff Sanny, “Newton’s Second Law for Rotation”, i *University Physics Volume 1*. TX, USA: OpenStax, 2016, kap. 10.7. [Online]. Tillgänglig: <https://openstax.org/books/university-physics-volume-1/pages/10-7-newtons-second-law-for-rotation> (hämtad: 2021-02-17)

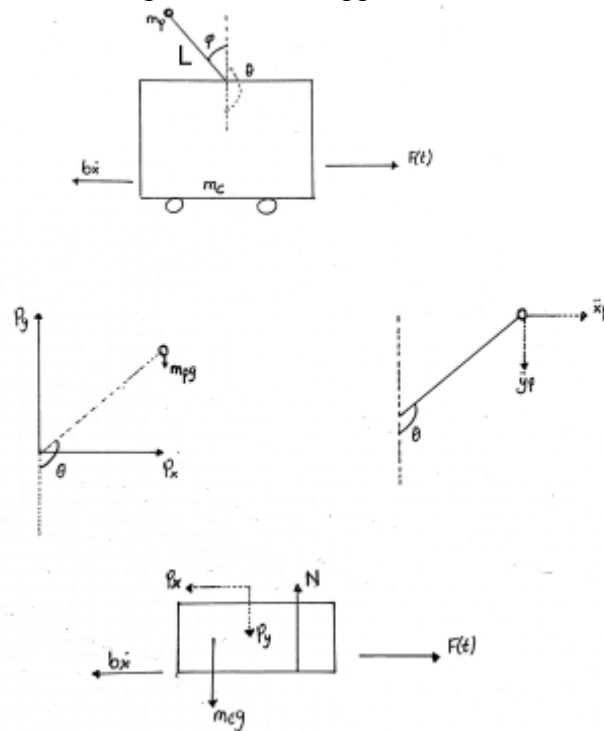
# Bilaga 1

## Beräkning av systemets mekanik

Beräkningar och ekvationer som har tagits fram i denna bilaga är inspirerade av tidigare arbeten som har behandlat liknande process [13],[14].

### Friläggning

Först delades de olika delarna av systemet upp i flera friläggningar, vilket visas i figur 1. Efter detta användes newtons andra lag för att ställa upp de första sambanden.



Figur A.1. Bild på friläggningar av systemet.

Newtons andra lag för vagnen:

$$m_c x'' = F(t) - bx' - P_x \quad (1)$$

Newtons andra lag för pendeln och dess rotation [19]:

$$I\theta'' = -y_p l \sin\theta - x_p l \cos\theta \quad (2)$$

$$m_p y_p'' = P_y - m_p g \quad (3)$$

$$m_p x_p'' = P_x \quad (4)$$

Ekvationer för pendelns position och hastighet ska nu beräknas. Detta görs genom att ta x- och y-koordinaterna vid pendelns tyngdpunkt och derivera dem tills man får hastigheten och accelerationen i båda axlarnas riktning.

Y-axeln:

$$y_p = -l \cos\theta \quad (5)$$

$$y_p' = l\theta' \sin\theta \quad (6)$$

$$y_p'' = l \sin\theta + l\theta'^2 \cos\theta \quad (7)$$

X-axeln:

$$x_p = x + l \sin\theta \quad (8)$$

$$x_p' = x' + l \cdot \theta' \cos\theta \quad (9)$$

$$x_p'' = x'' + l \cdot \theta'' \cdot \cos\theta - l \cdot \theta'^2 \sin\theta \quad (10)$$

Notera att den positiva riktningen för y i koordinatsystemet är uppåt, men eftersom vinkeln startar från den negativa biten av y-axeln så behövs det en negation för att koordinaten ska stämma. Därav är det ett minustecken i ekvation (5).

Ekvation (3) tillsammans med ekvation (7) ger:

$$P_y = m_p (g + l\theta'' \sin\theta + l\theta'^2 \cos\theta) \quad (11)$$

På samma sätt ger ekvationer (4) och (10):

$$P_x = m_p (x'' + l\theta'' \cos\theta - l\theta'^2 \sin\theta) \quad (12)$$

Ekvation (11) in i (3), samt ekvation (12) in i (4) ger då två ekvationer som innehåller alla variabler av systemet:

$$(m_c + m_p)x'' + bx' + m_p l\theta'' \cos\theta - m_p l\theta'^2 \sin\theta = F(t) \quad (13)$$

$$(I + m_p l^2)\theta'' + m_p l x'' \cos\theta + m_p g l \sin\theta = 0 \quad (14)$$

## Linjärisering

För att kunna simulera systemet så kommer det att linjäriseras runt en viss punkt. Denna punkt valdes till att bli den vinkel då pendeln står rakt upp i "balanserat" tillstånd. För att förenkla detta skapades det en till vinkel, som då går mellan y-axelns positiva riktning och pendeln. Se figur A.1 för denna vinkel. Denna vinkel får representeras av  $\varphi$ , och dessa samband uppstår vid linjäriseringen:

$$\sin\theta = \sin(\pi + \varphi) \approx -\varphi$$

$$\cos\theta = \cos(\pi + \varphi) \approx -1$$

$$\theta'^2 = \varphi'^2 \approx 0$$

$$\theta'' = \varphi''$$

Kombineras nu dessa samband med ekvationer (7) och (8) så ger det:

$$(m_c + m_p)x'' + bx' - m_p l\varphi'' = F(t) \quad (15)$$

$$(I + m_p l^2)\varphi'' - m_p lx'' - m_p gl\varphi = 0 \quad (16)$$

## Rörelseekvationer

Nästa steg är att bryta ut  $x''$  och  $\varphi''$  ur sina respektive ekvationer (15) och (16).

$$x'' = \frac{F(t) - bx' + m_p l\varphi''}{(m_c + m_p)} \quad (17)$$

$$\varphi'' = \frac{m_p lx'' + m_p l\varphi}{(I + m_p l^2)} \quad (18)$$

Sista steget är att substituera in ekvationer (17) och (18) in i varandra för att få fram rörelseekvationerna för systemet. Ekvation (17) in i (18), samt ekvation (18) in i (17) ger:

$$\varphi'' = \frac{m_p l(F(t) - bx') + m_p gl\varphi(m_c + m_p)}{m_c I + m_c m_p l^2 + m_p I} \quad (19)$$

$$x'' = \frac{(F(t) - bx')(I + m_p l^2) + m_p^2 gl^2 \varphi}{m_c I + m_c m_p l^2 + m_p I} \quad (20)$$

## Bilaga 2

Lista på hårdvara som används i projektet

Namn	Typ	Beskrivning	Märke
SIMATIC s7-1513F-1	PLC	Kontrollern som styr allt och utför alla beräkningar	Siemens
SIMATIC Memory Card	SD-kort	Minneskort för PLC:n	Siemens
SIMATIC ET 200SP IM 155-6PN ST	I/O-enhet	Enhet som I/O-modulerna sitter inkopplade i	Siemens
F-DI 8x24VDC	Digital Input-modul	Digital Input-modul med stöd för PROFI-safe.	Siemens
DI 8x24VDC	Digital Input-modul	Input-modul för digitala signaler 0-24v	Siemens
DQ 8x24VDC	Digital Output-modul	Output-modul för digitala signaler 0-24v	Siemens
AI 2xU/I 2-/4-wire HS	Analog Input-modul	Analog Input-modul med en hög avläsningshastighet.	Siemens
SIMATIC HMI Unified Comfort Panel 12"	HMI-skärm	12" HMI-skärm med touchfunktion	Siemens
PDC 600F	Motordrivare	SIMATIC micro-drive som kontrollerar motorn	Siemens
SITOP PSEU200U	Säkringar	4 st säkringar för 24v, med max 3A ström.	Siemens
SITOP PSU8200	Nätaggreat	Omvandlar 230v AC till 24v DC. Max 40A ström.	Siemens
SCALANCE XB005	Nätverksswitch	Kopplar samman alla nätverkskablar	Siemens
RHL80	Linjär bana	Linjär bana med vagn som flyttas med hjälp utav en drivaxel och kuggrem	Rollco

<b>Namn</b>	<b>Typ</b>	<b>Beskrivning</b>	<b>Märke</b>
ECI 63.40 24V med PerformaxPlus 63.1 5:1	Motor	Motor med fastmonterad växel och encoder	ebm-papst
CSD LAiO2	Motorkabel	Skärmad kabel med matning och signaler	Knorrtec
VERT-X 31E6A736-151-402	Rotationsgivare	Hallsensor som mäter vinkel med en statisk del och en rörlig del som består av en magnet.	Contelec
XS208BLPAL2	Induktiv givare	Givare som ger signal då en metall är framför den	Schneider Electric