

# MOTION CONTROL OF THE LEADING EDGE SCANNER TESTRIG

Afstudeerscriptie

Robotics & Mechatronics - Faculteit Techniek  
2019/2020



# MOTION CONTROL OF THE LEADING EDGE SCANNER TESTRIG

Afstudeerscriptie

## AUTEUR

Robin Ruys      500759066

## CONTACT

Email: [Robin.ruijs@hva.nl](mailto:Robin.ruijs@hva.nl) - [Robinruijs1@gmail.com](mailto:Robinruijs1@gmail.com)

Telefoon: +31 6 28369059

## AFDELING

Robotics & Mechatronics - Faculteit Techniek

## ORGANISATIE

NLR - Nederlands Lucht- en Ruimtevaartcentrum

## AFDELING

Avionics Systems & Maintenance Engineering

## DATUM

21 januari 2020

## TYPE PROJECT

Eindverslag

## VERSIE

3

© 2020 Copyright Hogeschool Amsterdam

Alle rechten voorbehouden. Niets uit deze uitgave mag worden veeelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door print-outs, kopieën, of op welke manier dan ook, zonder voorafgaande schriftelijke toestemming van de Hogeschool Amsterdam.

**Motion control of the leading edge scanner testrig**

Robotics & Mechatronics - Faculteit Techniek – versie 3

© 2020 Copyright Hogeschool Amsterdam

# Voorwoord

Dit verslag is een product van mijn afstudeerstage voor de opleiding engineering. Voor het zoeken naar een afstudeerstage, ben ik op zoek gegaan naar een project dat aansluit bij mijn interesses en vaardigheden. Na enige oriëntatie bij diverse bedrijven, ben ik bij het Nederlandse Lucht- en Ruimtevaartcentrum (NLR) terecht gekomen. Ondanks dat ik geen ervaring in de luchtvaart heb heeft het wel altijd mijn interesse gehad. Door verder op de website van het NLR te kijken kwam ik een project tegen dat aansluit bij mijn studie. Namelijk, het aansturen van een robotarm die de voorvleugel van een vliegtuig inspecteert.

Tijdens mijn stage heb ik veel plezier gehad in het maken en ontwerpen tijdens het project. Samen met mijn collega's zijn wij in een half jaar heel ver gekomen, en hebben wij een goede basis gelegd voor het NLR. Ik heb genoten van de vrijheid die ik kreeg tijdens het project, echter was het wennen toen de corona crisis tijdens mijn stage de vrijheid even wegnam. Gelukkig is door flexibel optreden van het NLR en door onze werkgroep weinig hinder door de crisis ondervonden.

Het meest uitdagende van mijn stage is het aansturen van de robot geweest. Gelukkig heeft het eindresultaat de verwachtingen overtroffen. Het project heeft mij uitgedaagd om mijn programmeer skills en mijn ontwerp skills naar een hoger niveau te brengen. Door de gegeven vrijheid van het NLR, heb ik mij ook organisatorisch kunnen ontwikkelen.

Ik heb het meest genoten van de goede samenwerking in onze groep. Met een goede combinatie van stagiairs en specialisten is het fijn werken in deze groep. Daarom wil ik iedereen bedanken die in deze groep meewerkt aan dit project. Graag wil ik de Arjan de Jong en Erik Konnemann voor de begeleiding en beschikbaar stellen van de opdracht. Tot slot wil ik Timo Wassenaar extra bedanken voor de goede samenwerking van het afgelopen half jaar.

# Samenvatting

Het doel van het NLR is om de lucht en ruimtevaart veiliger, duurzamer, efficiënter en effectiever te maken. Het NLR doet onderzoek naar het verbeteren en automatiseren van vliegtuig inspecties. Hiervoor is de afdeling ASAM bezig met inspecties aan een leading edge. Een leading edge is het voorste deel van een vleugel bij een vliegtuig. Bij deze leading edge worden (3D) beelden gemaakt door middel van een robot. Dit rapport bestaat uit onderzoek naar de realisatie van motion control bij deze robot. Voor het onderzoek is de volgende hoofdvraag opgesteld "Hoe kan motion control in de huidige testopstelling worden geïmplementeerd, zodat de testopstelling functioneert naar de opdrachten van de operator?". De robotarm samen met een lineaire geleider is onderdeel van een testopstelling. Deze wordt doorontwikkeld tot een opstelling die op een ideale afstand, op het normaal het oppervlak, van een leading edge kan inspecteren. Hierbij wordt door middel van grenzen en randvoorwaarden het project afgebakend.

De testopstelling bevat een 4-assige gearticuleerde robot arm. Deze robotarm beweegt zich langs de leading edge door middel van een lineaire module. Dit zorgt ervoor dat de leading edge geïnspecteerd kan worden. Dit wordt een gehele testopstelling waar de robot op een ideale manier naar de leading edge kan kijken. Deze testopstelling heeft vijf vrijheidsgraden en zal van enige veiligheidseisen voorzien moeten worden. Deze veiligheidseisen worden gespecificeerd en geïnventariseerd in een adviesrapport. Dit adviesrapport wordt aan het NLR opgeleverd.

De robot heeft een kinematische structuur waar de oriëntaties van de joints en links in beschreven staan. De kinematica modellen zijn onderverdeeld in een forward en een reverse analyse. Met de forward analyse wordt de eindpositie bepaald door elke individuele positie van de robot joints. Bij de reverse analyse wordt het tegenovergestelde gedaan, de positie van de end-effector wordt bepaald en daarmee worden de individuele posities van de robot joints berekend.

Er worden simulaties gedaan voor de kinetica. Uit het onderzoek wordt de perfecte beweging voor de testopstelling gevonden. Een perfecte beweging is de beste balans tussen snelheid, acceleratie en jerk. De acceleratie is de afgeleide van de snelheid en de jerk is de afgeleide van de acceleratie.

Om de testopstelling te realiseren zijn er ontwerpkeuzes gekozen met betrekking tot het bereik en de aansturing van de lineaire module. Voor het bereik zijn er verschillende situaties overwogen waarbij er uiteindelijk twee verbeteringen zijn gekomen. Er wordt een extra afstand tussen de leading edge en de robot gerealiseerd. Ook wordt de kinematische structuur van de robot aangepast. Voor de aansturing van de lineaire module is een drietal opties overwogen. Hiervoor is het benodigde koppel van 9.5 N/m berekend. Door middel van keuzematrix is er gekozen voor een servo en gearbox combinatie die de lineaire module zal aansturen.

Voor de software zijn er twee belangrijke processen nodig, namelijk kalibratie en positionering. Voor de positionering wordt de kinematica geïmplementeerd binnen de Twincat software. Voor het tunen van de servomotor wordt de snelheid en acceleratie ingeregeld om de position lag zo laag mogelijk te houden. Om veiligheidsmaatregelen toe te passen zijn verschillende mogelijkheden van collision avoidance meegenomen. Uiteindelijk is er voor een goal-to-goal en een gemixte strategie gekozen. Het is mogelijk om met de robot te communiceren door middel van het netwerk.

De uiteindelijke conclusie is dat alle posities behaald kunnen worden. De kinematica modellen die zijn opgesteld worden gebruikt om de robot aan te sturen. Deze modellen zijn geïmplementeerd in TwinCAT. Uit de testen en simulaties zijn alle deel concepten ook behaald.

# Abstract

NLR is conducting research into improving aviation and space travel. One of these studies is to improve inspections of aircraft. To this end, the ASAM department is conducting inspections for a leading edge. For this (3D) images are made of this leading edge by a robot. This research consists of the implementation of motion control with this robot. This research involves the main question “How can motion control be implemented in the current test setup such that the test setup functions according to the director's assignments?”. This robot is part of a test setup. This is further developed into an arrangement that can inspect the surface at an ideal distance, on the normal level. The project is defined by means of several boundaries and preconditions.

There is an articulated robot arm with a linear module that allows the leading edge to be inspected. This will be a complete test set-up where the robot can look at the leading edge in an ideal way. This test set-up has five degrees of freedom and must be provided with some safety requirements. These safety requirements are specified and inventoried in an advisory report.

The robot has a kinematic structure in which the orientations of the joints and links are described. The DH notation is derived from this kinematic structure. Furthermore, the kinematics can be divided into a forward and a reverse analysis. With the forward analysis, the end position is determined based on the position of the link. The position of the robot is determined from this reverse analysis and simulations are carried out with it. Kinetics is the doctrine of movement force. With these movement profiles the perfect movement is investigated. The perfect balance between speed, acceleration and jerk is found.

In order to realize the test setup, design choices will be needed with regard to the range and control of the linear module. For the range, several situations have been considered, with two improvements eventually emerging. An extra distance is placed between the leading edge and the robot. The last link of the robot is also being modified. Three options have been considered for controlling the linear module. The required torque of 9.5 N / m has been calculated for this. By means of a trade-off table, a servo and gearbox combination has been chosen that will control the linear module.

The software requires two main processes, calibration and positioning. For positioning, the kinematics are implanted within the TwinCAT software. Before tuning the servo motor, the speed and acceleration are adjusted to keep the position lag as low as possible. To apply safety measures, various options of collision avoidance have been included. Ultimately, a goal-to-goal and a mixed strategy were chosen. It is possible to communicate with the robot through the network.

The final conclusion is that all positions can be achieved. The kinematics models that are drawn up are used to control the robot. These models are implemented in TwinCAT. All partial concepts were also obtained from the tests and simulations.

# Inhoudsopgave

<b>1.</b>	<b>Oriëntatiefase .....</b>	<b>7</b>
1.1	Bedrijfsomschrijving .....	7
1.2	Probleemstelling en doelstelling .....	7
1.2.1	Hoofd- en deelvragen .....	7
1.3	Opdrachtoomschrijving .....	8
1.3.1	Situatie omschrijving .....	8
1.3.2	De projectscope .....	9
1.3.3	Opleverproducten .....	9
1.3.4	De activiteiten tijdens het project .....	10
<b>2.</b>	<b>Analysefase .....</b>	<b>11</b>
2.1	Grenzen en randvoorwaarden .....	11
2.1.1	MoSCoW .....	11
2.2	Detecteren van schade en defecten .....	11
2.3	De testopstelling .....	12
2.3.1	Vrijheidsgraden .....	13
2.3.2	Type robot .....	13
2.4	Veiligheid testopstelling .....	13
2.4.1	Adviesrapport veiligheid testopstelling .....	14
2.4.2	Collision avoidance .....	14
2.4.3	Force and power limitation .....	15
2.5	Kinematica .....	16
2.5.1	Kinematisch structuur .....	16
2.5.2	Danavidi-Hartenberg notatie .....	17
2.5.3	Forward kinematica .....	18
2.5.4	Reverse kinematica .....	18
2.5.5	Simulatie .....	19
2.6	Kinetica .....	20
2.6.1	De "perfecte" beweging .....	20
2.7	Bereik robotarm .....	23
2.8	Implementatie software .....	23
2.9	Aansturing lineaire module .....	23
<b>3.</b>	<b>Ontwerpfase .....</b>	<b>24</b>
3.1	Bereik van de robotarm .....	24
3.1.1	Ontwerp mogelijkheden .....	25
3.1.2	Keuzematrix voor het bereik .....	26
3.2	Aansturing lineaire module .....	26
3.2.1	Ontwerp mogelijkheden .....	28
3.2.2	Keuzematrix lineaire module .....	29

<b>4.</b>	<b>Detailleringsfase</b>	<b>30</b>
4.1	Detaillering software	30
4.1.1	Flowchart	30
4.1.2	Kinematica	31
4.1.3	Kinetica	31
4.1.4	Inregelen van de servo motor	31
4.2	Detaillering hardware	32
4.2.1	Elektrische detaillering	32
4.2.2	Mechanische detaillering	33
4.3	Test van de kinematica	33
4.3.1	Testopzet	33
4.3.2	Resultaten	34
4.3.3	Conclusie	35
4.4	Test voor de aansturing lineaire geleider	35
4.4.1	Testopzet	35
4.4.2	Resultaten	35
4.4.3	Conclusie	36
<b>5.</b>	<b>Realisatiefase</b>	<b>37</b>
5.1	Verificatie specificatie	37
5.2	Oplever producten	37
5.3	Conclusie	38
5.4	Aanbeveling	39
<b>6.</b>	<b>Literatuurlijst</b>	<b>40</b>
	<b>Bijlage A: Adviesrapport safety</b>	<b>42</b>
	<b>Bijlage B: Reverse kinematics</b>	<b>53</b>
	<b>Bijlage C: Bereik robotarm</b>	<b>57</b>
	<b>Bijlage D Software</b>	<b>64</b>
	<b>Bijlage E Plots motion</b>	<b>75</b>



# 1. Oriëntatiefase

*In dit hoofdstuk wordt het probleem en doelstelling beschreven. Hierbij zijn de hoofd- en deelvragen opgesteld. Ook wordt de gehele scope van het project weergegeven.*

## 1.1 Bedrijfsomschrijving

Het NLR is het centrale instituut voor lucht- en ruimtevaart onderzoek in Nederland [1]. Het Koninklijke NLR bestaat al meer dan 100 jaar. Het doel is om de lucht en ruimtevaart veiliger, duurzamer, efficiënter en effectiever te maken. De innovatieve oplossingen en praktische adviezen versterken de concurrentiekracht van het bedrijfsleven en dragen bij aan oplossingen voor maatschappelijke vraagstukken. NLR werkt op objectieve wijze, voor en mét het (inter)nationale bedrijfsleven en overheidsinstanties. De visie van het NLR luidt: De uitdagingen in de luchtvaart zijn altijd groter dan de mogelijkheden van vandaag. Alleen het continu verbinden van een diepgaand inzicht in de klantbehoefte met toonaangevende kennis en onderzoeksfaciliteiten maakt snel innoveren mogelijk. NLR is hier de verbindende schakel tussen wetenschap, bedrijfsleven en overheid.

## 1.2 Probleemstelling en doelstelling

De Nederlandse luchtvaart is een snel groeiende sector. Om te blijven groeien, zal de luchtvaart enkele uitdagingen tegenkomen. Voor visuele inspecties is er een schaarste aan personeel. Er komen meer vliegtuigen bij waar nog meer onderhoud voor nodig zal zijn. Vliegtuigen worden continu geïnspecteerd en onderhouden. Door de schaarste aan onderhoudstechnici wordt het moeilijk om aan de hoge veiligheidseisen te voldoen. Hierdoor zal de afdeling Avionics Systems & Maintenance Engineering (ASAM) op zoek gaan naar verbeteringen en nieuwe mogelijkheden van inspectie voor vliegtuigen. Een van de verbeteringen waar Maintenance Engineering mee bezig is de inspectie van een leading edge. Een leading edge is het voorste stuk van een vleugel. Een leading edge is vele malen gevoeliger voor beschadigingen (bijvoorbeeld: hagelschade) en wordt daarom vaker geïnspecteerd.

Het doel van NLR voor deze opdracht is: Het automatiseren van een GVI (general visual inspection), door het gebruik van een robot die de leading edge van een vliegtuigvleugel inspecteert op defecten. Het project kan in twee onderdelen worden onderverdeeld. Het eerste onderdeel is 3D imaging en detectie software. Voor dit onderwerp is een groep stagiaires bezig met onderzoek voor het detecteren en classificeren van defecten in een materiaal. Voordat de defecten kunnen worden geclassificeerd moeten er allereerst foto opnames worden gemaakt, hierbij wordt gebruik gemaakt van een stereoscoop camera die ervoor zorgt dat er een 3D beeld van het object kan worden gereproduceerd. Het tweede onderdeel van dit project is dan het positioneren van deze camera om deze 3D foto opnames te maken. Dit wordt gedaan door automatisering en robotisering(s)technieken. Dit rapport focust zich op het tweede onderdeel. Het gaat over het ontwerp van de motion control voor een testopstelling die de leading edge kan inspecteren.

### 1.2.1 Hoofd- en deelvragen

Na onderzoek naar het project en overleg met de opdrachtgever is de hoofdvraag geformuleerd:

*“Hoe kan motion control in de huidige testopstelling worden geïmplementeerd zodat de testopstelling functioneert naar de opdrachten van de operator?”*

De bijbehorende deelvragen van deze hoofdvraag luiden als volgt:

- (i) Wat zijn de eisen, doelen en randvoorwaarden van het project?



- (ii) Hoe ziet de huidige testopstelling eruit en welke onderdelen zullen gebruikt moeten worden?
- (iii) Welke manieren van “collision avoidance” en andere safety aspecten zijn er, en hoe kunnen deze worden geïmplementeerd in het huidige systeem?
- (iv) Welke kinematica modellen moeten worden opgesteld voor de aansturing van de testopstelling?
- (v) Wat is de optimale kinematische structuur waarbij de leading edge het best geïnspecteerd kan worden.
- (vi) Hoe kunnen de kinematica modellen worden geïmplementeerd binnen de Beckhoff TwinCAT omgeving?
- (vii) Wat is een goed kinetica model waardoor visuele inspecties van de leading edge kan worden uitgevoerd?
- (viii) Welke aandrijving voor de lineaire module is het meest geschikt in een verticale testopstelling?

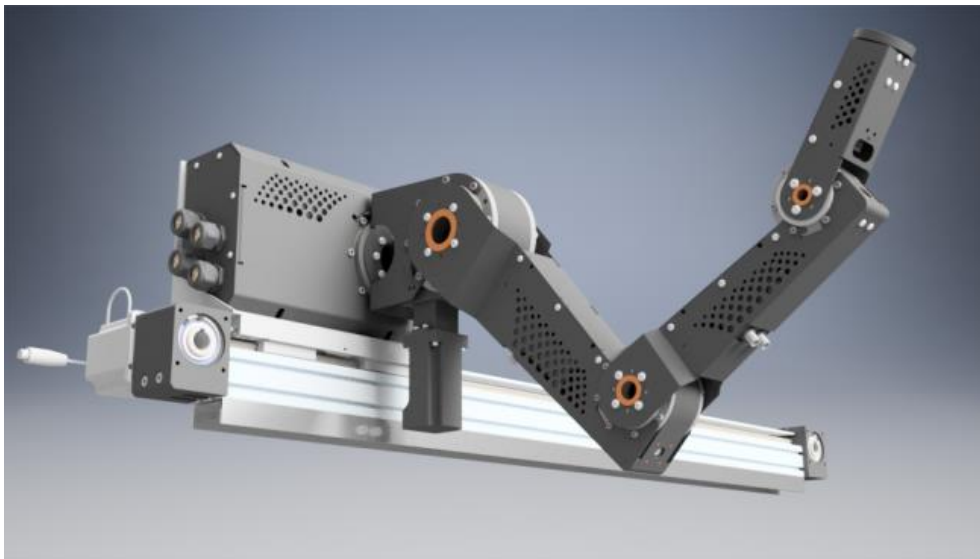
### 1.3 Opdrachtomschrijving

In het afgelopen jaar is er een prototype ontwikkeld die een visuele inspectie op een voorvleugel mogelijk maakt. De aansturing heeft slechts beperkte functionaliteit, hierdoor kan de arm wel in beweging worden gebracht, maar aspecten zoals motion control (point-to-point, kinematic translations, collision avoidance) zijn nog niet geïmplementeerd. De huidige proefopstelling moet worden klaargemaakt voor een industriële omgeving.

Om dit project te laten slagen zullen de motion control aspecten geïmplementeerd moeten worden. Hiervoor zijn enkele deelfuncties nodig. Dit zijn safety aspecten, kinematische modellen, een kinetica model en de aandrijving van de lineaire module. Als deze deelfuncties gerealiseerd zijn kan de leading edge geïnspecteerd worden. Dit zal in samenwerking gaan met het team dat zich focust op het detecteren en classificeren van schades en defecten. Het uiteindelijke doel vanuit het NLR is om geautomatiseerd te kunnen demonstreren dat de leading edge geïnspecteerd kan worden.

#### 1.3.1 Situatie omschrijving

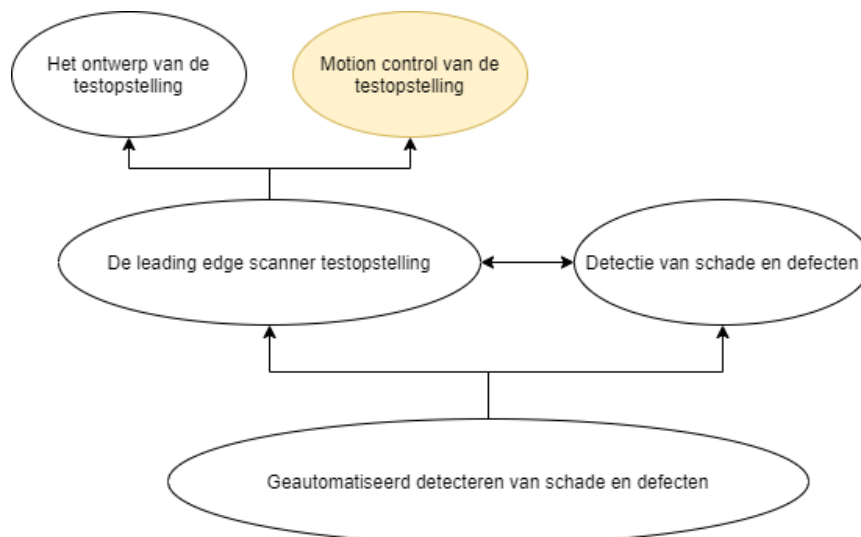
Om 3D opnames van een voorvleugel te kunnen maken zal er eerst bewezen moeten worden dat de camera zich kan positioneren door een robot. Hiervoor zal er een testopstelling ontworpen worden. Deze testopstelling bevat een robotarm die zich langs een voorvleugel van een vliegtuig kan bewegen. Het ontwerp en de mechanische opbouw van de testopstelling zal tijdens de stage door een medestagiair worden uitgevoerd. In afbeelding 1 is te zien hoe de huidige testopstelling eruit ziet. Voor de nieuwe testopstelling zal er een stellage komen waarbij er recht tegenover de testopstelling de voorvleugel geplaatst wordt. Hierdoor kan de robotarm over de gehele lengte van de voorvleugel zichzelf verplaatsen. De bedoeling is dat de leading edge verticaal geplaatst wordt. Dit is een eis van het NLR waardoor de opstelling ruimte zal besparen. Door de testopstelling verticaal te plaatsen zal er een nieuwe aansturing van de lineaire module nodig zijn. Voordat de stageperiode begon was er een robotarm waarvan de assen geïjg (handmatig geroteerd) konden worden. Daarnaast werkt het huidige kinematische model niet naar behoren, deze is namelijk voor een 6-assige robot waarbij wij gebruik maken van een 4-assige robotarm.



Afbeelding 1: De huidige testopstelling

### 1.3.2 De projectscope

Om geautomatiseerd een leading edge te kunnen inspecteren, zal er in een teamverband gewerkt moeten worden. Hierbij is er een team van stagiairs die zich volledig op het detecteren en classificeren van defecten richt (imaging deel). Daarnaast zal er een testopstelling ontworpen worden door een ander team van stagiaires. De scope van het project waar dit rapport over gaat is de motion control van deze testopstelling. Het is noodzakelijk dat de robot de posities kan behalen. Daarnaast zijn de safety factoren en het bereik van de robotarm onderdelen die binnen deze scope van dit project vallen. In figuur 2 is een schematische weergave van het gehele project te zien. De scope van dit project is gemarkeerd weergegeven.



Afbeelding 2 Schematische weergave scope project

### 1.3.3 Opleverproducten

Om het eindresultaat te halen zullen de volgende producten opgeleverd worden aan het NLR.

- De testopstelling moet naar x,y z posities en onder een hoek theta kunnen verplaatsen.
- Er moet een keuze worden gemaakt voor de aansturing van de lineaire geleider, zodat deze verticaal geplaatst kan worden.
- Er moet een kinematisch model worden opgeleverd, die de configuratie van de leading edge scanner wiskundig weergeeft.
- De optimale kinematische structuur die ervoor zorgt dat het bereik van de leading edge scanner optimaal wordt benut.
- Een werkende demo waarbij de robot zich kan positioneren om de leading edge in kaart te brengen.
- Een aanbeveling voor de safety aspecten van het project.

### 1.3.4 De activiteiten tijdens het project

- Het analyseren van het huidige systeem, inclusief de probleemstelling definiëren voor het eindresultaat.
- Het in kaart brengen van de eisen, doelstellingen en randvoorwaarden.
- Het in kaart brengen van de kinematische structuur.
- Een kinetisch model opstellen.
- Het documenteren van de gemaakte ontwerpkeuzes.
- Het gebruik maken van simulatiesoftware.
- Het gebruik maken van programmeersoftware.
- Samenwerking met collega's voor het gewenste eindresultaat.

## 2. Analysefase

In dit hoofdstuk worden de deelvragen beantwoord. Dit is het onderzoek van het project.

### 2.1 Grenzen en randvoorwaarden

(i) Wat zijn de eisen, doelen en randvoorwaarden van het project?

Om de grenzen en randvoorwaarden van dit project vast te stellen zijn er enkele specificaties geformuleerd. Deze specificaties worden onderverdeeld volgens de MoSCoW methode. Grenzen en randvoorwaarden worden bepaald om een duidelijk beeld te geven tussen de opdrachtgever en opdrachtnemer.

#### 2.1.1 MoSCoW

De MoSCoW methode bestaat uit verschillende onderdelen. De *Must have* categorie, in deze categorie staat alles wat het project moet hebben om te slagen. De *Should have* categorie zijn de eisen die zeer gewenst zijn. De *Could have* is de categorie waarbij het leuk is als het geïmplementeerd is maar niet noodzakelijk. Tot slot is er nog de categorie *Won't have* categorie, dit beschrijft de activiteiten die geen onderdeel zijn van de stage.

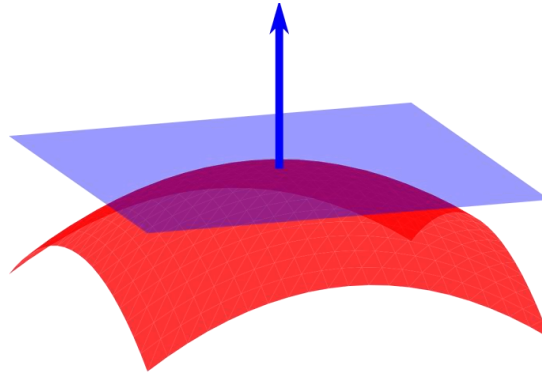
Categorie		Specificatie
Must (M)	(i)	Het aansturen van de robotarm door middel van x, y en z coördinaten, de nauwkeurigheid valt binnen de millimeter.
	(ii)	Het aansturen en positionering van de lineaire geleider, de nauwkeurigheid valt binnen de millimeter.
	(iii)	Het integreren van safety aspecten voor het beveiligen van de testopstelling binnen de Beckhoff software.
	(iv)	De eindeffector van de robotarm moet zich op <i>50 tot 70 centimeter</i> van de leading edge kunnen plaatsen.
	(v)	De robot mag bij de testopstelling geen botsing veroorzaken. De robot zal een error geven als de operator de opdracht geeft waar een botsing kan ontstaan.
Should (S)	(vi)	Een goed kinetica model waarin de robot vloeiend beweegt, zodat het image processing 'goed' kan verlopen.
Could (C)	(vii)	Een aanbeveling voor het eventueel gebruik van collision avoidance en power limitation voor de testopstelling.
Won't (W)	(viii)	Er moet gekeken naar een algemene vorm van collision avoidance voor eventuele toekomstige prototypes.

### 2.2 Detecteren van schade en defecten

(i) Wat zijn de eisen, doelen en randvoorwaarden van het project?

Naar aanleiding van voorgaande onderzoeken van stagiairs met betrekking tot detectie van defecten in een materiaal is geconcludeerd dat er een keuze gemaakt is voor drie soorten camera's [12] [13]. Een thermografische camera die ervoor zorgt dat er in het materiaal gekeken kan worden. Een RGB camera die de kleuren van het oppervlakte detecteert en een stereoscoop camera die van point clouds 3D beelden maakt van het object. Verder is uit dit onderzoek gekomen dat de objecten het best gedetecteerd

kunnen worden op een afstand van *50 tot 70 centimeter* tot het normaal van een oppervlakte. De normaal van een oppervlakte wordt geïllustreerd in afbeelding 3. De normaal is een vector loodrecht op het oppervlakte. Bij een rond oppervlak (zoals de leading edge) zijn er veel normaalvectoren in de ronding. Het positioneren van de uiteindelijke robot belangrijk, zodat er op elke plek van de leading edge op de normaal gekeken kan worden. De *50 tot 70 centimeter* van de normaal is voor een optimale field of view voor de camera's.

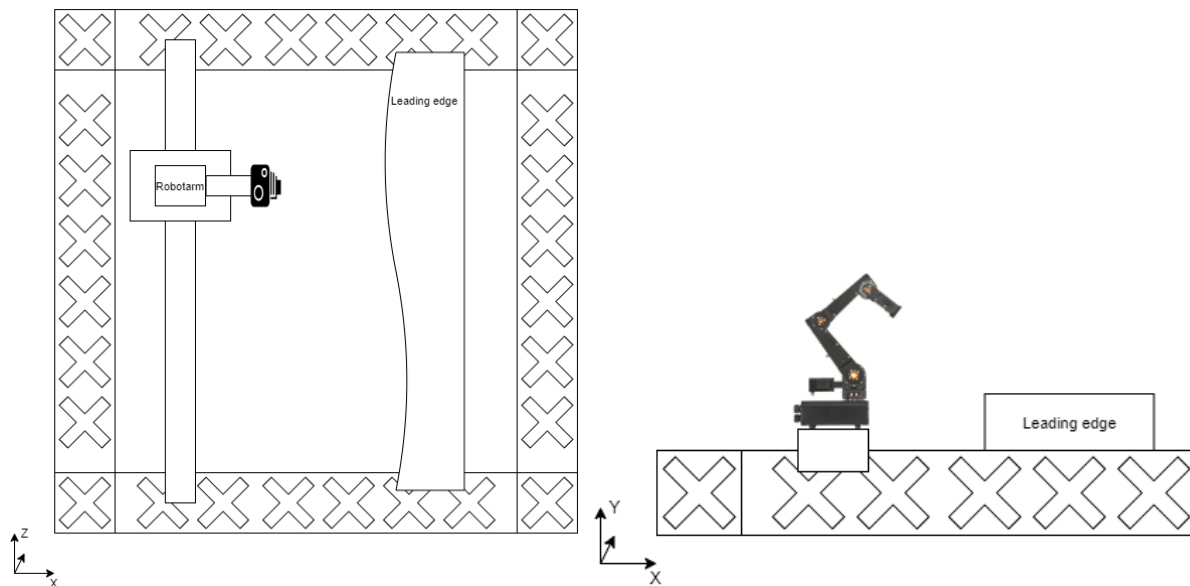


Afbeelding 3 Normaalvector van een rond oppervlak [14]

## 2.3 De testopstelling

(ii) *Hoe ziet de huidige testopstelling eruit en welke onderdelen zullen gebruikt moeten worden?*

De robot is een combinatie van een robotarm en een lineaire module. De robotarm is een Robolink RL-DC van het merk IGUS. De robotarm wordt aangestuurd met een Beckhoff PLC. Deze PLC is dan weer verbonden met een aantal drivers die de assen van de robotarm aansturen. De robotarm zit gemonteerd op een lineaire module. De lineaire module is een los onderdeel. Tijdens de stage zal er door een mede stagiair een testopstelling ontworpen worden, waarin de lineaire geleider verticaal geplaatst wordt. In dit ontwerp is ervoor gekozen dat de robotarm tegenover de leading edge is geplaatst. De gehele testopstelling is ontworpen zodat deze gemonteerd en gedemonteerd kan worden. Deze bestaat uit truss-elementen die vaak in podiumbouw worden gebruikt. Deze truss-elementen zorgen ervoor dat de testopstelling stabiel en modulair is. In afbeelding 4 is vereenvoudigde afbeelding te zien van de testopstelling.




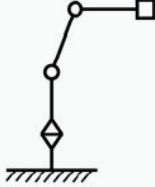

Afbeelding 4: Vooraanzicht en bovenaanzicht testopstelling, respectievelijk

### 2.3.1 Vrijheidsgraden

De robotarm heeft vier vrijheidsgraden en wordt verplaatst over een lineaire geleider. Deze lineaire geleider kan zich uitsluitend translerend gedragen en heeft daarom maar één vrijheidsgraad. Het totale robotsysteem heeft dus vijf vrijheidsgraden.

### 2.3.2 Type robot

De robotarm is een gearticuleerde robot. Dit houdt in dat de robot uitsluitend gebruik maakt van drie rotatie gewrichten. Dit type robot is ideaal om langs objecten heen te bewegen of in de testopstelling toepassing, om uit verschillende hoeken opnames van de voorvleugel te maken. Dit robot type lijkt ook het meest op een menselijke arm en wordt veel gebruikt in de robotica. In figuur 5 zijn de kenmerken van een gearticuleerde robot weergegeven. De lineaire module zorgt ervoor dat het werkgebied in de lengte verplaatst kan worden. Deze lineaire module heeft dus uitsluitend een translerend effect op het werkgebied van de robotarm.

type	kinematische structuur	werkgebied
		

Afbeelding 5: Kenmerken van een gearticuleerde robot [2]

## 2.4 Veiligheid testopstelling

- (iii) Welke manieren van “collision avoidance” en andere safety aspecten zijn er, en hoe kunnen deze worden geïmplementeerd in het huidige systeem?

Omdat de testopstelling in de ontwikkelingsfase zit, zal het lastig zijn om aan een testopstelling te bouwen die geheel voldoet aan alle veiligheidseisen. Echter, moet er wel veilig kunnen worden gewerkt met de testopstelling, daarom zijn er een aantal veiligheidsaspecten benodigd.

### 2.4.1 Adviesrapport veiligheid testopstelling

Om er voor te zorgen dat de testopstelling veilig wordt ontworpen, is er een adviesrapport geschreven. Dit document beschrijft wat er nodig is om een veilige testopstelling te ontwerpen, ook wordt er rekening gehouden met het op de markt brengen van een eventueel toekomstige inspectie robot. Hierbij zullen veel meer veiligheidseisen nodig zijn. Ook komen er zaken aan bod komen zoals noodstop veiligheid, “collision avoidance”, “Force and power limiting” en elektrische veiligheidseisen. Het adviesrapport is te vinden in bijlage A.

#### Gevareninventarisatie testopstelling

Het grootste gevaar voor de testopstelling is beknelling of aanraking van de robot met de mens. Er zullen diverse maatregelen worden genomen om dit te voorkomen. Een maatregel hiervan is dat bij een noodstop de assen geremd worden zodat de lineaire geleider niet naar beneden valt en er vervolgens een beklemming ontstaat. Een ander gevaar is de mechanische veiligheid. Daarbij wordt er gedacht aan een botsing tussen de leading edge en de robot. Als laatste moet de testopstelling voldoen aan de elektrische veiligheidseisen. Daarbij kan er gedacht worden aan fail-to-safe schakelingen van het noodstopcircuit. Dit houdt in dat de schakeling dubbel wordt uitgevoerd en deze wordt gecontroleerd op kortsluitingen. Al deze gevaren worden meegenomen in het uiteindelijke ontwerp voor de testopstelling.

#### Specificaties veiligheid

In het adviesrapport in bijlage A wordt dieper ingegaan op de veiligheid van de leading edge scanner. Uit dit advies rapport komen de volgende specificaties:

Categorie	Specificatie
Must (M)	(i) Bij een noodstop voldoet de testopstelling aan de eisen betreft machineveiligheid.
	(ii) De gehele reikwijdte van de robot is afgebakend.
	(iii) Als de robot aangestuurd wordt naar de no-go zone. Zal er een storing optreden en wordt de robot naar een veilige positie verplaatst.
Should (S)	(iv) De robot voldoet aan collision avoidance.
Could (C)	(v) De aansturing en de aansluiting voldoen aan de machinerichtlijn.
	(vi) De robot voldoet aan force and power limiting.
Won't (W)	(vii) De testopstelling voldoet (nog) niet aan de alle verplichtingen van de NEN. (zie hoofdstuk 2.1 punt II t/m punt V).
	(viii) Er wordt gedetecteerd wanneer een persoon binnen de reikwijdte van de robot komt. Er volgt een nood-stop.

### 2.4.2 Collision avoidance

Collision avoidance is een bekend begrip in de robot sector. Collision avoidance is het ontlopen en ontwijken van botsingen. Deze systemen worden gebruikt bij robots die samenwerken met mensen. Een voorbeeld is een zelfrijdende auto, waarbij een persoon plotseling oversteekt. De zelfrijdende auto detecteert dit met geavanceerde camera's en zal de persoon ontwijken. Bij robots wordt collision avoidance ingezet voor het detecteren van te naderen objecten. Objecten kunnen gedetecteerd op

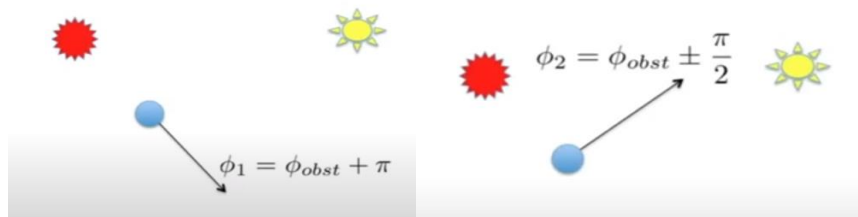
#### Motion control of the leading edge scanner testrig



verschillende manieren. Objecten kunnen gedetecteerd worden door sensoren. Deze objecten kunnen in een systeem bekend zijn, zoals bij de leading edge testopstelling.

### Strategie

Om obstakels te ontwijken is het nuttig om een algemeen algoritme te gebruiken. Obstakels ontwijken kan op veel verschillende manieren [10]. Eén manier is om een obstakel puur te ontwijken zoals afbeelding 6 illustreert. Dit wordt een “pure avoidance” genoemd. Hierbij is het blauwe symbool de robot, het zonnetje is het uiteindelijke doel en is de rode stip is het obstakel. Bij dit voorbeeld zal het obstakel nooit geraakt worden, alleen zal het lastig worden om het doel te bereiken. Een ander voorbeeld is om het obstakel onder een rechte hoek te ontwijken, zie afbeelding 6. Deze twee voorbeelden houden zich alleen bezig met het ontwijken van het obstakel. Hierbij wordt het doel niet meegenomen.



Afbeelding 6 De “pure avoidance” benadering

Ook is het mogelijk om te definiëren of er een obstakel op de route naar het doel ligt. Deze “goal to goal” benadering is in afbeelding 7 te zien is. Als laatste is er een combinatie van een “goal to goal” benadering en een “pure avoidance” benadering. De gemixte benadering wordt veel gebruikt bij mens en robot applicaties.



Afbeelding 7 De “goal to goal” benadering en de gemixte benadering, respectievelijk

### Collision avoidance Leading edge scanner

Bij de leading edge testopstelling zijn alle obstakels bekend. Om een compleet algoritme te gebruiken kan er bij de leading edge testopstelling op voorhand van een beweging twee situaties gedefinieerd worden. De eerste situatie is een beweging naar het doel zonder dat er een object tussen het traject ligt. Bij deze situatie kan de “goal to goal” methode gebruikt worden om obstakels te ontlopen. Een tweede situatie is een beweging naar het doel waarbij er een obstakel op het traject ligt. Hierbij zal de gemixte methode gebruikt worden. Deze tweede situatie zal in de praktijk bij twee mogelijkheden voorkomen. Als de robot van boven de leading edge naar de onderkant wordt gestuurd of andersom. Op dit moment zal de robot naar een vast punt gaan om de leading edge te ontwijken.

### 2.4.3 Force and power limitation

Force and power limitation wordt gebruikt om zowel mechanische veiligheid als persoonlijke veiligheid te waarborgen. Het doel van deze limitatie is als een persoon of een onderdeel bekend raakt door de robot,

### Motion control of the leading edge scanner testrig

de robot vervolgens stopt met bewegen en geen kracht meer levert. Deze methode wordt vaak gebruikt bij mens-robot applicaties.

### Theorie

Force and power limitation kan biomechanisch onderverdeeld worden in twee categorieën [10]. Translerend contact en Quasi-statisch contact. Het verschil is te zien in afbeelding 8. Translerend contact is het kort aanraken van een object. Het Quasi-statisch contact is een situatie waarbij een bekend object tussen de robot zit. Bij Quasi-statisch contact zal de kracht veel hoger zijn dan bij normaal bedrijf. Deze kracht zal oplopen en bij een piekkracht zal de motor geen kracht meer moeten leveren. Bij een translerend contact is de botsing van korte duur. Deze botsing is daarom ook een stuk moeilijker te detecteren. Translerend contact is iets wat vaker gebruikt wordt bij robotmens applicaties en wordt bij de leading edge scanner (nog) niet gebruikt.



Afbeelding 8 Het translerende contact en het Quasi-statisch contact respectievelijk

Quasi-statisch contact is bij de leading edge scanner van toepassing. Dit waarborgt de mechanische veiligheid en de persoonlijke veiligheid in het systeem.

## 2.5 Kinematica

- (iv) *Welke kinematica modellen moeten worden opgesteld voor de aansturing van de testopstelling?*

Kinematica is de studie die bewegingen van mechanismen beschrijft zonder de bijbehorende krachten. Deze studie wordt meestal los van de studie kinetika (dynamica) gezien, de leer der bewegingskrachten.

Bij een robot kan de kinematica worden onderverdeeld in twee categorieën. De forward kinematics, die de positie van de end-effector berekent en de inverse kinematics, die aan de hand van de positie van de end-effector de hoeken van de gewichten berekend.

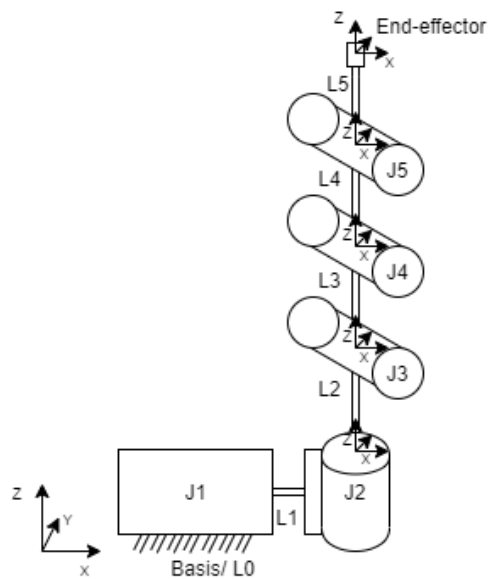
### 2.5.1 Kinematisch structuur

Om de kinematica van een robot te ontwikkelen is een duidelijk overzicht noodzakelijk. Daarvoor worden alle onderdelen van de robot onderverdeeld. Om het uiteindelijke kinematische model vereenvoudigd te houden is er voor gekozen om de robotarm te scheiden van de lineaire geleider. Omdat de lineaire geleider enkel een translerend effect heeft is deze niet meegenomen in het kinematische model. Later wordt dit wel meegenomen in de uiteindelijke positie bepaling.

De robot bestaat uit een basis (L0), joints (J1-J5), links (L1-L5) en een end-effector, zie figuur 9. Elke robot heeft N joints en N Links. Bij elke joint hoort een specifieke oriëntatie in het x, y en z stelsel. Joints worden aangeduid als translerende objecten of roterende objecten. De leading edge scanner heeft één translerende joint en vier roterende joints. In afbeelding 9 is de Kinematische structuur van de robot te zien. De eerste joint is translerend en geeft de lineaire geleider weer. De andere joints zijn geven de robotarm weer.

#### Motion control of the leading edge scanner testrig

Robotics & Mechatronics - Faculteit Techniek – versie 3  
© 2020 Copyright Hogeschool Amsterdam



Afbeelding 9 kinematische structuur waarbij Joint [Jn] Link [Ln] weergeven

### 2.5.2 Danavid-Hartenberg notatie

De Danavid-Hartenberg notatie (DH notatie) is een veel gebruikt in de kinematische analyse voor robot manipulatie [4]. De DH notatie bestaat uit vier parameters. Deze vier parameters beschrijven de gehele kinematische structuur van de robot. De DH notatie wordt weergegeven in een getransponeerde matrix [A].

$$A = Rz(\Phi_i)Tz(di)Tx(A_i)Rx(\alpha_i)$$

Hierbij is  $[\Phi_i]$  de joint hoek,  $[d_i]$  de link offset,  $[A_i]$  de link lengte en  $[\alpha_i]$  de link twist. Met deze transponatie matrix kan de voorwaartse kinematica gedefinieerd worden. Ook kan er door middel van deze matrix simulaties bijvoorbeeld het bereik van de robot weergegeven worden. In onderstaande tabel is de transponatie matrix weergegeven met de DH notatie.

Joint [i]	$\Phi [i]$	A [i]	d [i]	$\alpha [i]$
1	$\Phi_1$	<b>0</b>	$A_1$	$-\frac{1}{2} * \pi$
2	$\Phi_2$	$A_2$	<b>0</b>	<b>0</b>
3	$\Phi_3$	$A_3$	<b>0</b>	<b>0</b>
4	$\Phi_4$	$A_4$	<b>0</b>	<b>0</b>

Om de gehele DH notatie compleet te maken zijn de beweging limieten noodzakelijk. De beweging limieten worden per joint weergegeven. Elke joint heeft een fysieke beperking in zijn bewegingsvrijheid. De beweging limieten in radialen voor de roterende joint zijn:

Joint ( $\Phi_1$ ) hoek 1 gaat van  $(-\pi)$  tot  $(\pi)$

Joint ( $\Phi_2$ ) hoek 2 gaat van  $(\frac{1}{2} \pi)$  tot  $(\frac{7}{18} \pi)$

Joint ( $\Phi_3$ ) hoek 3 gaat van  $(\frac{3}{4} * \pi)$  tot  $(0)$

Joint ( $\Phi_4$ ) hoek 4 gaat van  $(\frac{3}{4} * \pi)$  tot  $(0)$

Om het kinetische model eenvoudig te houden is er gekozen om het translaterende onderdeel gescheiden te houden. Dit betekent dat bij de uiteindelijke positie bepaling de translatie wordt meegenomen. De lineaire module is recht evenredig op de uiteindelijke Z-positie van de robot. Dit is een verhouding waarbij een rotatie gelijk staat aan een bepaalde afstand in de Z-richting.

### 2.5.3 Forward kinematica

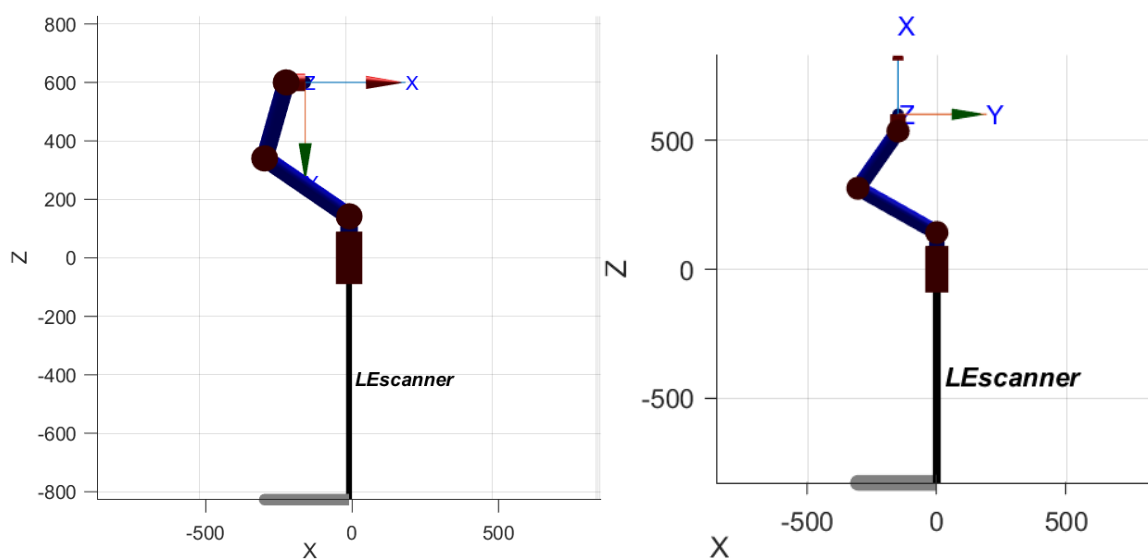
Forward kinematica is zoals eerder beschreven een berekening voor de positie van de end-effector. De input hiervan zijn de hoeken van elke joint en de lengtes van elke link. De uitkomst is een positie in het x-, y- en z- stelsel. Hiermee kunnen simulaties worden gedaan. Het helpt met de inverse kinematics en het kan worden gebruikt bij het ontwijken van objecten (collision avoidance).

In MATLAB kan er vanuit de DH notatie functies worden gemaakt. Deze functies hebben alle joint hoeken en link lengtes als input. Als output wordt de x- y- en z- coördinaten gegenereerd. Deze functies zijn:

$$\begin{aligned} x &= \cos \Phi_1 * (l_2 * \cos \Phi_2) + (l_3 * \cos(\Phi_3 + \Phi_2)) + l_4 * \cos \Phi_1 * \cos(\Phi_2 + \Phi_3 + \Phi_4) \\ y &= \sin \Phi_1 * (l_2 * \cos \Phi_2) + l_3 * \cos(\Phi_2 + \Phi_3) + l_4 * \sin \Phi_1 * \cos(\Phi_2 + \Phi_3 + \Phi_4) \\ z &= l_1 + (l_2 * \sin \Phi_2) + l_3 * \sin(\Phi_2 + \Phi_3) + l_4 * \sin(\Phi_2 + \Phi_3 + \Phi_4) \end{aligned}$$

### 2.5.4 Reverse kinematica

Reverse kinematics is het tegenovergestelde van de Forward kinematics. Dit betekent dat de input de positie van de end-effector is en de output de motor hoeken. Met deze uitgangspunt zijn er natuurlijk meerdere mogelijkheden. Daarom is er gekozen voor een specifieke configuratie. In figuur 10 is deze configuratie uitgebeeld. Bij deze configuratie kijkt de end-effector altijd naar de x-as. De oriëntatie kan aangepast worden ten opzichte van de x-as. Met deze configuratie kan elke positie behaald worden.



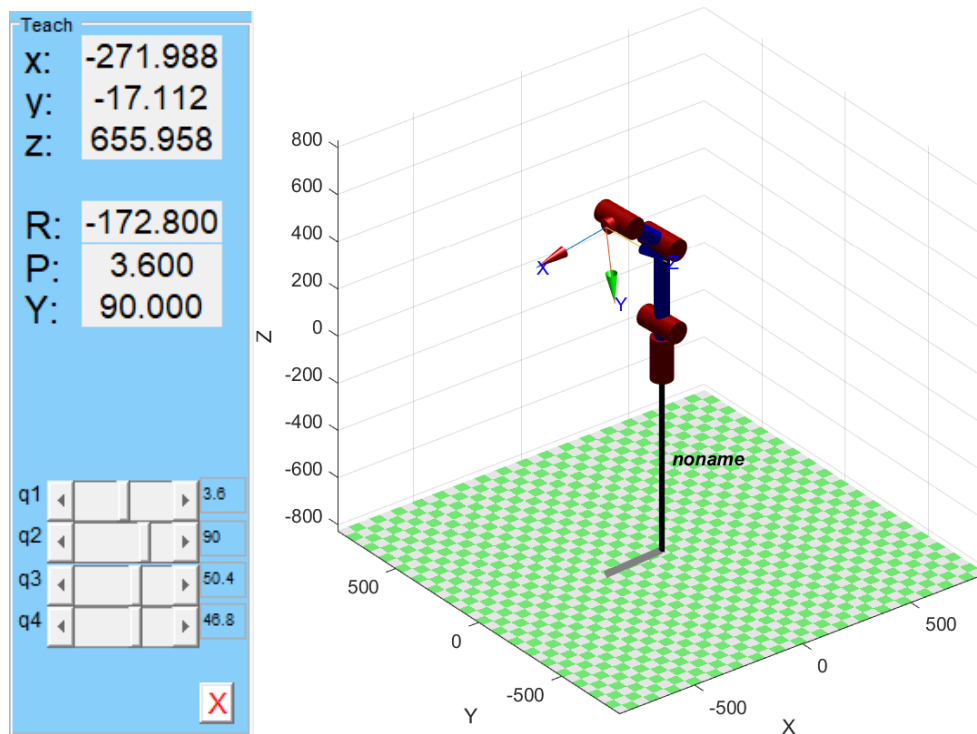
Afbeelding 10 Forward kinematics oriëntatie bij  $\theta = 0$  en  $\theta = 90$ , respectievelijk

In bijlage B zijn de berekeningen die tot de vergelijkingen komen te zien. Er zijn diverse voorbeelden beschikbaar waarbij er in vergelijkbare situaties de reverse kinematics bepaald worden. Met deze voorbeelden is er ook voor deze toepassing de reverse kinematics gemaakt. Er is door middel van een boven- en een zijaanzicht verschillende functies herleid. Dit is gebeurd met behulp van goniometrie. In de bijlage zijn de stappen tot de reverse kinematica weergegeven. In deze stappen is te zien hoe er op de vergelijkingen is gekomen. Hierbij is  $x$ ,  $y$ ,  $z$  en  $\theta$  de input daaruit komt elke motorhoek per joint. De vergelijkingen voor reverse kinematics zijn:

$$\begin{aligned}\Phi 1 &= \text{atan} 2(y, x) \\ \Phi 2 &= \text{atan} 2(z - l1 - l3 * \sin \theta), \pm \sqrt{(\sqrt{l3 * \sin \Phi 3})^2 + (l2 + l3 * \cos \Phi 3)^2 - (z - l1 - l4 * \sin \theta)^2} - \text{atan} 2(l3 * \sin \Phi 3), (l2 + l3 * \cos \Phi 3) \\ \Phi 3 &= \text{acos}(A^2 + B^2 + C^2 - l2^2 - l3^2) / (2 * l2 * l3) \\ \Phi 4 &= \theta - \Phi 2 - \Phi 3 \\ A &= x - l4 * \cos \Phi 1 * \cos \theta \\ B &= y - l4 * \sin \Phi 1 * \cos \theta \\ C &= z - l1 - l4 * \sin \theta\end{aligned}$$

### 2.5.5 Simulatie

Om de reverse kinematica eerst te simuleren in plaats van dit op de robot te testen, wordt de kans op schade of mislukkingen verkleind. Hiervoor is met een robot toolbox een simulatie mogelijk gemaakt. Deze robot toolbox is een extensie in MATLAB. Met deze toolbox kan de gehele kinematica getest worden. In deze simulatie kan door sliders de oriëntatie van elke joint bepaald worden. Hiermee kan de reverse kinematica getest worden. In afbeelding 11 is deze simulatie weergegeven. In deze simulatie staat  $q1$  voor joint (hoek) 1,  $q2$  voor joint (hoek) 2 etc. Hierin is snel de eindpositie van de robot te zien.



Afbeelding 11 Simulatie van de robot

## 2.6 Kinetica

- (v) *Wat is een goed kinetica model waardoor visuele inspecties van de leading edge kan worden uitgevoerd?*

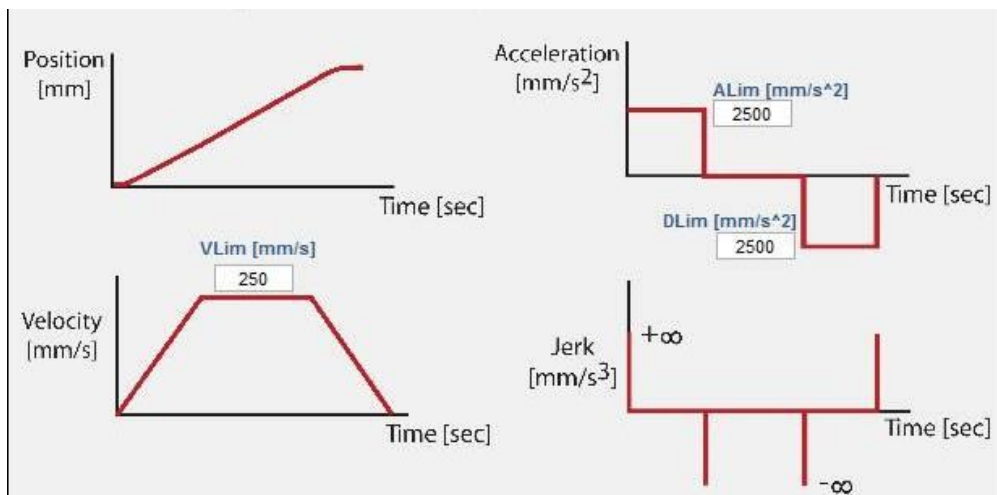
Kinetica is de leer der bewegingskrachten [5]. Dit beschrijft de samenhang tussen bewegingen en krachten. In deze paragraaf worden de bewegingen van de robotarm behandeld. Met het uiteindelijke doel, om de 'perfecte' beweging te maken.

### 2.6.1 De "perfecte" beweging

Een perfecte beweging is een relatief begrip. Bij een perfecte beweging wordt er gedacht aan een goede balans in de snelheid, acceleratie en de ruk [6]. Er zijn verschillende manieren om een mechanisch systeem te laten bewegen. Afhankelijk van het systeem en de toepassing kan de beweging bepaald worden. Elke manier van bewegen heeft voor en nadelen. Meestal heeft een beweging effect op de totale tijd van verplaatsen. Dit is dus een nadeel voor sommige toepassingen. Bij de leading edge scanner heeft de snelheid (nog) weinig prioriteit.

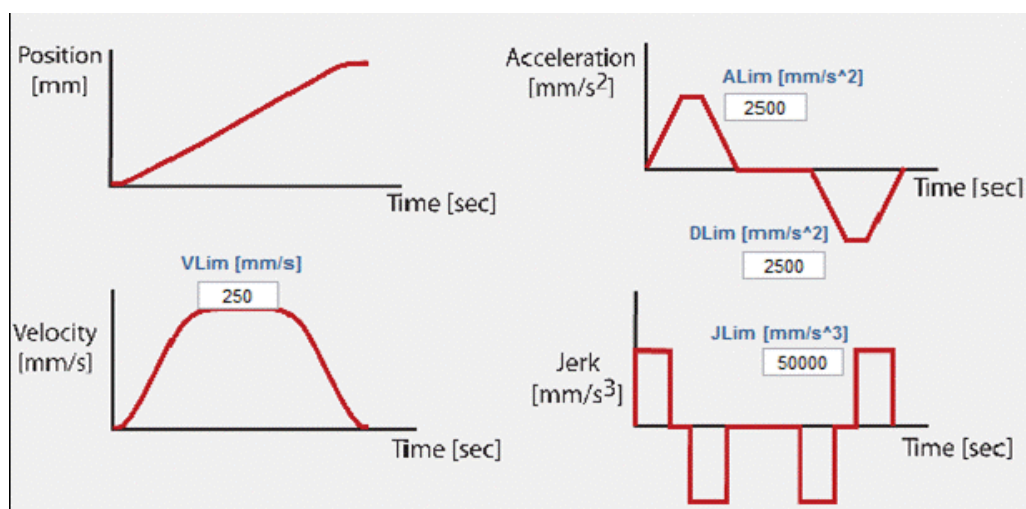
Bij het bedenken van de beweging is er één belangrijke input. Dit is dat de sensor op de end-effector zo weinig mogelijk vibratie ondergaat bij een beweging. De reden hiervan is dat als de camera meerdere foto's tegelijk neemt, de camera zo gestabiliseerd mogelijk een foto kan maken van het object. Dan zal de camera niet vibreren als er een beweging is geweest.

Zoals al eerder besproken is goede balans tussen de snelheid, acceleratie en ruk, erg belangrijk voor het goed functioneren voor de robotarm. Dit zijn elkaars afgeleiden. De ruk is belangrijk in dit proces, als de ruk te hoog wordt zal dit aan het mechanische systeem "trekken" en zorgt dit voor een schokkerig systeem [7]. Wat op voorhand een logische oplossing is om een trapeziodale beweging te ontwerpen zie figuur 12. Hierbij wordt de snelheid constant verhoogd. Dit zorgt echter zorgt voor een (theoretische) oneindige ruk.



Afbeelding 12: Een beweging met een trapeziodale snelheid [20].

Om dit te voorkomen is het noodzakelijk om de acceleratie niet lineair te maken. Omdat de ruk de afgeleide van acceleratie is zorgt dit voor een niet lineaire ruk. Een voorbeeld hiervan is te zien in afbeelding 13. Deze beweging is een S-curve en zorgt voor een mindere ruk.



Afbeelding 13: Een beweging met een S curve snelheid [20].

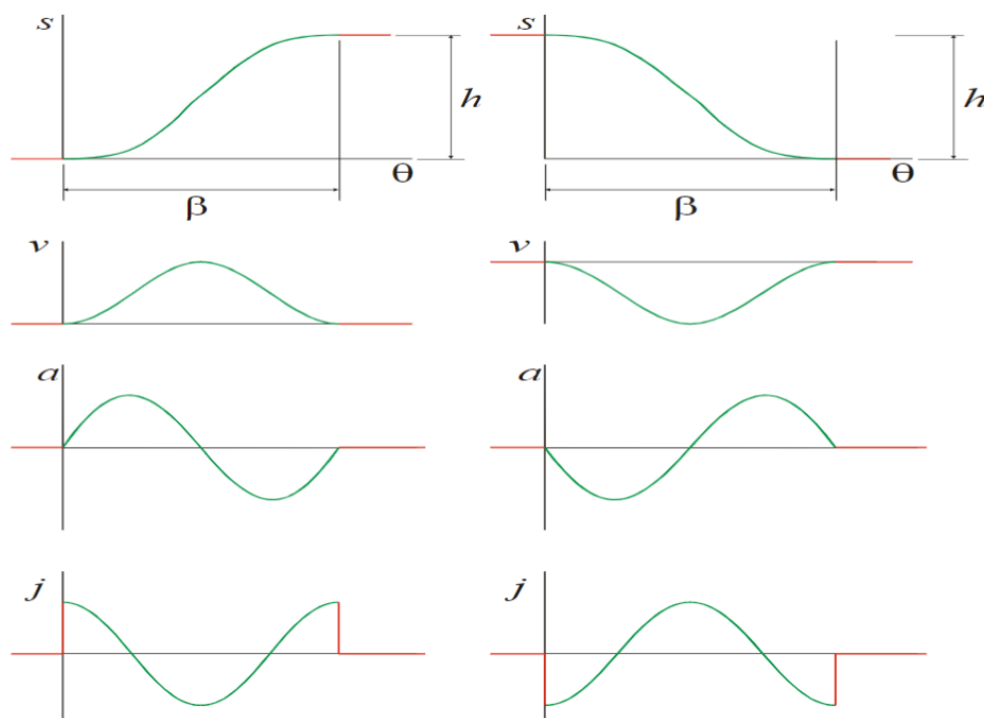
Er zijn veel verschillende soorten bewegingen, zoals eerder gezegd hebben deze bewegingen allemaal hun voor en nadelen. In figuur 14 zijn verschillende bewegingen weergegeven. Hierin is H de hoogte van de eenheid zoals te zien is in figuur 15.



Motion Scheme	Peak Velocity	Peak Acceleration	Peak Jerk
Constant Velocity	1.000 $H/T$	$\infty$	$\infty$
Constant Acceleration	2.000 $H/T$	4.000 $H/T^2$	$\infty$
Harmonic	1.571 $H/T$	4.945 $H/T^2$	$\infty$
Cycloidal	2.000 $H/T$	6.283 $H/T^2$	40 $H/T^3$
Trapezoidal	2.000 $H/T$	5.300 $H/T^2$	44 $H/T^3$
Modified Trapezoidal	2.000 $H/T$	4.888 $H/T^2$	61 $H/T^3$
3-4-5 Polynomial	1.875 $H/T$	5.777 $H/T^2$	60 $H/T^3$
4-5-6-7 Polynomial	2.188 $H/T$	7.526 $H/T^2$	52 $H/T^3$
Modified Sine	1.760 $H/T$	5.528 $H/T^2$	69 $H/T^3$

Afbeelding 14 Bewegingen vergeleken [8]

In deze tabel is er één beweging waarbij de jerk minimaal is. Omdat de tijd geen vereiste is bij het scannen van de leading edge, is dit de beste optie. De cycloïdale beweging is een combinatie van verschillende functies. Maar vereenvoudigd is het een sinus met een knik. Figuur 15 is het SVAJ diagram van een cycloïdale functie.



Afbeelding 15: SVAJ diagram cycloïdale functie [8]

Hierbij horen de volgende bewegingsvergelijkingen [8]. Deze bewegingsvergelijkingen worden geïmplementeerd binnen de Beckhoff software. Hierna zullen deze bewerkingen aangepast worden om de karakteristiek te krijgen zoals te zien is in afbeelding 15.

$$v = \frac{ds}{dt} \quad a = \frac{dv}{dt} \quad j = \frac{da}{dt}$$

$$s = \frac{h}{2} * (1 - \cos \cos \left( \pi * \frac{\theta}{\beta} \right)) - \frac{1}{4} (1 - \cos \cos \left( 2\pi * \frac{\theta}{\beta} \right))$$

$$v = -\frac{\pi h}{\beta^2} (\pi * \frac{\theta}{\beta}) + \frac{1}{2} \sin \sin \left( 2\pi * \frac{\theta}{\beta} \right)$$

$$a = -\frac{\pi^2}{\beta^2} * \frac{h}{2} (\cos \cos \left( \pi * \frac{\theta}{\beta} \right) + \cos \cos \left( 2\pi * \frac{\theta}{\beta} \right))$$

$$j = \frac{\pi^3}{\beta^3} * \frac{h}{2} (\sin \sin \left( \pi * \frac{\theta}{\beta} \right) + 2 \sin \sin \left( 2\pi * \frac{\theta}{\beta} \right))$$

## 2.7 Bereik robotarm

- (vi) *Wat is de optimale kinematische structuur waarbij de leading edge het best geïnspecteerd kan worden.*

Om het bereik van de robotarm optimaal te benutten zijn er ontwerpkeuzes gemaakt. In *paragraaf 3.1 Bereik van de robotarm* zijn deze ontwerpen uiteengezet. Dit zal gebeuren door het gebruik van een keuzematrix. Waarbij verschillende criteria ervoor zorgen dat de beste keuze wordt gemaakt.

## 2.8 Implementatie software

- (vii) *Hoe kunnen de kinematica modellen worden geïmplementeerd binnen de Beckhoff TwinCAT omgeving?*

De in kinematische modellen die in *paragraaf 2.5 kinematica* zijn herleid worden in de software geïntegreerd. Dit wordt in de detailleringfase verder toegelicht.

## 2.9 Aansturing lineaire module

- (viii) *Welke aandrijving voor de lineaire module is het meest geschikt in een verticale testopstelling?*

Voor de aansturing van de lineaire module zijn er verschillende ontwerpkeuzes gemaakt. In *paragraaf 3.2 aansturing lineaire module* zijn deze ontwerpen uiteengezet. Deze aanpak van dit ontwerp zal hetzelfde zijn als bij het bereik van de robotarm.

## 3. Ontwerpfase

In dit hoofdstuk worden twee ontwerpen uiteengezet. Het eerste ontwerp betreft het vergroten en optimaliseren van het bereik van de robotarm. Het tweede ontwerp is de aansturing van de lineaire module.

### 3.1 Bereik van de robotarm

Met behulp van de camera's zal de robotarm op een afstand van *50 tot 70 centimeter* gepositioneerd worden, dit is de ideale afstand voor het gezichtsveld van de camera's. Hierbij is het belangrijk dat de end-effector op de normaal van het vliegtuig oppervlakte geplaatst wordt. Ook is het belangrijk dat de end-effector van de robotarm onder verschillende hoeken van de normaal in de richting van oppervlakte gepositioneerd wordt, dit om de invloeden van reflectie te onderzoeken.

De robotarm heeft een beperkt bereik. Om bij de leading edge scanner goed onderzoek te kunnen doen, is het belangrijk dat de robotarm zich kan positioneren in een specifiek bereik ten opzichte van de leading edge. Hierbij zorgt de lineaire module ervoor dat de robotarm zich kan verplaatsen over de gehele lengte van de leading edge. Het is belangrijk om te bepalen waar de lineaire module geplaatst wordt ten opzichte van de leading edge. Er zijn twee ontwerpkeuzes gemaakt. De eerste is waarop de lineaire module ten opzichte van de leading edge geplaatst wordt. De tweede gaat over het modificeren van de robotarm. Uiteindelijk zal een keuzematrix bepalen wat de beste oplossing is.

#### **Uitgangspunten bereik**

- Het is belangrijk dat de ronding van de leading edge in zijn geheel in beeld komt. Dit is het voorste deel van de leading edge.
- Bij het in kaart brengen van dit deel zal de camera op *minimaal 50 centimeter en maximaal 70 centimeter* gepositioneerd worden.
- Het is belangrijk dat de robot op de normaal van deze ronding kan komen en daarbij ook verschillende hoeken kan aannemen.
- De robot moet zoveel mogelijk van de leading edge in kaart kunnen brengen.
- Het belangrijk dat een representatief stuk van de onderkant en de bovenkant getest kan worden.

#### **Tweede link modificatie robotarm**

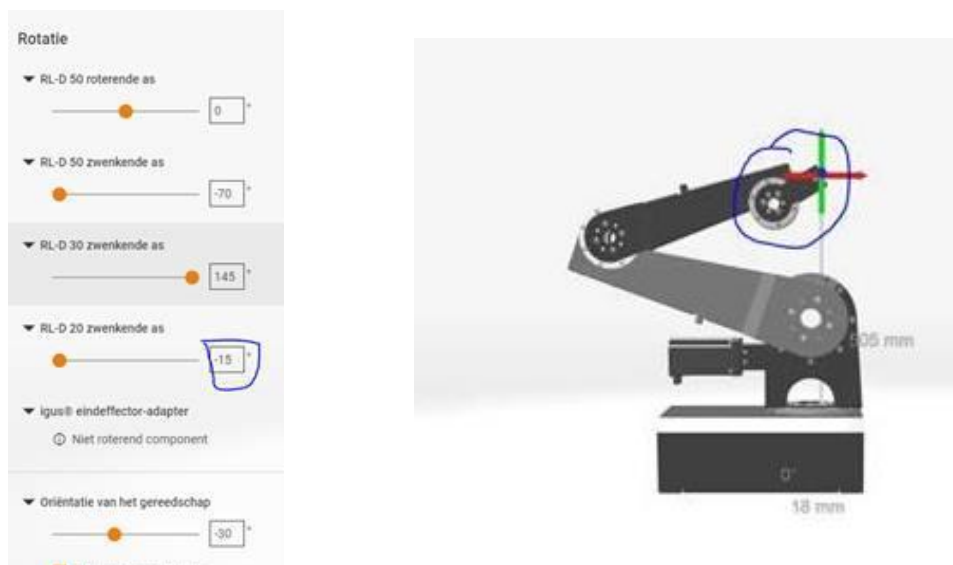
Om deze uitgangspunten zo goed mogelijk mee te nemen in het ontwerp voor de plaatsing van de onderdelen, zijn er een aantal aanpassingsmogelijkheden. Het is bijvoorbeeld mogelijk om de robotarm aan te passen om het bereik te vergroten. De robot wordt bij sommige onderdelen fysiek beperkt tot bepaalde bereiken. In figuur 16 is een fysieke beperking te zien. Op punt 12 in deze afbeelding is een motor te zien. Deze motor beperkt de bewegingshoek van joint 2. Door enige aanpassingen kan het uiteindelijke bereik geoptimaliseerd worden. Dit zorgt voor een ander (kijk) bereik richting de leading edge. In afbeelding 12 is een weergave te zien van deze eventuele modificatie. Hierbij is nummer elf omgekeerd gemonteerd. In de eerste situatie kan de derde link zich maar tot 50° naar links roteren en in de nieuwe situatie kan de derde link naar 85°. Dit staat met lijnen staat weergegeven in afbeelding 16.



Afbeelding 16 De nieuwe modificatie van de robotarm, hierbij is de oude situatie en de nieuwe situatie respectievelijk te zien. [9]

### Laatste link modificatie

Een andere mogelijkheid om de robotarm te modificeren is door de laatste link “mobiel” te maken. Hierdoor zal deze link langs de derde link kunnen bewegen. In afbeelding 17 is een voorbeeld te zien hierbij is te zien dat het bereik beperkt is tot  $-15$  graden. Wanneer deze laatste link langs de derde link kan bewegen, zal het bereik tot  $-90$  graden beperkt zijn. Het doel van de laatste link modificatie is om dit bereik te vergroten.



Afbeelding 17 het bereik van de laatste link via de robotsimulator [9]

### 3.1.1 Ontwerp mogelijkheden

In bijlage C zijn verschillende mogelijkheden meegenomen om tot een ontwerpkeuze te komen. Er is in een tekenprogramma bij drie verschillende kijkhoeken de afstand tot de leading edge bepaald. Hierbij is de laatste link modificatie overal meegenomen, aangezien bij deze modificatie het bereik alleen groter wordt. Verder zijn in deze bijlage de volgende ontwerpkeuzes respectievelijk meegenomen:

- Optie 1: De huidige situatie zonder tweede link modificatie.  
 Optie 2: 250 millimeter extra afstand zonder tweede link modificatie.  
 Optie 3: 500 millimeter extra afstand zonder tweede link modificatie.  
 Optie 4: De huidige situatie met tweede link modificatie.  
 Optie 5: 250 millimeter extra afstand met tweede link modificatie.  
 Optie 6: 500 millimeter extra afstand met tweede link modificatie.

### 3.1.2 Keuzematrix voor het bereik

Het volgende criterium met bijbehorende punten zijn voor dit ontwerp opgesteld.

- Er wordt door drie criteria gekeken of de positie binnen de gewenste 50-70 centimeter komt. 10 punten als deze hierbinnen valt en 5 punten als deze hier net buiten valt.
- Er wordt gekeken of de ronding in zijn geheel in kaart wordt gebracht. 10 punten als dit in zijn geheel lukt.
- Bij het laatste criterium wordt er gekeken of er een representatief stuk van de leading edge in kaart kan worden gebracht. Met een representatief stuk wordt er een inschatting gedaan of er meer dan 50% van de leading edge in beeld kan worden gebracht.

Keuzematrix bereik						
	Optie 1	Optie 2	Optie 3	Optie 4	Optie 5	Optie 6
Positie 1 (w = 3)	0	0	5	0	5	5
Positie 2 (w = 3)	0	0	0	0	5	10
Positie 3 (w = 3)	0	0	10	0	10	0
Ronding in kaart (w = 3)	0	0	0	0	10	10
Representatief stuk Leading edge (w = 2)	0	0	10	0	10	10
<b>Totaal</b>	<b>0</b>	<b>0</b>	<b>65</b>	<b>0</b>	<b>110</b>	<b>95</b>

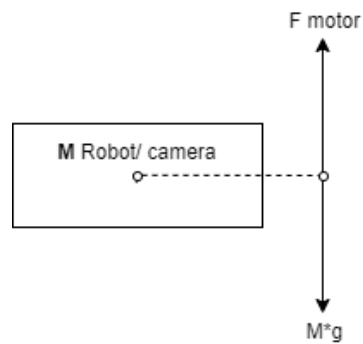
Uit de keuzematrix blijkt dat optie 5 en optie 6 bijna gelijk zijn geëindigd. Duidelijk is wel dat de tweede link gemodificeerd moet worden. Ook zal er een extra afstand tussen de leading edge en de lineaire module komen. De ideale afstand van de lineaire module tot de leading edge zal tussen de 250 millimeter en de 500 millimeter extra afstand liggen.

## 3.2 Aansturing lineaire module

Voor de lineaire module is er ooit gekozen voor een stepper motor van Beckhoff. Er is voor deze motor gekozen toen de lineaire module horizontaal was geplaatst. Hierbij waren er een aantal nadelen nu in de nieuwe situatie de lineaire module verticaal wordt geplaatst. De motor heeft bijvoorbeeld geen encoder, levert niet genoeg koppel om de payload omhoog te tillen, en de robotarm inclusief camerasysteem kan vallen als de spanning wegvalt. Daarom is er gekozen om een nieuwe aansturing te ontwerpen.

### Benodigde kracht voor de motor

De lineaire geleider is een tandriem module en wordt verticaal geplaatst. De lineaire geleider verplaatst de robotarm met de camera. Het totale gewicht van de robotarm en de camera is 28 kilogram. Met deze gegevens is het vrij lichaamsschema gemaakt, zie figuur 19. Voor het overzicht is er gekozen om de krachten te verplaatsen in het schema. In werkelijkheid zitten deze in het hart van de massa.



Afbeelding 97 VLS robotarm

Om hiervan de benodigde kracht te berekenen wordt de kracht in dezelfde richtingen bij elkaar opgeteld.

$$F_m = M \cdot g$$

Om de benodigde kracht te berekenen wordt de massa vermenigvuldigd met de gravitatieconstante.

$$28 \cdot 9.81 \approx 275 \text{ N}$$

Hierna wordt het maximale koppel van de aansturing bepaald. Hierbij is het koppel (T) afhankelijk van de kracht (F) en de straal (r) van de lineaire module.

$$T = F \cdot r$$

De lineaire module is een tandriem aandrijving en heeft een straal van  $31\text{mm}$  [11]. Hierbij heeft de tandriemaandrijving een losbreekkoppel van  $1 \text{ N/m}$ . Dit is het benodigde koppel om de tandriem in beweging te brengen. Deze wordt uiteindelijk opgeteld bij de berekening.

$$T = (275 \cdot 0.031) + 1$$

Hieruit volgt dat het maximale benodigde koppel  $9.5 \text{ N/m}$  is. Deze kracht wordt later meegenomen in de keuze voor de motor.

## Specificaties aansturing

Belangrijk is dat de motor minimaal deze kracht aankan. Dit gegeven wordt meegenomen in de motor keuze van de lineaire geleider. Ook is het een vereiste dat de motor samen werkt met de Beckhoff PLC. Verder is het noodzakelijk dat de robot niet naar beneden valt, wanneer de spanning van het systeem wegvalt. Als laatste moet de motor de lineaire module kunnen positioneren binnen de millimeter. De specificaties in het MoSCoW model:

Categorie	Specificatie	
Must (M)	(i)	De motor moet aan te sturen zijn met de Beckhoff PLC en maakt gebruik van de compact drive technologie
	(ii)	De koppel die de motor en eventuele gearbox moet minimaal 3 x de benodigde kracht leveren.
	(iii)	Wanneer de spanning van het systeem weg valt moet de robotarm niet naar beneden vallen.
	(iv)	De motor moet de lineaire module aan kunnen sturen binnen de millimeter.
Should (S)	(v)	De motor en de motoraansturing werken op een spanning van 24 [V]
Could (C)	(vi)	In de aansturing zit geïntegreerde veiligheids mogelijkheden.

### 3.2.1 Ontwerp mogelijkheden

Met deze gegevens zijn er drie alternatieven mogelijkheden gevonden. Eerst worden deze mogelijkheden beschreven, waarna later de opties beoordeeld worden. De mogelijkheid met de hoogste beoordeling wordt uiteindelijk gebruikt.

#### Optie 1:

De huidige stappenmotor gebruiken om de robotarm te liften. Hierbij is het belangrijk dat de robotarm niet naar beneden valt wanneer de spanning wegvalt. Hiervoor zal er een contragewicht systeem bedacht worden. Hiervoor zal er ook minder kracht nodig zijn om het systeem omhoog te verplaatsen. Echter is het een nadeel dat de huidige stappenmotor geen encoder heeft. Hierdoor zal het positioneren lastiger worden en wellicht onnauwkeurig als de motor kan gaan slippen. Bij deze optie is het noodzakelijk om een referentiepunt op de as te monteren zodat er goed gekalibreerd kan worden.

#### Optie 2:

Een tweede mogelijkheid is om een stappenmotor in te kopen met een encoder. Hierdoor zal het positioneren van de lineaire module gemakkelijker en nauwkeuriger worden. Het nadeel is dat de sterkste stappenmotor die Beckhoff verkoopt en op 24 Volt werk, een koppel kan leveren van 8 Newton per meter. Dit is niet genoeg koppel en zal er een gearbox bij moeten komen en/of een contragewicht systeem erbij gemaakt moeten worden.

#### Optie 3:

Een derde mogelijkheid is om een servomotor aan te schaffen die de benodigde koppel in een keer kan leveren. Servo motoren hebben altijd een terugkoppeling hierdoor zal positioneren van de lineaire module eenvoudiger worden. Ook zal het positioneren nauwkeurig worden ten opzichte van een stappenmotor die een referentiepunt wordt gebruikt. Servomotoren van Beckhoff zijn in veel verschillende spanningen beschikbaar. De sterkste servomotor die op een spanning 24 Volt werkt is de AM814x. Deze servo heeft een koppel van 2,4 Newton per meter. Hierbij zal een gearbox nodig zijn. De gearbox zal minimaal een ratio nodig hebben van 1/20. Hierdoor zal het systeem langzamer worden maar wel sterker en nauwkeurig in verband met de gearbox verhouding. Het grote voordeel van deze motoren in vergelijking met optie 1 en optie 2 is dat deze servomotoren een holding brake hebben. Een holding brake is een rem op een motor wanneer de spanning wegvalt. Dit wordt gedaan met magneten die op de as worden aangetrokken als de spanning wegvalt. Deze motoren hebben een safe-torque geïntegreerd. Dit is een geïntegreerde beveiliging die het koppel van de servomotoren weghaalt. Het voordeel van deze optie is dat er geen extra contragewicht systeem ontworpen zal worden.



### 3.2.2 Keuzematrix lineaire module

Nu de opties bekend zijn, kan er een definitieve keuze gemaakt worden voor de aansturing van de lineaire module. Hiervoor is de onderstaande keuzematrix gemaakt. Hierin is een wegingsfactor meegenomen die meegeeft welke criteria belangrijker zijn. De criteria zijn kosten, haalbaarheid en de naar schatting beste technische oplossing. Er is bij het criterium kosten een inschatting gemaakt of dit deelontwerp binnen het budget past. De haalbaarheid is een inschatting of de aansturing te realiseren is. De technische oplossing is een inschatting wat de op voorhand beste technische oplossing is.

Keuzematrix			
	Optie 1	Optie 2	Optie 3
Kosten ( $w = 1$ )	2	4	9
Haalbaarheid ( $w = 8$ )	2	2	8
Technische oplossing ( $w = 10$ )	3	5	8
Totaal	48	60	152

Uit de keuzematrix komt naar voren dat optie 3 een beste oplossing zou zijn. Een groot voordeel van het gebruik van optie 3 is dat er geen extra contragewicht oplossing bedacht hoeft te worden. Deze mogelijkheid voldoet ook als enige aan alle op voorhand gespecificeerde eisen.

## 4. Detailleringfase

*In dit hoofdstuk worden de deelconcepten van het resultaat beschreven. Dit is opgedeeld in een software gedeelte en een hardware gedeelte. In dit hoofdstuk zijn ook de testen uiteengezet.*

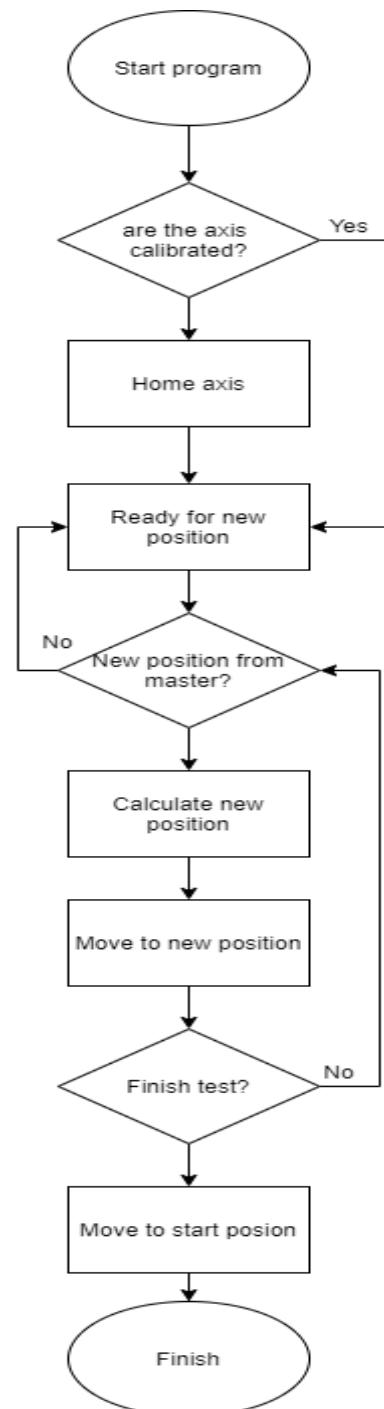
### 4.1 Detaillering software

Voor het hoofdprogramma is er een flowchart die de functionaliteit beschrijft. Daarnaast is er de implementatie van de kinematica en de kinetica. Daarbij was het noodzakelijk om de aansturing van de lineaire module in te regelen.

De integratie van de software wordt gedaan in het programma Twincat. Twincat wordt in veel toepassingen gebruikt. Twincat wordt gebruikt binnen de Microsoft Visual Studio omgeving. Het gehele programma is geschreven in ST (structured text). Structured text is afgeleid van de programmeertaal C. Het gehele programma is te zien in bijlage D.

#### 4.1.1 Flowchart

De software van het main programma wordt in de volgende flowchart gevisualiseerd.



Afbeelding 19 Flowchart main programma leading edge scanner

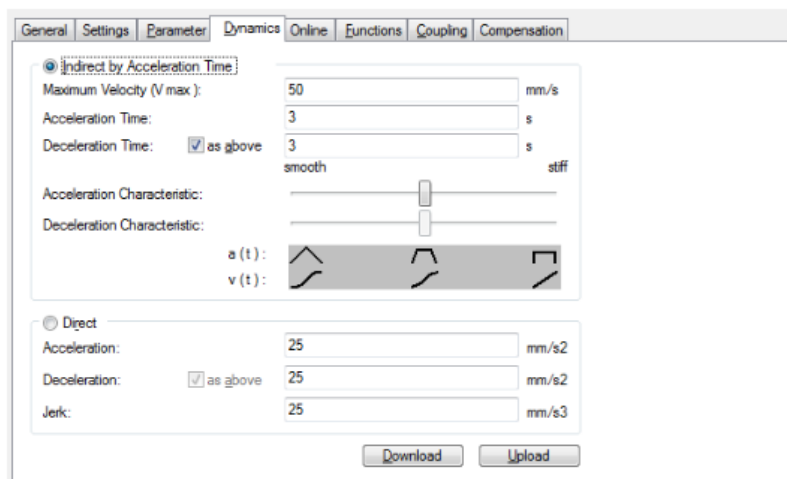
### 4.1.2 Kinematica

De reverse kinematics wordt gebruikt om de robotarm te positioneren. Voor de verplaatsing in de Z richting wordt de lineaire module gebruikt. Daarom zal de eerste link van de robotarm in eerste instantie niet gebruikt worden. Deze link zal richting de lineaire module gericht zijn. De link kan in de toekomst wel gebruikt worden om vanuit een andere hoek naar het object te kijken. Dit kan nodig zijn als er enige reflectie te zien is. Deze hoek is dan een rotatie ten opzicht van de Z-richting.

Voor de x-, y- en de oriëntatie van de robot kan de gehele voorwaartse kinematica overgenomen worden. Voor de z- positie van de robot is de verhouding nodig van de servomotor tot de uiteindelijke tandriempositie. De slag per omwenteling van de tandriemmodule is  $170\text{ mm}$  [11]. De servo gearbox verhouding is 1 op 10. Dit betekent dat tien omwentelingen van de servomotor een verplaatsing is van  $170\text{ mm}$  in de Z –richting. Hieruit volgt dat de volgende verhouding voor de verplaatsing in de Z – richting: 720: 17

### 4.1.3 Kinetica

Het doel van het integreren van de kinetica is om de ruk in het systeem te minimaliseren. Binnen Beckhoff is het mogelijk om de karakteristiek in te stellen. Hierbij is er voor gekozen om de grafiek zo veel mogelijk te laten lijken op de cycloidale. Aansluitend hierop zijn de berekende acceleraties en de acceleratietijd toegevoegd zoals te zien is in afbeelding 20.

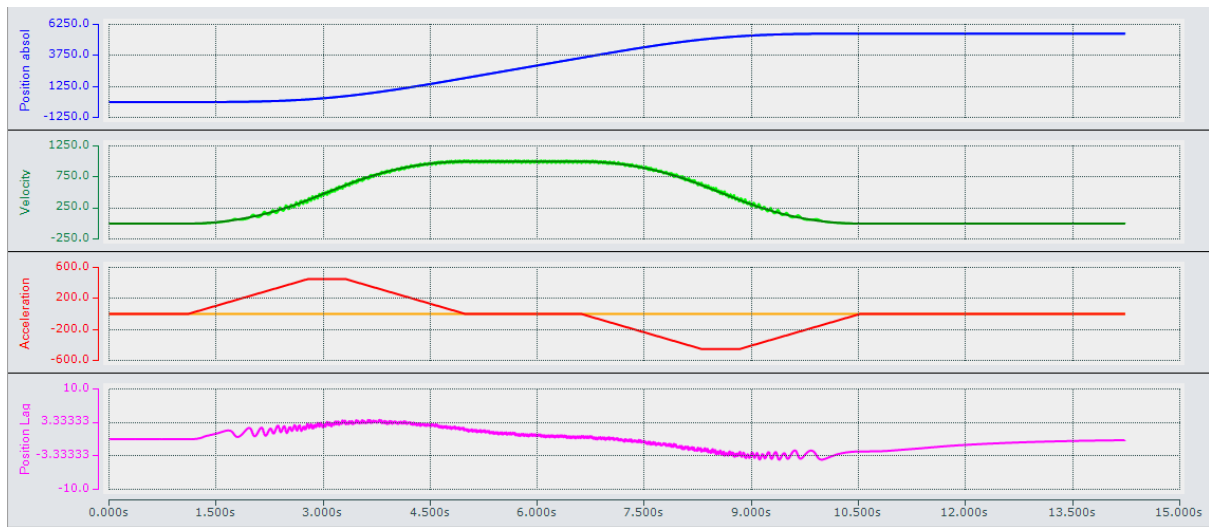


Afbeelding 20 versnellingen profiel in twincat

### 4.1.4 Inregelen van de servo motor

Om de servomotor te laten werken op het systeem wordt deze motor ingeregeld. De servomotor wordt grotendeels door Beckhoff vooraf ingesteld. Alleen is elk systeem anders. In het systemen kunnen factoren meespelen zoals: het gewicht, massastraagheid en gearbox verhoudingen. Hiervoor moeten sommige parameters ingesteld worden. Dit zijn parameters als de acceleratie, voedingsspanning, de snelheid. Verder is het belangrijk dat de servomotor “getuned” wordt afhankelijk van het systeem. Hiervoor is de terugkoppeling *position lag* belangrijk. De *position lag* bepaald de nauwkeurigheid van de servomotor. De positon lag is afhankelijk van de snelheid en de acceleratie en is uitgedrukt in een vertraging in millimeters. Beckhoff adviseert om het systeem te tunen waarbij de position lag lager is dan  $5\text{ mm}$  . Om dit te realiseren is er proefondervindelijk de acceleratie aangepast om deze position lag onder de  $5\text{ mm}$  te houden. In *bijlage E plots motion* zijn de gegevens en de plots van het tunen te vinden. Uiteindelijk is er voor een acceleratie van  $435\text{ graden/s}^2$  gekozen. Hierbij hoort de plot te vinden in

afbeelding 21. In deze plot is de respectievelijk de positie, de snelheid, de acceleratie en de position lag te zien.



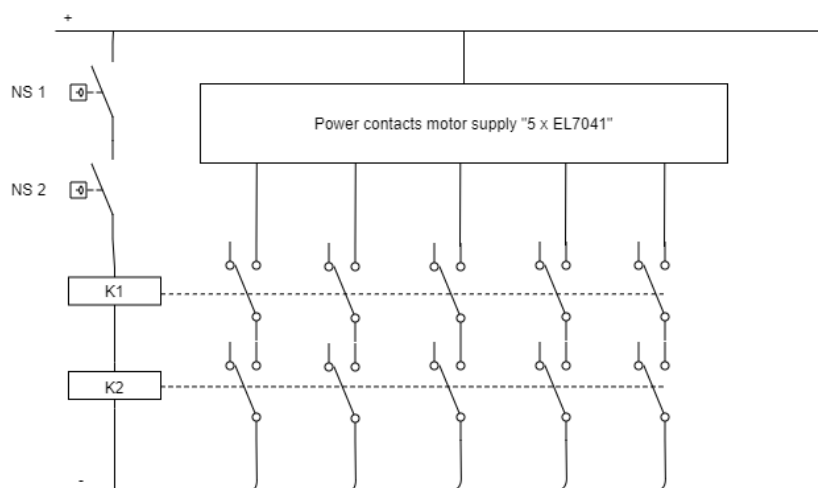
Afbeelding 21 tune plot voor de position lag

## 4.2 Detaillering hardware

De detaillering van de hardware kan onderverdeeld worden in twee categorieën. De eerste is de elektrische deralering. De tweede categorie is de mechanische detaillering.

### 4.2.1 Elektrische detaillering

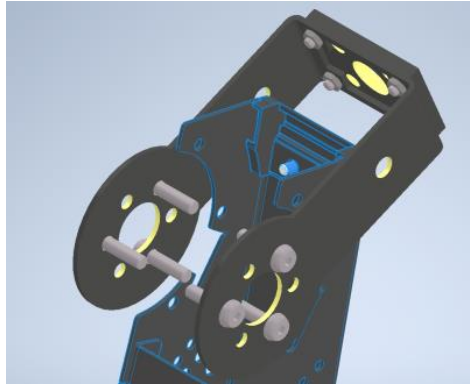
Voor de veiligheid onderdeel is er een noodstop circuit ontworpen. Dit noodstopcircuit is uit het onderdeel het adviesrapport voor de veiligheid van de testopstelling gekomen. De elektrische schakeling is ziet er als volgt uit:



Afbeelding 22 Electrische schema noodstopcircuit

#### 4.2.2 Mechanische detaillering

Voor de detaillering van de hardware zijn er in hoofdstuk 3 ontwerpkeuzes gemaakt met betrekking tot het bereik. Er is gekozen om de laatste link te modificeren en om de tweede link omgekeerd te monteren. In afbeelding 23 is het ontwerp van de laatste link te zien.



Afbeelding 23 het ontwerp van de nieuwe laatste link

#### 4.3 Test van de kinematica

Om een algemeen beeld te krijgen of de robotarm de nodige precisie haalt is er een testopstelling bedacht. Het doel van deze testopstelling is verificatie van de kinematica. Het zal een globale status geven van de nauwkeurigheid van de robotarm. In dit deel wordt de eerste specificatie (i) getest.

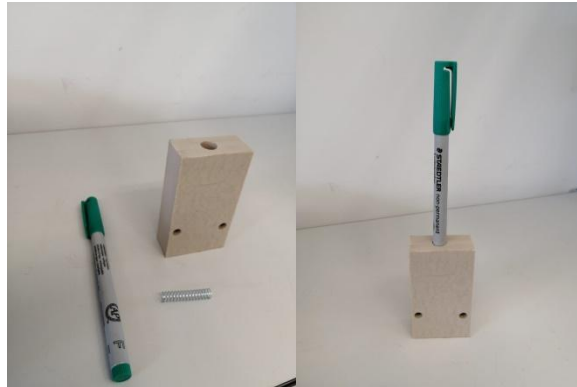
##### 4.3.1 Testopzet

Met deze test wordt er op de end-effector een stift gemonteerd. Er wordt een whiteboard aan de zijde van de robot geplaatst. De robot markeert vervolgens de behaalde posities op het whiteboard. Omdat de robot in drie dimensies kan bewegen en een whiteboard plat is. Is er gekozen voor twee testopstellingen. In figuur 24 representeert het linker deel van de afbeelding de test in de Z richting. Het rechterdeel van figuur 24 laat zien hoe de X richting en de Y richting getest wordt.



Afbeelding 24 Testopstelling waarbij Z richting wordt getest en de X en Y richting, respectievelijk

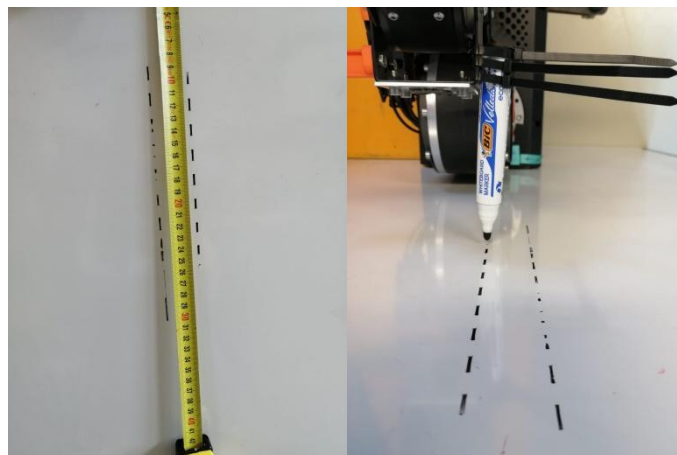
Om er voor te zorgen dat de stift spanning zal houden op het bord. Is er een systeem bedacht waarbij de stift in een geleidend gat van een blokje zit. Onder deze stift zit een veer die de stift op het bord drukt. In figuur 25 is het veersysteem te zien.



Afbeelding 25 Het stift/ veersysteem

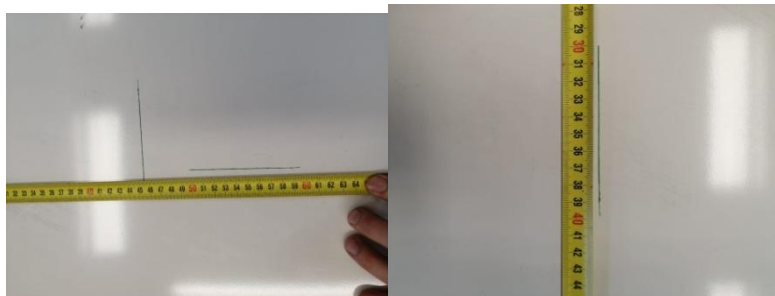
#### 4.3.2 Resultaten

In figuur 26 is het resultaat van de test in de Z-richting te zien. Bij deze test is er vanuit het programma de opdracht gegeven om streepjes van één centimeter te markeren.



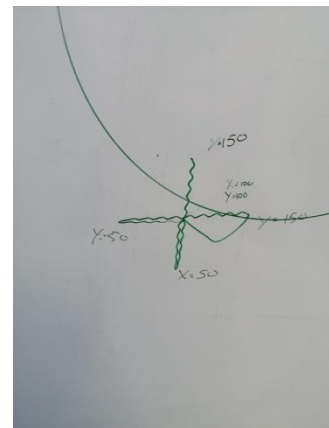
Afbeelding 26 Resultaat van de test in de Z-richting

In figuur 27 zijn de resultaten in de x en de y richting respectievelijk te zien. Hierbij is er de opdracht gegeven om een lijn van tien centimeter te tekenen. In figuur 28 zijn de resultaten te zien van een samenwerking tussen de x- en de y- richting.



Afbeelding 27 De resultaten in de X- en de Y- richting, respectievelijk

Afbeelding 28 Test in de X en de Y richting



### 4.3.3 Conclusie

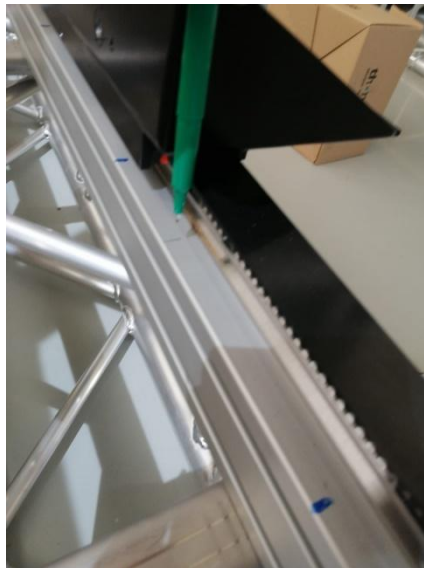
Uit deze test kan er geconcludeerd worden dat de robot in de X, Y en de Z richting aangestuurd kan worden binnen de millimeter. Echter wanneer de robot in de X en de Y richting wordt aangestuurd lijkt er een bepaald trapezium beweging in te komen. Dit is een offset die waarschijnlijk komt door de niet symmetrische vorm van de robotarm. Deze specifieke verplaatsing zal nog niet gebruikt worden in de huidige testopstelling. De overige verplaatsingen zullen wel gebruikt worden in de testopstelling en voldoen aan de specificaties.

## 4.4 Test voor de aansturing lineaire geleider

Als tweede wordt de positionering van de lineaire geleider getest. Hiervoor is er een servomotor en een gearbox uitgekozen. In dit hoofdstuk wordt getest of de op voorhand bepaalde specificaties behaald kunnen worden. Het doel was om de lineaire geleider binnen de millimeter te kunnen positioneren.

### 4.4.1 Testopzet

De lineaire geleider beweegt zich in één richting. Om de testen of de robot zich binnen de millimeter kan positioneren wordt er een pen langs het bewegende deel geplaatst zie figuur 29. Hier wordt een eerste streep neergezet. Vervolgens wordt er door de software een verplaatsing op de lineaire geleider toegepast. Deze verplaatsing vertegenwoordigt in de software voor een aantal centimeter. Vervolgens wordt door middel van een meetlint geverifieerd of deze verplaatsing klopt.

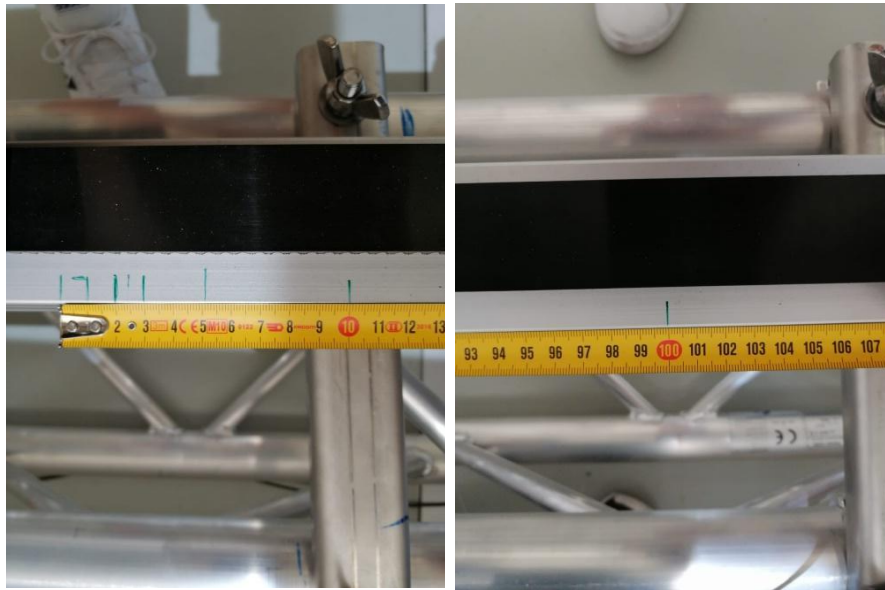


Afbeelding 29 De markering op de lineaire geleider met pen.

### 4.4.2 Resultaten

De resultaten zijn te zien in afbeelding 30, In deze afbeeldingen zijn de streepjes te zien die na het aansturen van de lineaire geleider gemaakt zijn. In de software is de lineaire module aangestuurd naar 1 cm, 2 cm, 3 cm, 5 cm, 10 cm, en naar 100 cm. Deze volgorde is in de afbeelding ook aangehouden.





Afbeelding 30 Resultaten van de test

#### 4.4.3 Conclusie

Aan de hand van deze resultaten kan er geconcludeerd worden dat de lineaire module zich kan verplaatsen volgens de specificaties. Op de foto's is goed te zien dat de module zich binnen de millimeter zich kan verplaatsen.

## 5. Realisatiefase

In dit hoofdstuk worden de specificaties en de opleverproducten geverifieerd. Daarnaast wordt het gehele onderzoek geconcludeerd. Verder wordt er een aanbeveling gedaan voor het bereik, veiligheidsaspecten en het kinematische model.

### 5.1 Verificatie specificatie

Om het project met succes te halen zullen de specificaties worden geverifieerd. In onderstaande tabel worden deze specificaties benoemd. Daarbij is ook weergegeven in welke paragraaf de specificatie is behaald.

Specificatie	Behaald	Referentie
(i) Het aansturen van de robotarm door middel van x, y en z coördinaten, de nauwkeurigheid valt binnen de millimeter.	Ja	Zie <b>4.3 Test van de kinematica</b>
(ii) Het aansturen en positionering van de lineaire geleider, de nauwkeurigheid valt binnen de millimeter.	Ja	Zie <b>4.4 Test voor de aansturing van de lineaire geleider</b>
(iii) Het integreren van safety aspecten voor het beveiligen van de testopstelling binnen de Beckhoff software.	Ja	Onderzoek zie <b>5.3 Veiligheid testopstelling</b> Integratie zie <b>4.2.1 Elektrische detaillering</b>
(iv) De eindeffector van de robotarm moet zich op 50 tot 70 centimeter van de leading edge kunnen plaatsen.	Ja	Onderzoek bereik: <b>3.1 bereik robotarm</b> Positionering robot (zie <b>paragraaf 4.3 en 4.4</b> )
(v) De robot mag bij de testopstelling geen botsing veroorzaken. De robot zal een error geven als de operator de opdracht geeft waar een botsing kan ontstaan.	Ja	Dit is geïntegreerd in de software. Door middel van de forward kinematics weet de robot waar die is en kan daarmee de obstakels definiëren. De forward kinematica is te vinden in paragraaf <b>2.5.3 Forward kinematica</b>
(vi) Een goed kinetica model waarin de robot vloeiend beweegt, zodat het image processing 'goed' kan verlopen.	Ja	Zie <b>paragraaf 2.6 Kinetica</b>
(vii) Een aanbeveling voor het eventueel gebruik van collision avoidance en power limitation voor de testopstelling.	Ja	Onderzoek zie paragraaf <b>2.4 veiligheid testopstelling</b> Zie <b>5.4 Aanbeveling</b>

### 5.2 Oplever producten

De op voorhand opgestelde oplever producten zijn behaald. Hier is een overzicht hoe deze producten zijn behaald.

Oplever product	Behaald
<ul style="list-style-type: none"> <li>De testopstelling moet naar x,y z en theta kunnen verplaatsen.</li> </ul>	<b>Paragraaf 4.3 en 4.4</b>
<ul style="list-style-type: none"> <li>Er moet een kinematisch model worden opgeleverd, die de configuratie van de leading edge scanner wiskundig weergeeft.</li> </ul>	<b>Paragraaf 2.5</b>
<ul style="list-style-type: none"> <li>De optimale kinematische structuur die ervoor zorgt dat het bereik van de leading edge scanner optimaal wordt benut.</li> </ul>	<b>Zie 5.4 aanbeveling</b>
<ul style="list-style-type: none"> <li>Een werkende demo waarbij de robot zich kan positioneren om de leading edge in kaart te brengen.</li> </ul>	<b>Er is een video met een demo gemaakt</b>
<ul style="list-style-type: none"> <li>Een aanbeveling voor de safety aspecten van het project.</li> </ul>	<b>Bijlage A en 5.4 aanbeveling</b>
<ul style="list-style-type: none"> <li>Er moet een keuze worden gemaakt voor de aansturing van de lineaire geleider, zodat deze verticaal geplaatst kan worden.</li> </ul>	<b>Paragraaf 3.2</b>

### 5.3 Conclusie

In dit verslag zijn deelvragen beantwoord, die voortgekomen zijn uit de hoofdvraag. De kinematica modellen die zijn opgesteld worden gebruikt om de robot aan te sturen. Deze modellen zijn geïmplementeerd in TwinCAT. Uit de testen is voortgekomen dat hiermee posities behaald kunnen worden.

Voor de aansturing van de lineaire geleider is een goede oplossing bedacht. Uit het onderzoek zijn meerdere oplossingen naar voren gekomen waarna er met een keuzematrix gekozen is voor een servomotor met een brake. Met deze motor kan de robotarm zich goed verticaal verplaatsen. Ook wordt de as geremd en zal de robotarm niet naar beneden vallen als de spanning wegvalt. Uit de testen is gekomen dat de eindposities binnen de millimeter behaald kunnen worden.

Om de leading edge zo goed mogelijk in kaart te brengen zijn verschillende opties meegenomen in het ontwerp van de testopstelling. Hierbij is nu een afstand gekozen waarbij de ronding van de leading edge goed in kaart gebracht kan worden.

Voor de dynamica van de robot zijn de bewegingen geanalyseerd. Hieruit is voortgekomen dat er bij dit systeem het best bewogen kan worden met een cycloïdale beweging. Door deze beweging is te zien dat de robot zich soepel en fijn beweegt.

Voor de safety is er veel onderzoek gedaan voor een veilige testopstelling. Daarbij is collision avoidance onderzocht en hieruit kan geconcludeerd worden dat dit meer van toepassing zal zijn als er variabele obstakels zijn. Toch kan dit meegenomen worden als de testopstelling meer naar een robot mens applicatie gaat. Ook kan er een fundamentele basis gelegd worden voor veiligheidseisen als dit systeem in de toekomst op de markt wordt gebracht. Hiervoor zal wel meer onderzoek nodig zijn. Deze veiligheidseisen zullen nog wel geïmplementeerd moeten worden. Tot slot kan de power limitation toegepast worden in het systeem. Hiervoor is enig onderzoek gedaan, maar om power limitation te implementeren zal er nog meer onderzoek nodig zijn.

Tot slot het antwoord op de hoofdvraag: *“Hoe kan motion control in de huidige testopstelling worden geïmplementeerd, zodat de testopstelling functioneert naar de opdrachten van de operator?”*. Dit wordt gedaan door de gemaakte berekeningen, die gesimuleerd zijn. Deze berekeningen zijn te implementeren in de TWINCAT controller van Beckhoff. Hierbij is het wel belangrijk dat alle veiligheidseisen zijn meegenomen in het proces. Doordat de kinematische structuur van de testopstelling is aangepast kan een groot gedeelte van de leading edge kaart worden gebracht.

## 5.4 Aanbeveling

Nu het project is afgerond zijn er nog wel een enkele verbeterpunten voor de testopstelling. De opstelling werkt en er kan naar posities bewogen worden. Toch kan er gekeken worden naar een andere vormgeving voor de positionering. Op dit moment wordt de positie aangenomen door middel van een x-, y- en een z- positie met hoek  $\varphi$  over de x-as. Hierbij is de hoekverdraaiing op de y as mogelijk maar niet meegenomen in het kinematische model, omdat dit niet nodig was voor de testopstelling. Het format zal dan veranderen in een X-, RX-, Y-, RY-, Z- en RZ- posities in plaats van X, Y, Z en  $\varphi$ . Waarbij hoek  $\varphi$  de rotatie is ten opzichte van de Z-as. Echter, dit is voor de huidige testopstelling (waarschijnlijk) niet nodig. Maar mochten er problemen komen waarmee er reflecties in het beeld komen dan kan dit worden toegepast.

Ook kan er meer onderzoek gedaan worden naar de kinematische structuur van de testopstelling. Op dit moment kan de leading edge grotendeels in beeld worden gebracht. Echter, is de afstand tussen de lineaire geleider en de leading edge nog niet optimaal. Er kunnen meer testen worden gedaan om de ideale afstand te bepalen.

Als laatste kunnen de veiligheidseisen geïmplementeerd worden, zodat deze testopstelling in de toekomst op de markt kan worden gebracht. Er zal hiervoor nog wel enig onderzoek nodig zijn om aan alle regels van de machine veiligheid te kunnen voldoen.

## 6. Literatuurlijst

- [1] "NLR," Netherlands Aerospace Centre, 08-May-2020. [Online]. Available: <https://www.nlr.nl/>. [Accessed: 28-Feb-2020].
- [2] "Ontwerp een gearticuleerde robot op schaal," N. van den Hout, 2013. Available: [http://essay.utwente.nl/63822/1/niekvandenhout\\_bachelorverslag.pdf](http://essay.utwente.nl/63822/1/niekvandenhout_bachelorverslag.pdf) [Accessed: 28-Feb-2020].
- [3] "Semantically enriching point clouds the case of street level", G.R. Veldkamp, 2013 . Available: ["https://pure.tue.nl/ws/portalfiles/portal/2407423/82208.pdf"](https://pure.tue.nl/ws/portalfiles/portal/2407423/82208.pdf) [Accessed: 28-Feb-2020].
- [4] "Kinematics modeling of a 4 DOF robotic arm", Mehmet Sunar, [Online]. Available: [https://Mohammed\\_and\\_Sunar\\_2015\\_Kinematics\\_modeling\\_of\\_a-4-DOF\\_robotic\\_arm.pdf](https://Mohammed_and_Sunar_2015_Kinematics_modeling_of_a-4-DOF_robotic_arm.pdf) [Accessed: 28-Feb-2020].
- [5] "Kinetica (dynamica)," Wikipedia, 11-Jun-2013. [Online]. Available: [https://nl.wikipedia.org/wiki/Kinetica\\_\(dynamica\)](https://nl.wikipedia.org/wiki/Kinetica_(dynamica)). [Accessed: 28-May-2020].:
- [6] D. Collins, "How to reduce jerk in linear motion systems," Linear Motion Tips, 30-Sep-2019. [Online]. Available: <https://www.linearmotiontips.com/how-to-reduce-jerk-in-linear-motion-systems/>. [Accessed: 28-May-2020].
- [7] "S-Curve Motion Profiles - Vital For Optimizing Machine Performance," Lewin Performance Motion Devices. [Online]. Available: <https://www.pmdcorp.com/resources/type/articles/get/s-curve-profiles-deep-dive-article>. [Accessed: 28-May-2020].
- 8: "MBD-Dyn7-Kinematica", Ko Hartog, 2020 . Not available: <https://dlo.mijnhva.nl/d2l/le/content/45036/viewContent/195837/View>
- [9] "robolink® - modulair systeem voor robotica", IGUS [Online]. Available: <https://www.igus.nl/robolink/robot>. [Accessed: 28-May-2020].
- 10: "Control of Mobile Robots- 2.8 Obstacle Avoidance [online] Available: ["https://www.youtube.com/watch?v=bsdBFHMSI2E&t=470s"](https://www.youtube.com/watch?v=bsdBFHMSI2E&t=470s) [Accessed: 28-May-2020].
- [11] "Collaborative Robots – Power and Force Limiting", Björn Matthias [Online]. Available: <https://www.robotics.org/userAssets/riaUploads/file/7-Bjorn.pdf> [Accessed: 28-May-2020].
- [12] "leading edge scanner- stageverslag", B Mous, 2019. [Accessed: 28-May-2020].
- [13] "Linmotion tandriem modules," <https://www.lmsystems.nl/wp-content/uploads/2017/06/LinMotion-LME-tandriem-modules.pdf>. .
- [14] "Beckhoff Information System" [Online]. Available: <https://infosys.beckhoff.com/english.php?content=../content/1033/el73x2/2075577355.html&id=>. [Accessed: 28-May-2020].
- [15] "ARVI Point Cloud Image Processing Assignment", [Accessed: 11-Jun-2019].
- [16] "ARVI Image Processing Internship Report", R pope, 2016. [Accessed: 09-Jan-2019].
- [17] "Normaalvector," Wikipedia, 13-Jan-2019. [Online]. Available: <https://nl.wikipedia.org/wiki/Normaalvector>. [Accessed: 28-May-2020].

- [18] "ADS communicatie in TwinCAT 3 tussen 2 controllers," Beckhoff.nl, 11-Jul-2019. [Online]. Available: [https://www.google.com/search?q=ADS communicatie in TwinCAT 3 tussen 2 controllers&rlz=1C1GCEA\\_enNL889NL889&oq=ADS communicatie in TwinCAT 3 tussen 2 controllers&aqs=chrome..69i57.1220j0j4&sourceid=chrome&ie=UTF-8](https://www.google.com/search?q=ADS+communicatie+in+TwinCAT+3+tussen+2+controllers&rlz=1C1GCEA_enNL889NL889&oq=ADS+communicatie+in+TwinCAT+3+tussen+2+controllers&aqs=chrome..69i57.1220j0j4&sourceid=chrome&ie=UTF-8). [Accessed: 20-Feb-2020].
- [19] "Robotics Toolbox | Peter Corke", Peter Corke, 2020. [Online]. Available: <https://petercorke.com/toolboxes/robotics-toolbox/>. [Accessed: 29- Apr- 2020].
- [20] "user manual\Kinematic Profile Configuration\_7.htm", Malinc.com, 2020. [Online]. Available: [https://www.malinc.com/wp-content/uploads/2014/12/VCNC/VirtualCNC/usermanual\\_kinematicprofileconfiguration\\_7.htm](https://www.malinc.com/wp-content/uploads/2014/12/VCNC/VirtualCNC/usermanual_kinematicprofileconfiguration_7.htm). [Accessed: 29- May- 2020].
- [21] "Machineveiligheid", NEN, 2020. [Online]. Available: <https://www.nen.nl/NEN-Shop/Machineveiligheid-1.htm>. [Accessed: 29- Feb- 2020].
- [22] "Artikel: Vergeten veiligheidsaspecten aspecten van een robot | D&F", Denf.nl, 2020. [Online]. Available: <https://www.denf.nl/artikelen/machineveiligheid/2016/vergeten-veiligheidsaspecten-aspecten-een-robot>. [Accessed: 29- Apr- 2020].
- [23] Pure.tue.nl, 2020. [Online]. Available: <https://pure.tue.nl/ws/files/4412125/463824.pdf>. [Accessed: 29- Feb- 2020].

## Bijlage A: Adviesrapport safety



Dedicated to innovation in aerospace





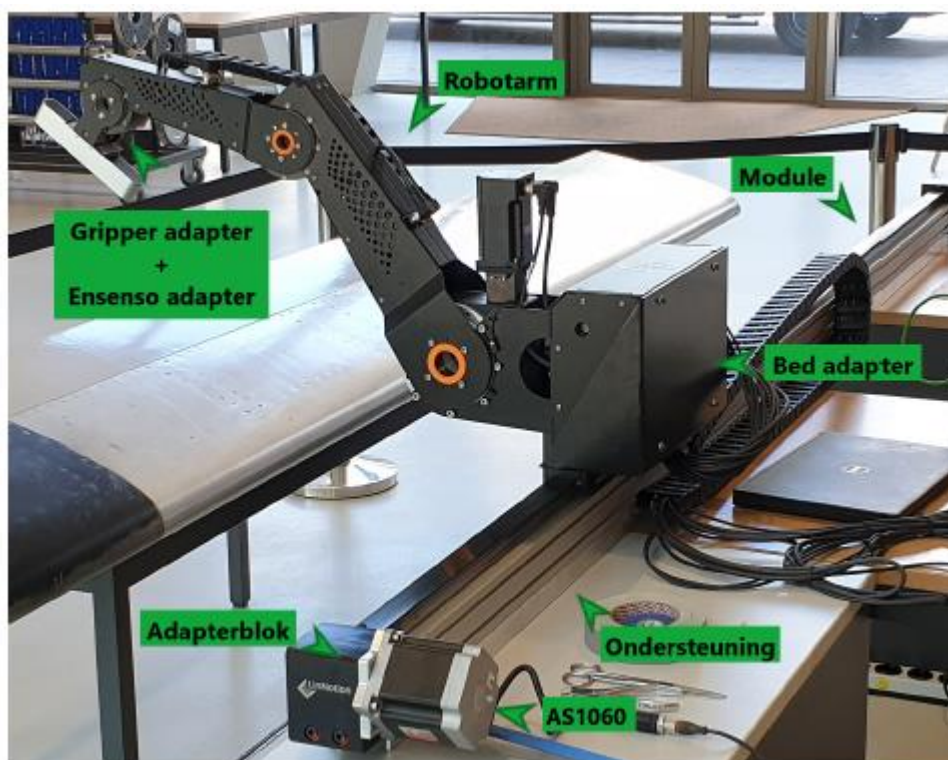
# Inhoudsopgave van Bijlage A

<b>1.</b>	<b>Introductie</b>	<b>47</b>
<b>2.</b>	<b>De veiligheidseisen</b>	<b>47</b>
2.1	De algemene norm	47
2.2	Gevaren inventarisatie testopstelling	48
<b>3.</b>	<b>Veiligheidsmaatregelen testopstelling</b>	<b>49</b>
3.1	Persoonlijke veiligheid	49
3.2	Mechanische veiligheid	49
3.3	De noodstop	49
3.4	Specificatie safety ARBI testopstelling	53
<b>4</b>	<b>Implementatie Noodstop circuit</b>	<b>54</b>
4.1	Bij een noodstop	51
4.1.1	Optie 1	51
4.1.2	Optie 2	52
4.2	Conclusie	53

# 1. Introductie

Het NLR ontwikkelt een testopstelling voor visuele inspecties van een voorvleugel. Deze testopstelling bevat veel bewegende delen in veel mogelijke richtingen. De testopstelling is een robot met 5 vrijheidsgraden (zie figuur 31). Het doel is om deze testopstelling in de toekomst op de markt te brengen. Daarom is het belangrijk om rekening te houden met richtlijnen die daarvoor voorgeschreven zijn. Het eerste doel van de testopstelling is om aan te tonen dat er defecten gedetecteerd kunnen worden. Daarom wordt er rekening gehouden met de machineveiligheid.

In dit document wordt er beschreven wat er nodig is om een veilige testopstelling te ontwerpen. Ook wordt er rekening gehouden met het op de markt brengen van een eventueel toekomstige inspectie robot. Hierbij zullen veel meer veiligheidseisen nodig zijn. Ook zullen er zaken aan bod komen zoals noodstop veiligheid, "collision avoidance", "Force and power limiting" en elektrische veiligheidseisen.



Afbeelding 31 De testopstelling

## 2. De veiligheidseisen

Een robot is een machine en moet daarom voldoen aan dezelfde veiligheidseisen als machines. Dit is de machinerichtlijn veiligheidsnorm. Deze is zeer abstract en uitgebreid. Om de inspectierobot ooit op de markt te brengen moet hier zeker aan worden voldaan. Voor de testopstelling kan er aan andere veiligheidseisen worden voldaan.

### 2.1 De algemene norm

De veiligheidseisen wordt door de NEN gespecificeerd. De belangrijkste verplichtingen vanuit de NEN [21] zijn:

- I. Machines moeten zodanig zijn ontworpen en geproduceerd, dat zij voldoen aan de fundamentele of essentiële eisen over machineveiligheid.
- II. Er dient een Technisch constructiedossier te worden samengesteld.
- III. Daarnaast dient een gebruiksaanwijzing of –handleiding te worden opgesteld en bij de machine te worden (mee)geleverd. Naast de “gewone” instructies inzake inbedrijfstelling, gebruik, etc., bevat de gebruiksaanwijzing tevens voorschriften voor het gebruik van de machine in situaties waarin gevaren kunnen optreden, welke niet op een andere manier, bijvoorbeeld door middel van een technische beveiliging, kunnen worden weggenomen.
- IV. Voorts dient een EG-verklaring van overeenstemming te worden opgesteld. Dit is de verklaring van de fabrikant van de machine, dat hij aan alle relevante eisen uit alle toepasselijke CE-markeringsrichtlijnen heeft voldaan.
- V. Er moet een CE-markering worden aangebracht.

Omdat de inspectierobot in het begin van het ontwerpproces zit. Is het voor nu belangrijk om op punt (I) te focussen. De norm van de NEN betrekking tot machineveiligheid is de EN-ISO10218-1. Deze norm is ingewikkeld en uitgebreid. In de het eerste deel is van toepassing voor de ontwerper van de “kale” industriële robot [22]. In het geval van de testopstelling zal dit gedekt zijn door de fabrikanten van de robotarm (igus) en de lineaire as (linmotion). Het tweede deel van deze norm geldt voor de zogeheten integrator/ machinebouwer. Hierin staan zaken als de veilige integratie van industriële robots en robotsystemen. De belangrijkste zaken uit deze wetgeving zijn als volgt op prioriteit gerangschikt [23] :

- I. **Eliminatie:** Hierbij wordt het gevaar in zijn geheel weggenomen en kan dus niet meer optreden.
- II. **Preventie:** Men weet dat het gevaar kan optreden, maar probeert de kans hierop te verkleinen.
- III. **Detectie:** Het gevaar treedt op maar men wordt tijdig gewaarschuwd, waarna een correctieve actie kan plaatsvinden.
- IV. **Correcte:** Het gevaar treedt op, maar wordt alsnog tijdig voorkomen. Deze actie kan pas na detectie plaatsvinden.
- V. **Protectie:** Het gevaar treedt op en is niet meer te voorkomen. De grootte van de schade wordt hiermee beperkt.
- VI. **Reparatie:** Het gevaar heeft schade aangericht. De schade wordt hersteld.

Hierin kan er veiligheidsmaatregel genomen worden. Deze veiligheidsmaatregelen kunnen in twee categorieën onderverdelen. Persoonlijke veiligheid en functionele/mechanische veiligheid.

## 2.2 Gevaren inventarisatie testopstelling

Het grootste gevaar voor de testopstelling is beknelling of aanraking van de robot met de mens. Hierbij zullen er diverse maatregelen worden genomen om dit te voorkomen. Ook is het belangrijk dat bij een noodstop de assen geremd worden zodat de lineaire geleider niet naar beneden valt en er vervolgens een beklemming ontstaat. Een ander gevaar is de mechanische veiligheid. Daarbij wordt er gedacht aan een botsing tussen de leading edge en de robot. Ook moet de testopstelling voldoen aan de elektrische veiligheid eisen. Daarbij kan er gedacht worden aan fail-to-safe schakelingen van het noodstopcircuit. Dit houdt in dat de schakeling dubbel wordt uitgevoerd en deze wordt gecontroleerd op kortsluitingen

### 3. Veiligheidsmaatregelen ARBI testopstelling

Het elimineren van de gevaren bij deze testopstelling zal niet mogelijk zijn. Wel is het belangrijk om de gevaren te voorkomen. Als deze voorzorgsmaatregelen niet voldoende zijn zullen er extra correctie veiligheidsmaatregelen zijn.

#### 3.1 Persoonlijke veiligheid

Om de beknelling tussen de mens en de robot te voorkomen zal het werkgebied van de robot visueel worden afgebakend. Ook zal de testopstelling worden voorzien van voldoende gevarenborden, zodat het voor iedereen duidelijk zal zijn wat het “gevaarlijke gebied” is. In het ontwerp wordt ervoor gezorgd dat niemand in het werkgebied van de robot zal komen. Denk hierbij aan dat de besturingskast buiten het werkgebied geplaatst zal worden. Maar ook de werkschakelaar, de programmeer omgeving en de besturings omgeving. Mocht er een gevaarlijke situatie voorkomen zal er een noodstop schakelaar in de testopstelling geïmplementeerd worden. De bediening van deze noodstop kan binnen het werkgebied van de robot. Deze kan hier gebruikt worden, omdat als er door menselijk toedoen toch de gevarezone wordt betreden, de noodstop alsnog bediend kan worden. Ook komt er een nood-stop buiten het bereik van de robot, zodat als er iemand bekneld deze persoon alsnog geholpen kan worden. Mochten al deze voorzorgsmaatregelen niet voldoende zijn zal de testopstelling gebruik maken van een “force and power limiting”. Dit houdt in dat als er bij menselijk falen er toch contact wordt gemaakt met de robot. De robot een onverwachte kracht piek zal detecteren. De noodstop zal hier op volgen.

#### 3.2 Mechanische veiligheid

Om de mechanische veiligheid te waarborgen zal “collision avoidance” worden toegepast. Dit zal gebeuren in de twincat omgeving van Beckhoff. Hierbij zal er in de software een no-go zone worden toegepast. Dit is een zone waar geen enkel deel van de robot mag komen. Verder onderzoek hierna zal volgen. Maar op dit moment zal de no-go zone worden berekend door middel van de Forward kinematics. Dit is de kinematics die een eindpositie berekent met als input de motor hoeken.

#### 3.3 De noodstop

De noodstop is een situatie die eigenlijk nooit gebruikt gaat worden. Maar als deze gebruikt wordt moet het wel voor zorgen dat alles klopt. Hiervoor zal de schakeling van de noodstop fail to safe worden toegepast. Een fail to safe schakeling een beveiliging. Als er een defect is in het systeem is, zal het systeem alsnog uitschakelen. Hierbij worden contact dubbel-polig aangesloten en worden er “normally closed” contacten gebruikt. Als de nood-stop in gaat is het belangrijkste dat alle bewegende delen direct stoppen. De noodstop zal er dus voor zorgen dat alle motor spanning direct worden uitgeschakeld.

### 3.4 Specificatie safety ARBI testopstelling

Het is belangrijk om aan alle veiligheidseisen voor de testopstelling te voldoen. Op dit moment kan er gespecificeerd worden wat de prioriteiten zijn voor het ontwerp van het safety aspect voor de testopstelling. Voor de meeste onderdelen is het belangrijk om te voldoen aan de norm.

Categorie	Specificatie	
Must (M)	I.	Bij een noodstop voldoet de testopstelling aan de eisen betreft machineveiligheid.
	II.	De gehele reikwijdte van de robot is afgebakend.
	III.	Als de robot aangestuurd wordt naar de no-go zone. Zal er een storing optreden en wordt de robot naar een veilige positie verplaatst.
Should (S)	IV.	De robot voldoet aan collision avoidance.
Could (C)	V.	De aansturing en de aansluiting voldoet aan de machine veiligheid.
	VI.	De robot voldoet aan force and power limiting.
Won't (W)	IV.	De testopstelling voldoet (nog) niet aan de alle verplichtingen van de NEN. (zie hoofdstuk 2.1 punt II t/m punt V).
	V.	Er wordt gedetecteerd wanneer een persoon binnen de reikwijdte van de robot komt. Er volgt een nood-stop.

## 4 Implementatie Noodstop circuit

Het doel van de noodstop circuit is dat er bij gevaren alle bewegende delen direct stoppen. In eerste instantie worden deze gevaren benaderd door middel van de twee noodstop schakelaars. In de toekomst kunnen hier inputs bijkomen. Zoals het gebruik van een lichtgordijn of een deurschakelaar.

### 4.1 Bij een noodstop

Om alle bewegende delen te stoppen zijn er meerdere mogelijkheden. Alle bewegende delen van de robot worden aangestuurd door stepper motoren. Deze zitten vervolgens op de motor controllers van de Beckhoff PLC. Als eerste kunnen de motoren gestopt worden door deze niet meer aan te sturen in de Beckhoff software. Dit zal gebruikt worden bij de collision avoidance en de force limitation methodes. Echter om te functioneren als een nood-stop zal dit niet voldoen volgens de machinerichtlijn. Daarvoor zijn er drie mogelijkheden.

- (i) De gehele PLC kan uitgeschakeld worden.
- (ii) De voeding van de I/O kaarten kan onderbroken worden.
- (iii) De voeding van de motor kan onderbroken worden.

Bij optie (i) en (ii) kan de PLC in storing treden of de PLC zal uitgaan. Bovendien is het in dat geval niet duidelijk waarom de noodstop is ingegaan. Hierbij gaat de voorkeur uit naar optie (iii) omdat het zeker is dat de motoren gestopt worden en de PLC zal nog in bedrijf zijn en zal kunnen aangeven waarom en waar er een gevaar optreedt. In dit geval moet er rekening gehouden worden met de stroom van de motoren. De piekstroom van de motoren variëren tussen 1,8 [A] en de 5 [A].

In de toekomst zal er gekeken kunnen worden naar andere motor drivers van Beckhoff. Er zijn bijvoorbeeld ook motor drivers met een ingebouwde "Safe torque off". Echter hebben de huidige motor drivers deze optie niet en zijn deze vrij kostbaar.

#### 4.1.1 Optie 1: noodstop doormiddel van twinsafe

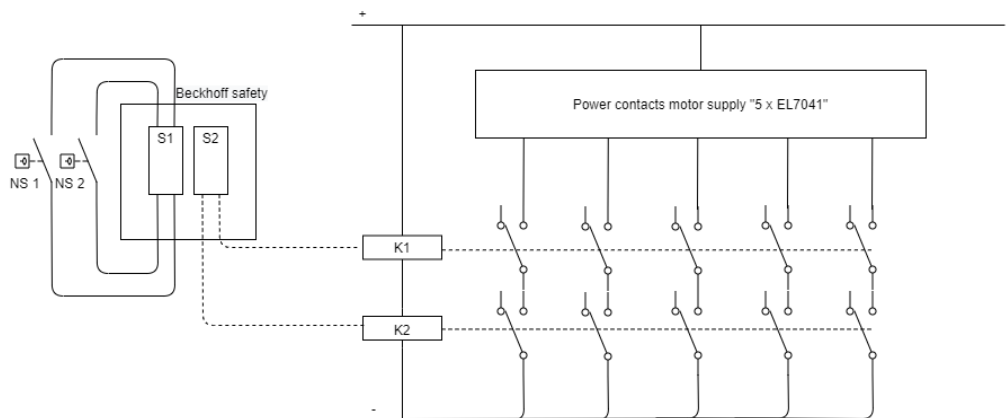
Dit is de oplossing door het gebruik maken van de twinsafe modules. Deze modules zijn een uitbreiding van de Beckhoff controller. Hierbij worden de voeding van de motoren gehaald door een relais. Dit relais wordt bekrachtigd door de twinsafe module. De inputs zullen binnen komen op de twinsafe input module. De voordelen van deze optie is dat het systeem weet dat er een fout is opgetreden. Waarbij het systeem opnieuw opgestart kan worden zonder dat het hele systeem gereset wordt. Ook voldoet deze optie aan de machineveiligheid richtlijnen. Omdat de twinsafe modules hiervoor ontworpen zijn.

##### **Kosten optie 1:**

Twinsafe modules 385,- euro

Relays 150,- euro

Totaal: 535,- euro

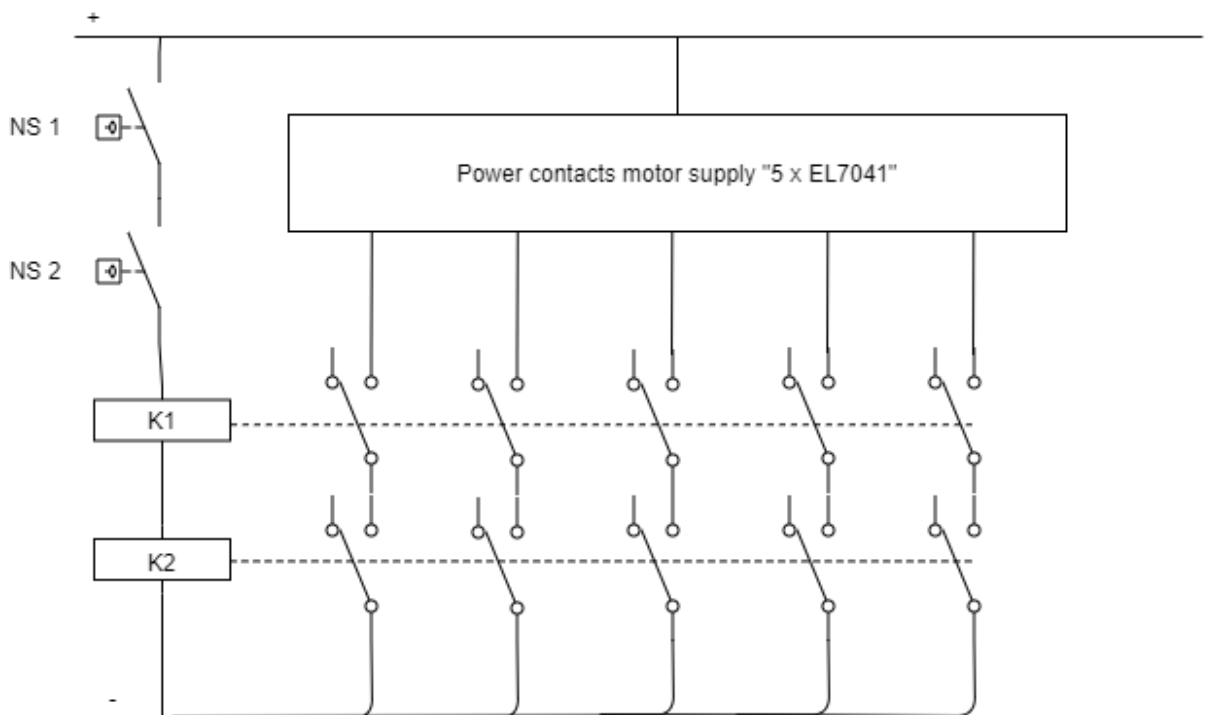


#### 4.1.2 Optie 2: noodstop volledig hardwarematig

Deze optie is een stuk goedkoper. Hiervoor hoeven er alleen de relais aangeschaft te worden. Het nadeel is dat dit niet voldoet aan de richtlijnen voor machineveiligheid. Echter kan dit met een kleine aanpassing wel aan deze eisen voldoen. Ook zal de PLC nog niet weten of er een noodstop is. Daarvoor zal tijdens de testen het systeem handmatig gereset worden.

##### Kosten optie 2:

Relais 150,- euro





## 4.2 Conclusie

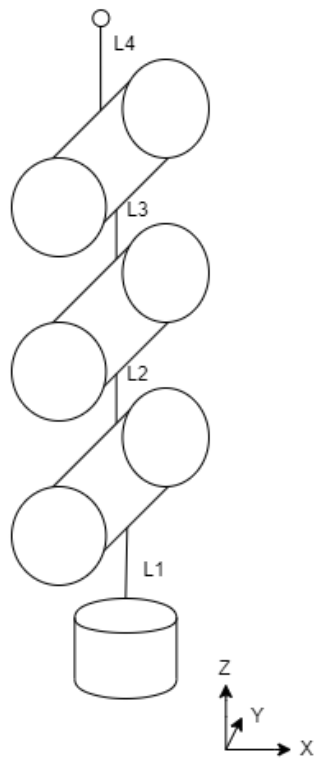
Op dit moment kunnen we ons sterk afvragen wat de toegevoegde waarde is van de Twincat modules. Het voordeel is dat het voldoet conform de veiligheidseisen voor machineveiligheid. Ook kunnen we de Twincat modules gebruiken voor onze trainingsdoeleinde. Op dit moment van de ontwikkeling voor de testopstelling zijn deze Twincat modules nog niet noodzakelijk. De investering kan in een later stadium van de ontwikkeling plaatsvinden. Daarom is er gekozen voor de tweede optie. Zodat de kosten voor de ontwikkeling op dit moment lager uitkomen.

## Bijlage B: Reverse kinematics

In deze bijlage zijn de berekeningen weergegeven die tot de reverse kinematics lijden. Deze kinematica is herleid met behulp van goniometrie. Er wordt vanuit de “side view” en de “top view” de hoeken in de x- y- en z- richting bepaald. Vanuit de “top view” kan de z berekening bepaald worden. De X, en de Y componenten worden bepaald door side en de top view. De oriëntatie van de robot is afhankelijk van de hoeken van joint 2, joint 3, en joint 4. Deze zijn samen 180 graden. Uiteindelijk volgen er 11 formules. Deze formule zijn door middel van substitutie vereenvoudigd tot de volgende formules:

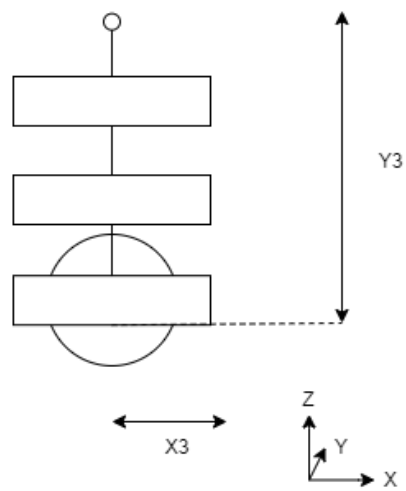
$$\begin{aligned}\Phi 1 &= \text{atan2}(y, x) \\ \Phi 2 &= \text{atan2}(z - l1 - l3 * \sin \theta, \pm \sqrt{(\sqrt{l3 * \sin \Phi 3})^2 + (l2 + l3 * \cos \Phi 3)^2 - (z - l1 - l4 * \sin \theta)^2}) - \text{atan2}(l3 * \sin \Phi 3, (l2 + l3 * \cos \Phi 3)) \\ \Phi 3 &= \text{acos}(A^2 + B^2 + C^2 - l2^2 - l3^2) / (2 * l2 * l3) \\ \Phi 4 &= \theta - \Phi 2 - \Phi 3 \\ A &= x - l4 * \cos \Phi 1 * \cos \theta \\ B &= y - l4 * \sin \Phi 1 * \cos \theta \\ C &= z - l1 - l4 * \sin \theta\end{aligned}$$

# Reverse analyse



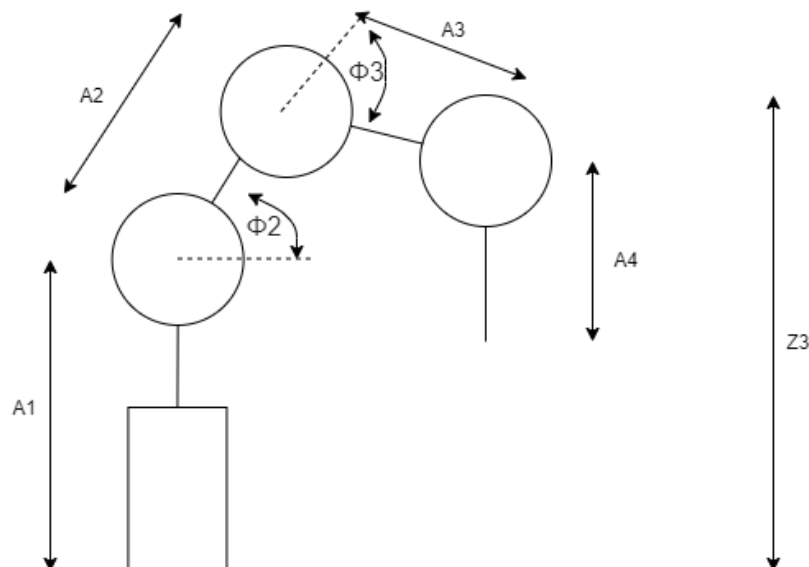
"Top" view

$$\Phi 1 = \text{atan2} \left( \frac{Y_3}{X_3} \right)$$

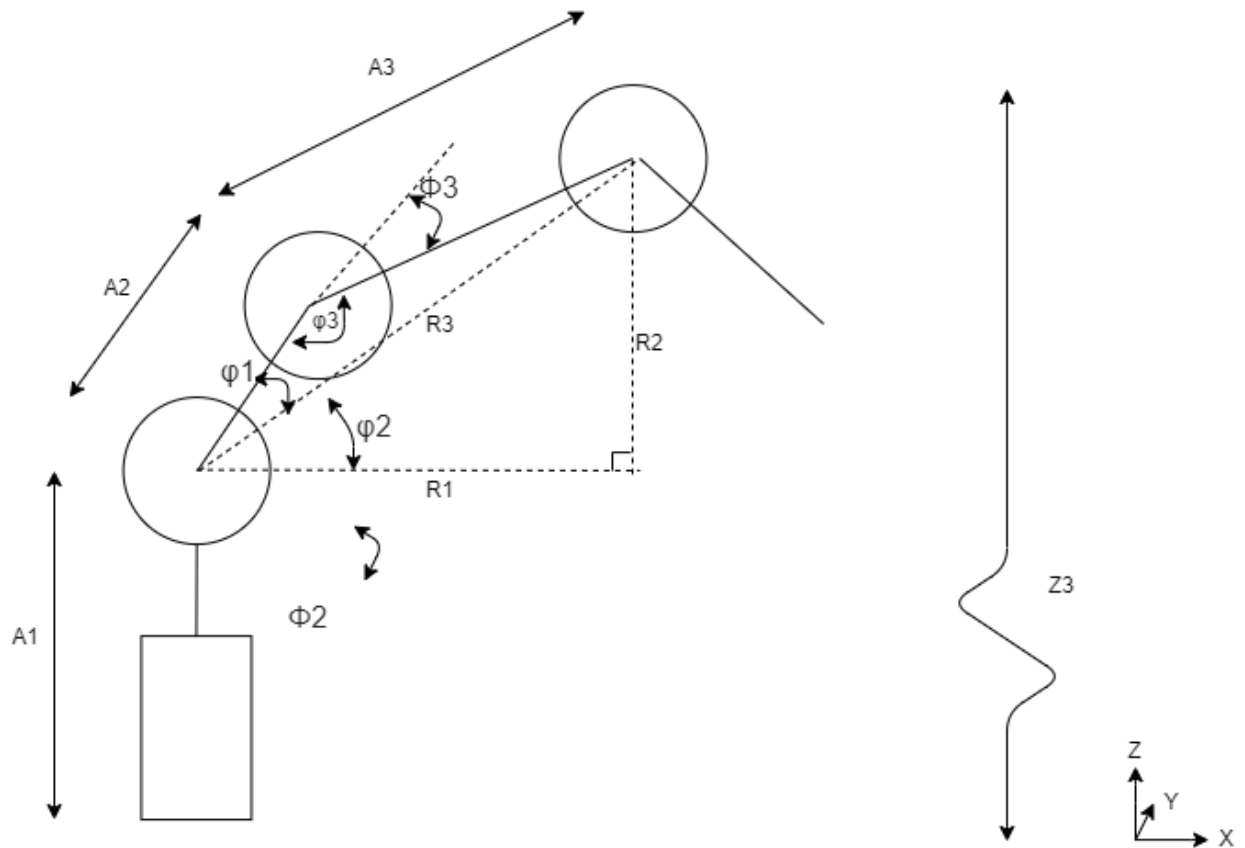


$$r1 = \sqrt{(x_3^2 + y_3^2)}$$

"side" view



"side" view



$$R3 = Z3 - A1$$

$$\Phi2 = \text{atan2}(r3, r2)$$

$$\varphi1 = \cos^{-1}\left(\frac{A3^2 - A2^2 - r3^2}{-2A2 * R3}\right)$$

$$r2 = \sqrt{(r1^2 + r2^2)}$$

$$\Phi3 = 180 - \varphi3$$

$$\varphi3 = \cos^{-1}\left(\frac{R3^2 - A2^2 - A3^2}{-2A2A3}\right)$$

$$\Phi4 = \theta - \Phi2 - \Phi3$$

$$\Phi1 = \text{atan2}(y, x)$$

$$\Phi2 = \text{atan2}(z - l1 - l3 * \sin\theta, \pm \sqrt{(\sqrt{l3 * \sin\Phi3})^2 + (l2 + l3 * \cos\Phi3)^2 - (z - l1 - l4 * \sin\theta)^2}) - \text{atan2}(l3 * \sin\Phi3, (l2 + l3 * \cos\Phi3))$$

$$\Phi3 = \text{acos}(A^2 + B^2 + C^2 - l2^2 - l3^2) / (2 * l2 * l3)$$

$$\Phi4 = \theta - \Phi2 - \Phi3$$

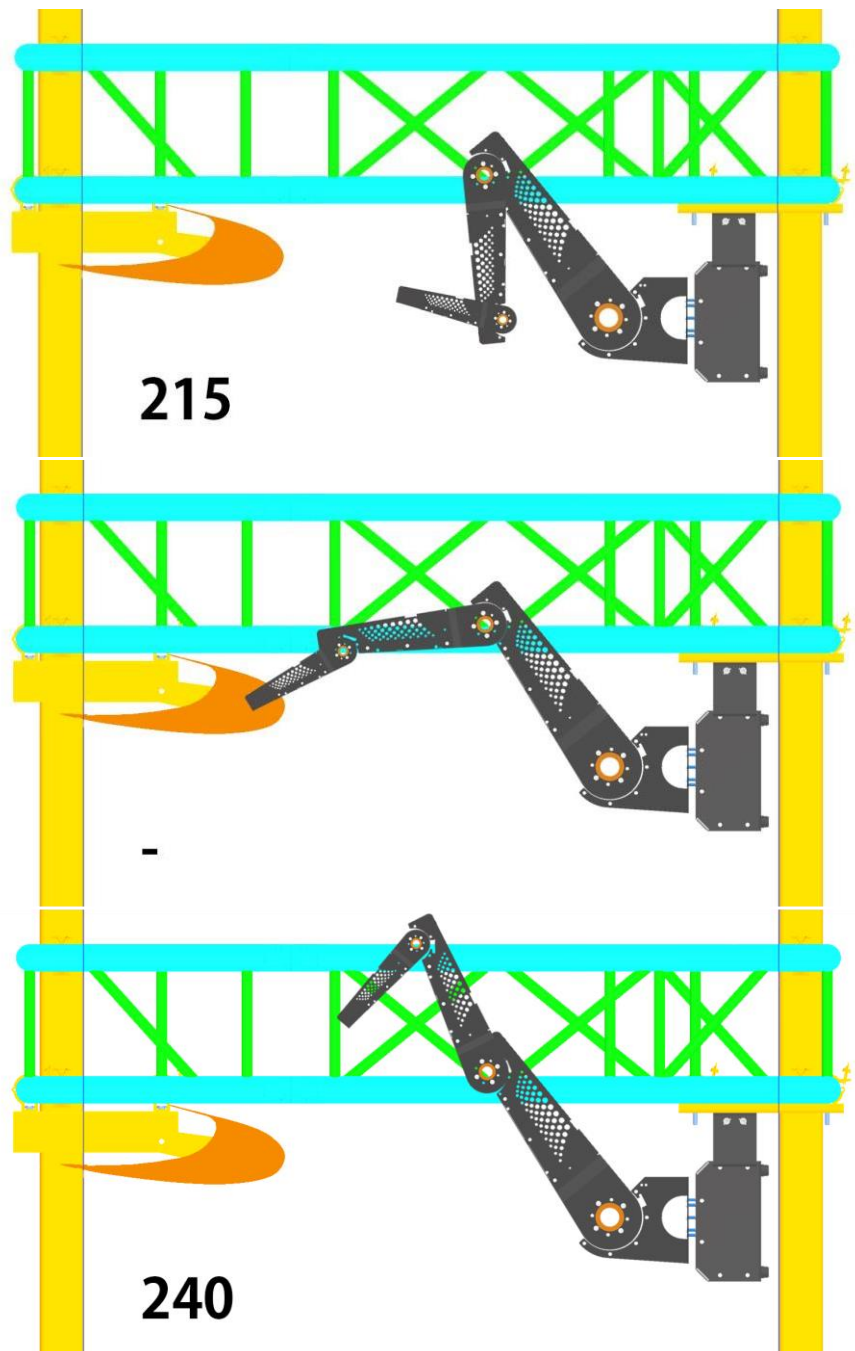
$$A = x - l4 * \cos\Phi1 * \cos\theta$$

$$B = y - l4 * \sin\Phi1 * \cos\theta$$

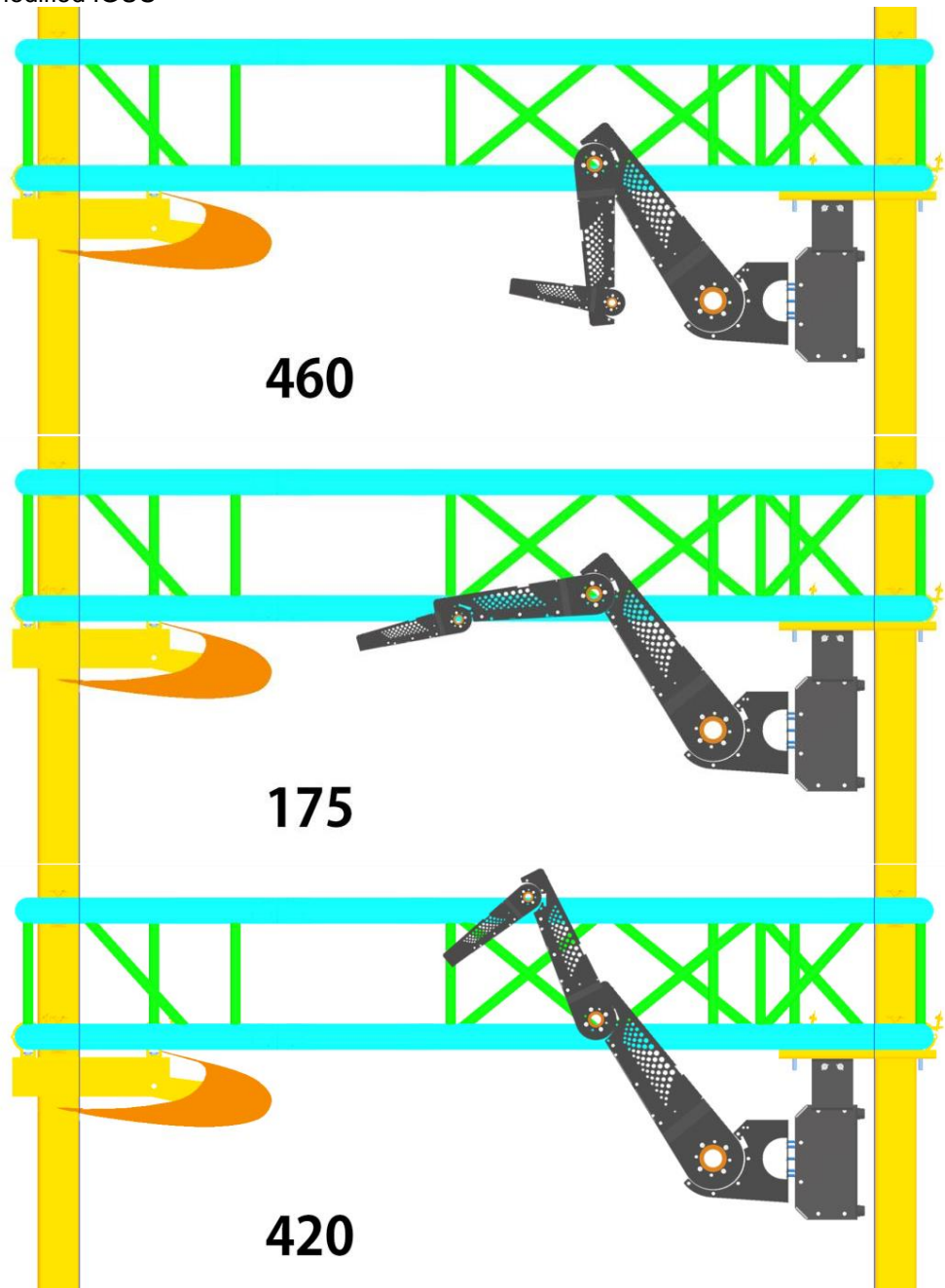
$$C = z - l1 - l4 * \sin\theta$$

## Bijlage C: Bereik robotarm

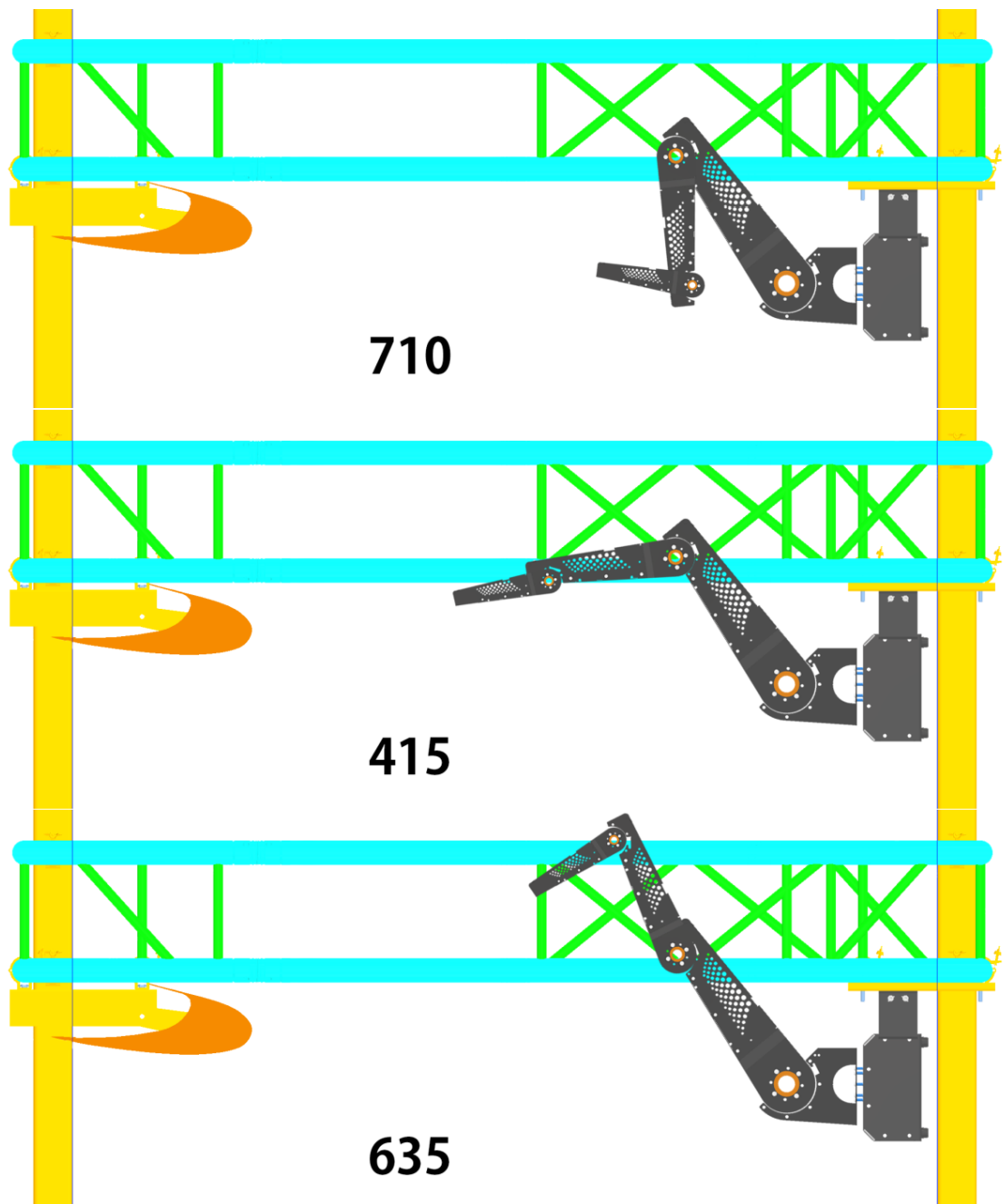
Truss +0, Unmodified IGUS



Truss +250, Unmodified IGUS

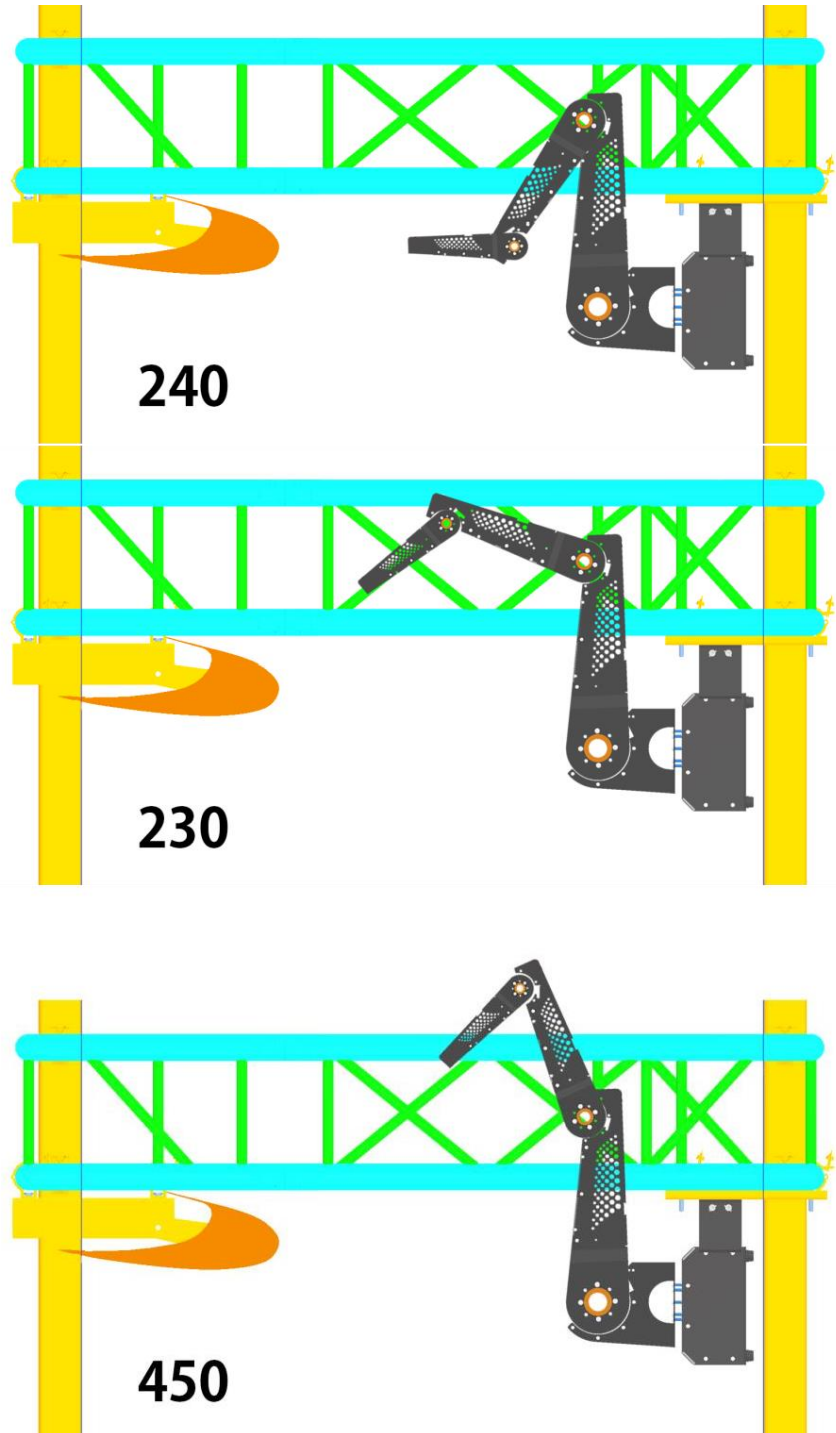


Truss +500, Unmodified IGUS

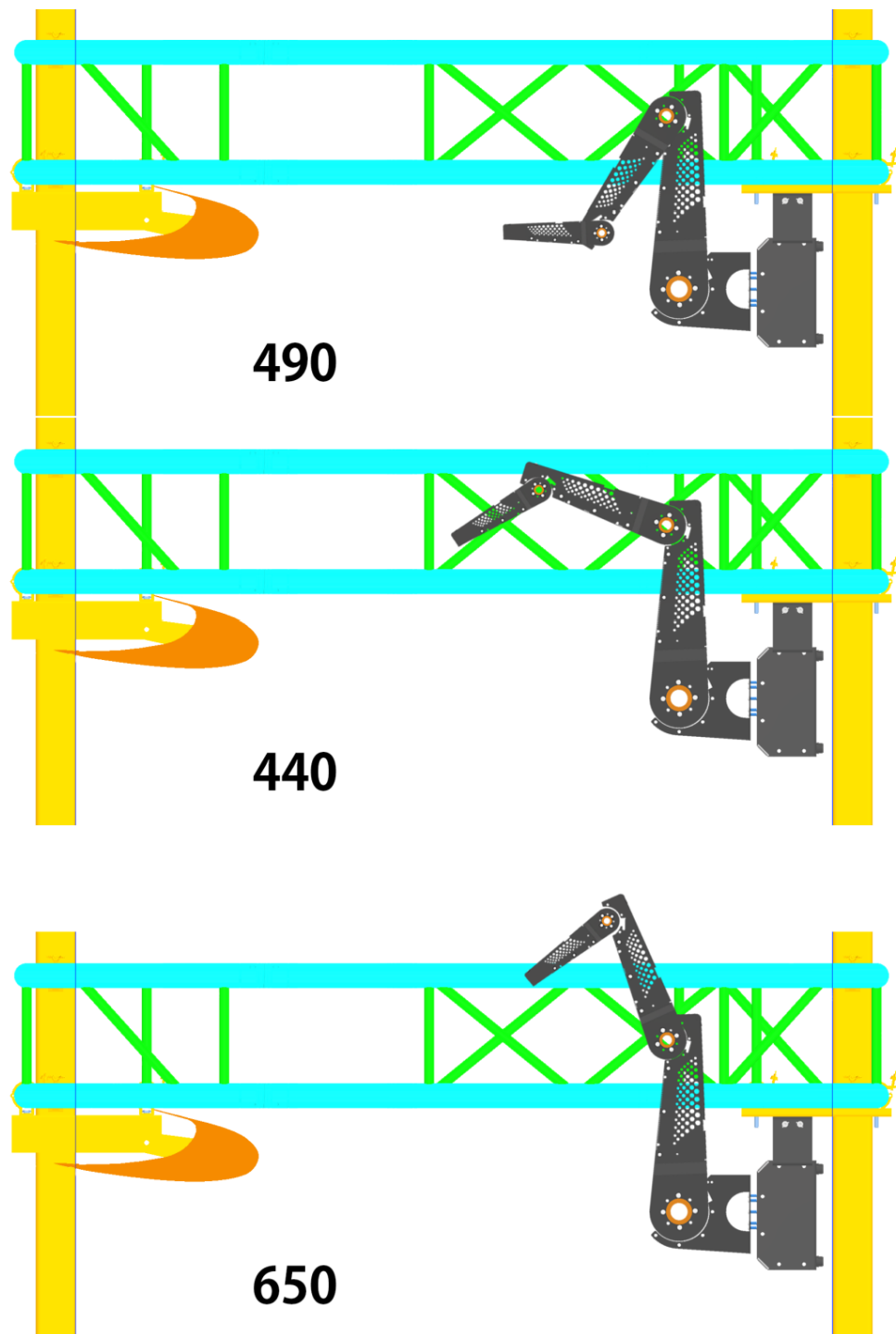




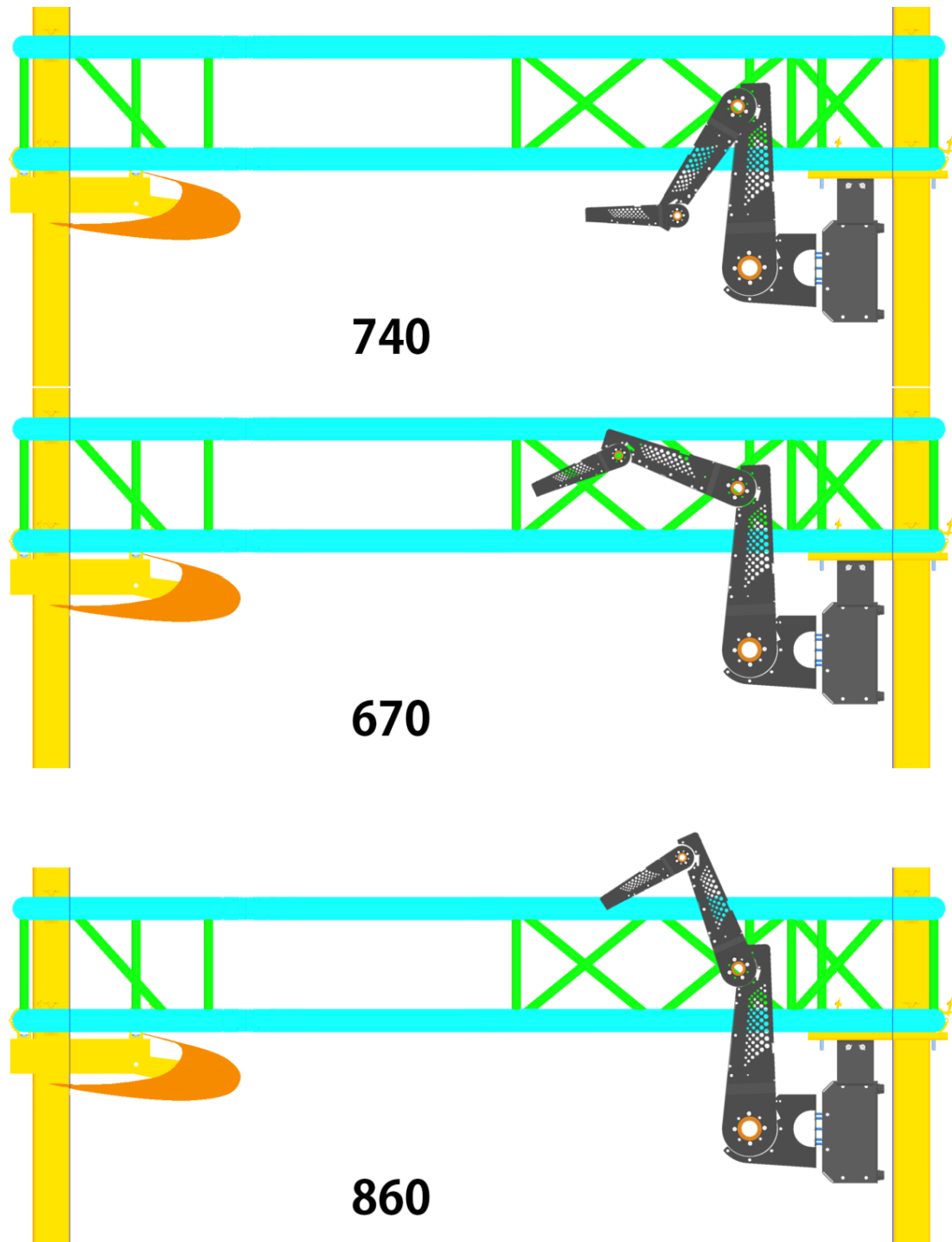
Truss +0, modified IGUS



Truss +250, modified IGUS



Truss +500, modified IGUS



# Bijlage D Software

```

1  PROGRAM MAIN
2  VAR
3      AxisA          : AXIS_REF ;           // axis for
linear module
4      AxisACS        : ARRAY [ 1..5 ] OF AXIS_REF ;           // axes
of the robot arm
5      FbAxes         : ARRAY [ 1..5 ] OF FB_AxisWithRefSwitch ; // ini
set switches for calibration of the arm.
6      UpperRefswitch AT %I* : BOOL ; // limit switch for AxisA
7      LowerRefswitch AT %I* : BOOL ; // limit switch for AxisA
8      idx            : INT ;
9
10     //////////////////////////////////////
11     btActivator      : BOOL := FALSE ; // activates configuration
12     btHoming         : BOOL := FALSE ; // starts homing/calibration
process.
13     btMoveStartPos   : BOOL := FALSE ; // movement to start position
14     btMoveSafePos    : BOOL := FALSE ; // movement to safe position in
order to displace the entire arm
15     //////////////////////////////////////
16     McMoveAbs        : ARRAY [ 1..30 ] OF MC_MoveAbsolute ; // absolute
movement, takes 0 position as reference
17     McMoveRel        : ARRAY [ 1..9 ] OF MC_MoveRelative ; //
relative movement compared to its current position
18     McPower          : ARRAY [ 1..15 ] OF MC_Power ; // sets
power active to the axis
19     McReset          : ARRAY [ 1..15 ] OF MC_Reset ; // resets
the axis
20     McJogging        : ARRAY [ 1..15 ] OF MC_Jog ; // enables
manual jogging of axis
21     McVelocity       : MC_MoveVelocity ; // determines
velocity of movement
22     McStop           : ARRAY [ 1..12 ] OF MC_Stop ; // stops
the axis
23     McHome           : ARRAY [ 1..12 ] OF MC_Home ; //
starts calibration process
24     //////////////////////////////////////
25     FbCalculateXYZ   : FbCalculateXYZ ;
26     X_coordinate     : LREAL ;
27     Y_coordinate     : LREAL ;
28     Z_coordinate     : LREAL ;
29     Graden_orientatie : LREAL ;
30     CalculatedJ1      : LREAL ;
31     CalculatedJ2      : LREAL ;
32     CalculatedJ3      : LREAL ;
33     CalculatedJ4      : LREAL ;
34     //////////////////////////////////////
35     JOG_Value_X_axis : DINT := 10 ;
36     JOG_Value_Y_axis : DINT := 10 ;

```

```
36     JOG_Value_Z_axis      : DINT := 10 ;
37     JOG_Value_phi_axis    : DINT := 5 ;
38
39     //////////////////////////////////////,
40     aStateDesc             : STRING ;           // used to describe
current state of state machine
41     aNoodstop              : STRING ;
42     iState                 : DINT ;           // number of the
state
43     MotorJog               : DINT ;
44     bError                 : BOOL ;           //
45     btResetfull            : BOOL := FALSE ;   // full reset of
program
46     btReturn               : BOOL := FALSE ;   // return to start
of program
47     StState                : INT ;
48     aJogDesk               : STRING ;         // used to describe
current state of JogMotor
49
50     //////////////////////////////////////,
51
52     END_VAR
53
```

---

```
1      (*
2
3      Project: Leading edge scanner
4      2020 Copyright (c) NLR. All rights reserved.
5      Author: Robin Ruijs
6
7      Contact
8      Email:      Robin.ruijs@NLR.nl / Robinruijs1@gmail.com
9      Tel:        +316 28369059
10
11     Contributor:
12     Timo Wassenaar
13
14
15     See my final raport for more information about the kinematics and develop
choices.
16     U:\MRO\maintenanceTechnology etc.....!!!!
17     For more information about TwinCAT or the programming language see
"squizybrained" on youtube
18
19     Good luck! And don't hasetate to contact me with any questions.
20
21     *)
22
```

```

23 // Stepper calculation tool squizybrained encoder
24
25 AxisA.ReadStatus();
26 AxisACS[1].ReadStatus();
27 AxisACS[2].ReadStatus();
28 AxisACS[3].ReadStatus();
29 AxisACS[4].ReadStatus();
30
31
32
33
34 McPower[5](Axis := AxisA, Enable := TRUE, Enable_Positive := TRUE,
Enable_Negative := TRUE, Override := 100);
35 McPower[1](Axis := AxisACS[1], Enable := TRUE, Enable_Positive := TRUE,
Enable_Negative := TRUE, Override := 100);
36 McPower[2](Axis := AxisACS[2], Enable := TRUE, Enable_Positive := TRUE,
Enable_Negative := TRUE, Override := 100);
37 McPower[3](Axis := AxisACS[3], Enable := TRUE, Enable_Positive := TRUE,
Enable_Negative := TRUE, Override := 100);
38 McPower[4](Axis := AxisACS[4], Enable := TRUE, Enable_Positive := TRUE,
Enable_Negative := TRUE, Override := 100);
39
40 //McJogging[5](Axis:=AxisA);
41 McJogging[1](Axis := AxisACS[1]);
42 McJogging[2](Axis := AxisACS[2]);
43 McJogging[3](Axis := AxisACS[3]);
44 McJogging[4](Axis := AxisACS[4]);
45
46
47 McReset[5](Axis := AxisA);
48 McReset[1](Axis := AxisACS[1]);
49 McReset[2](Axis := AxisACS[2]);
50 McReset[3](Axis := AxisACS[3]);
51 McReset[4](Axis := AxisACS[4]);
52
53
54 McStop[1](Axis := AxisACS[1], Deceleration := 1000, Jerk := 5000);
55 McStop[2](Axis := AxisACS[2], Deceleration := 1000, Jerk := 5000);
56 McStop[3](Axis := AxisACS[3], Deceleration := 1000, Jerk := 5000);
57 McStop[4](Axis := AxisACS[4], Deceleration := 1000, Jerk := 5000);
58 McStop[5](Axis := AxisA, Deceleration := 1000, Jerk := 5000);
59
60
61 McHome[1](Axis := AxisACS[1], bCalibrationCan := NOT fbAxes[1].
HomingSensor);
62 McHome[2](Axis := AxisACS[2], bCalibrationCan := NOT fbAxes[2].
HomingSensor);
63 McHome[3](Axis := AxisACS[3], bCalibrationCan := NOT fbAxes[3].
HomingSensor);
64 McHome[4](Axis := AxisACS[4], bCalibrationCan := NOT fbAxes[4].
HomingSensor);

```

```

65 //McHome[5] (Axis:=AxisA, bCalibrationCam:=NOT bRefSwitch);
66
67 CalculatedJ1 := FBCalculateXYZ . JointAngle_1 ;
68 CalculatedJ2 := FBCalculateXYZ . JointAngle_2 ;
69 CalculatedJ3 := FBCalculateXYZ . JointAngle_3 ;
70 CalculatedJ4 := FBCalculateXYZ . JointAngle_4 ;
71
72
73 // Safety procedure
74 (*IF UpperRefswitch = FALSE OR LowerRefswitch = FALSE THEN
75     iState := 800;
76     sStateDesc := '!ERROR Limit switch detected!';
77 END_IF*)
78
79 CASE iState OF
80
81     0 : // Initial
82         sStateDesc := '0: Waiting for Activation ' ;
83
84
85     5 : // AxisA: Start Drive reset
86         sStateDesc := '5: Axis calibration.' ;
87
88         McReset [ 5 ] ( Axis := AxisA , Execute := TRUE ) ;
89         McReset [ 1 ] ( Axis := AxisACS [ 1 ] , Execute := TRUE ) ;
90         McReset [ 2 ] ( Axis := AxisACS [ 2 ] , Execute := TRUE ) ;
91         McReset [ 3 ] ( Axis := AxisACS [ 3 ] , Execute := TRUE ) ;
92         McReset [ 13 ] ( Axis := AxisACS [ 4 ] , Execute := TRUE ) ;
93
94         IF (*McReset[5].Done AND *) McReset [ 1 ] . Done AND McReset [ 2 ] . Done
95         AND McReset [ 3 ] . Done AND McReset [ 13 ] . Done THEN
96             iState := 6 ;
97             ELSIF McReset [ 5 ] . Error AND McReset [ 1 ] . Error AND McReset [ 2 ] .
98             Error AND McReset [ 3 ] . Error AND McReset [ 13 ] . Error THEN
99                 iState := 999 ;
100             END_IF
101
102     6 : // AxisA wait for drive reset
103         sStateDesc := '6: AxisA - Wait for drive reset.' ;
104
105         (*McHome[5] (Axis:=AxisA,Execute:=FALSE);*)
106         McHome [ 1 ] ( Axis := AxisACS [ 1 ] , Execute := FALSE ) ;
107         McHome [ 2 ] ( Axis := AxisACS [ 2 ] , Execute := FALSE ) ;
108         McHome [ 3 ] ( Axis := AxisACS [ 3 ] , Execute := FALSE ) ;
109         McHome [ 10 ] ( Axis := AxisACS [ 4 ] , Execute := FALSE ) ;
110
111         IF (*McPower[5].Status AND*) McPower [ 1 ] . Status AND McPower [ 2 ] .
112         Status AND McPower [ 3 ] . Status AND McPower [ 4 ] . Status AND btHoming THEN
113             iState := 7 ;
114         END_IF

```



```

113       7 : //AxisA start homing process
114         sStateDesc := '7: AxisA - Start homing process.' ;
115
116         McHome [ 5 ] ( Axis := AxisA , bCalibrationCam := NOT UpperRefswitch , Execute
:= TRUE ) ;
117         McHome [ 2 ] ( Axis := AxisACS [ 2 ] , bCalibrationCam := NOT fbAxes [ 2 ] .
HomingSensor , Execute := TRUE ) ;
118         McHome [ 3 ] ( Axis := AxisACS [ 3 ] , bCalibrationCam := NOT fbAxes [ 3 ] .
HomingSensor , Execute := TRUE ) ;
119         McHome [ 4 ] ( Axis := AxisACS [ 4 ] , bCalibrationCam := NOT fbAxes [ 4 ] .
HomingSensor , Execute := TRUE ) ;
120         McHome [ 1 ] ( Axis := AxisACS [ 1 ] , bCalibrationCam := NOT fbAxes [ 1 ] .
HomingSensor , Execute := TRUE ) ;
121         IF McHome [ 5 ] . Busy AND McHome [ 1 ] . Busy AND McHome [ 2 ] . Busy AND
McHome [ 3 ] . Busy AND McHome [ 4 ] . Busy THEN
122           iState := 8 ;
123         END_IF
124
125       8 : // AxisA homing running
126         sStateDesc := '8: AxisA - Homing running.' ;
127
128         IF McHome [ 5 ] . Done AND McHome [ 1 ] . Done AND McHome [ 2 ] . Done AND
McHome [ 3 ] . Done AND McHome [ 4 ] . Done THEN
129           McHome [ 5 ] ( Axis := AxisA , Execute := FALSE ) ;
130           McHome [ 1 ] ( Axis := AxisACS [ 1 ] , Execute := FALSE ) ;
131           McHome [ 2 ] ( Axis := AxisACS [ 2 ] , Execute := FALSE ) ;
132           McHome [ 3 ] ( Axis := AxisACS [ 3 ] , Execute := FALSE ) ;
133           McHome [ 4 ] ( Axis := AxisACS [ 4 ] , Execute := FALSE ) ;
134           iState := 10 ;
135         ELSEIF McHome [ 5 ] . Error OR McHome [ 1 ] . Error OR McHome [ 2 ] . Error
OR McHome [ 3 ] . Error OR McHome [ 4 ] . Error THEN
136           McHome [ 5 ] ( Axis := AxisA , Execute := FALSE ) ;
137           McHome [ 1 ] ( Axis := AxisACS [ 1 ] , Execute := FALSE ) ;
138           McHome [ 2 ] ( Axis := AxisACS [ 2 ] , Execute := FALSE ) ;
139           McHome [ 3 ] ( Axis := AxisACS [ 3 ] , Execute := FALSE ) ;
140           McHome [ 4 ] ( Axis := AxisACS [ 4 ] , Execute := FALSE ) ;
141           iState := 10 ; // Dit was 905 error code maar as I homed niet
142         END_IF
143
144
145       10 :
146         sStateDesc := '10: Robot ready to move!' ;
147
148       11 :
149         sStateDesc := '11: Jog mode' ; // select with
button JOGMODE HMI
150
151         CASE MotorJog OF // Select the motors
to JOG on HMI
152           1 : sJogDesk := 'Joint 1 CCW' ; // JOG Forward
153           McJogging [ 1 ] ( Axis := AxisACS [ 1 ] , JogForward := TRUE ) ;

```

```

154         2 : sJogDesk := 'Joint 2 CCW' ;
155         McJogging [ 2 ] ( Axis := AxisACS [ 2 ] , JogForward := TRUE ) ;
156         3 : sJogDesk := 'Joint 3 CCW' ;
157         McJogging [ 3 ] ( Axis := AxisACS [ 3 ] , JogForward := TRUE ) ;
158         4 : sJogDesk := 'Joint 4 CCW' ;
159         McJogging [ 4 ] ( Axis := AxisACS [ 4 ] , JogForward := TRUE ) ;
160         5 : sJogDesk := 'Joint 1 CW' ;
161         McJogging [ 1 ] ( Axis := AxisACS [ 1 ] , JogBackwards := TRUE ) ;
162         6 : sJogDesk := 'Joint 2 CW' ;
163         McJogging [ 2 ] ( Axis := AxisACS [ 2 ] , JogBackwards := TRUE ) ;
164         7 : sJogDesk := 'Joint 3 CW' ;
165         McJogging [ 3 ] ( Axis := AxisACS [ 3 ] , JogBackwards := TRUE ) ;
166         8 : sJogDesk := 'Joint 4 CW' ;
167         McJogging [ 4 ] ( Axis := AxisACS [ 4 ] , JogBackwards := TRUE ) ;
168         100 : sJogDesk := 'Select joint' ;
169         McJogging [ 1 ] ( Axis := AxisACS [ 1 ] , JogForward := FALSE ) ;
170         McJogging [ 2 ] ( Axis := AxisACS [ 2 ] , JogForward := FALSE ) ;
171         McJogging [ 3 ] ( Axis := AxisACS [ 3 ] , JogForward := FALSE ) ;
172         McJogging [ 4 ] ( Axis := AxisACS [ 4 ] , JogForward := FALSE ) ;
173         McJogging [ 1 ] ( Axis := AxisACS [ 1 ] , JogBackwards := FALSE ) ;
174         McJogging [ 2 ] ( Axis := AxisACS [ 2 ] , JogBackwards := FALSE ) ;
175         McJogging [ 3 ] ( Axis := AxisACS [ 3 ] , JogBackwards := FALSE ) ;
176         McJogging [ 4 ] ( Axis := AxisACS [ 4 ] , JogBackwards := FALSE ) ;
177     END_CASE
178
179
180     12 :         sStateDesc := '12: Calculate pos!' ;
181         fbCalculateXYZ ( X_coordinate := X_coordinate , Y_coordinate :=
Y_coordinate , Z_coordinate := Z_coordinate , Graden_orientatie :=
Graden_orientatie ) ;
182         iState := 13 ;
183     13 : sStateDesc := '13: Moving!' ;
184         McMoveAbs [ 4 ] ( Axis := AxisACS [ 1 ] ,
185             Position := FBCalculateXYZ . JointAngle_1 ,
186             Velocity := 20 ,
187             Acceleration := 17 ,
188             Deceleration := 17 ,
189             Jerk := 13 ,
190             Execute := TRUE ) ;
191         McMoveAbs [ 5 ] ( Axis := AxisACS [ 2 ] ,
192             Position := FBCalculateXYZ . JointAngle_2 ,
193             Velocity := 25 ,
194             Acceleration := 17 ,
195             Deceleration := 17 ,
196             Jerk := 13 ,
197             Execute := TRUE ) ;
198         McMoveAbs [ 6 ] ( Axis := AxisACS [ 3 ] ,
199             Position := FBCalculateXYZ . JointAngle_3 ,
200             Velocity := 25 ,
201             Acceleration := 17 ,
202             Deceleration := 17 ,

```

```

203         Jerk := 13 ,
204         Execute := TRUE ;
205     McMoveAbs [ 7 ] ( Axis := AxisACS [ 4 ] ,
206         Position := FBCalculateXYZ . JointAngle_4 ,
207         Velocity := 25 ,
208         Acceleration := 17 ,
209         Deceleration := 17 ,
210         Jerk := 13 ,
211         Execute := TRUE ) ;
212     IF McMoveAbs [ 4 ] . Done AND McMoveAbs [ 5 ] . Done AND McMoveAbs [ 6 ] .
213     Done AND McMoveAbs [ 7 ] . Done AND NOT
214     McMoveAbs [ 4 ] . Error OR McMoveAbs [ 5 ] . Error OR McMoveAbs [ 6 ] .
215     Error OR McMoveAbs [ 7 ] . Error THEN
216         McMoveAbs [ 4 ] ( Axis := AxisACS [ 1 ] , Execute := FALSE ) ;
217         McMoveAbs [ 5 ] ( Axis := AxisACS [ 2 ] , Execute := FALSE ) ;
218         McMoveAbs [ 6 ] ( Axis := AxisACS [ 3 ] , Execute := FALSE ) ;
219         McMoveAbs [ 7 ] ( Axis := AxisACS [ 4 ] , Execute := FALSE ) ;
220         iState := 10 ;
221     END_IF
222 14 : sStateDesc := 'Move to home position' ;
223     McMoveAbs [ 4 ] ( Axis := AxisACS [ 1 ] ,
224         Position := 0 ,
225         Velocity := 20 ,
226         Acceleration := 80 ,
227         Deceleration := 80 ,
228         Jerk := 500 ,
229         Execute := TRUE ) ;
230     McMoveAbs [ 5 ] ( Axis := AxisACS [ 2 ] ,
231         Position := 0 ,
232         Velocity := 20 ,
233         Acceleration := 80 ,
234         Deceleration := 80 ,
235         Jerk := 500 ,
236         Execute := TRUE ) ;
237     McMoveAbs [ 6 ] ( Axis := AxisACS [ 3 ] ,
238         Position := 0 ,
239         Velocity := 20 ,
240         Acceleration := 80 ,
241         Deceleration := 80 ,
242         Jerk := 500 ,
243         Execute := TRUE ) ;
244     McMoveAbs [ 7 ] ( Axis := AxisACS [ 4 ] ,
245         Position := 0 ,
246         Velocity := 20 ,
247         Acceleration := 80 ,
248         Deceleration := 80 ,
249         Jerk := 500 ,
250         Execute := TRUE ) ;
251     IF McMoveAbs [ 4 ] . Done AND McMoveAbs [ 5 ] . Done AND McMoveAbs [ 6 ] .
252     Done AND McMoveAbs [ 7 ] . Done AND NOT
253     McMoveAbs [ 4 ] . Error OR McMoveAbs [ 5 ] . Error OR McMoveAbs [ 6 ] .

```

```

Error OR McMoveAbs [ 7 ] . Error THEN
251     McMoveAbs [ 4 ] ( Axis := AxisACS [ 1 ] , Execute := FALSE ) ;
252     McMoveAbs [ 5 ] ( Axis := AxisACS [ 2 ] , Execute := FALSE ) ;
253     McMoveAbs [ 6 ] ( Axis := AxisACS [ 3 ] , Execute := FALSE ) ;
254     McMoveAbs [ 7 ] ( Axis := AxisACS [ 4 ] , Execute := FALSE ) ;
255     iState := 10 ;
256     END_IF
257
258     800 :   sStateDesc := 'an error has occurred, please restart the system' ;
259           McStop [ 1 ] . Execute := TRUE ;
260           McStop [ 2 ] . Execute := TRUE ;
261           McStop [ 3 ] . Execute := TRUE ;
262           McStop [ 4 ] . Execute := TRUE ;
263           McStop [ 5 ] . Execute := TRUE ;
264
           //fbEStopNCI(sNciToPlc:=in_stNciToPlc,bExecute:=TRUE,tDec:=800,tJerk:=20000);
265
266     995 : // Stop motor
267           sStateDesc := '900, Stop Motor.' ;
268
269           McStop [ 1 ] ( Axis := AxisA , Execute := TRUE ) ;
270           IF btActivator THEN
271               McStop [ 1 ] ( Axis := AxisA , Execute := FALSE ) ;
272               iState := 40 ;
273           END_IF
274
275     901 : // Error 1
276
277           sStateDesc := 'Error 901' ;
278
279           McHome [ 1 ] ( Axis := AxisACS [ 1 ] , Execute := FALSE ) ;
280           McReset [ 1 ] ( Axis := AxisACS [ 1 ] , Execute := TRUE ) ;
281           IF btReturn THEN
282               iState := 10 ;
283           END_IF
284
285     902 : // Error 2
286
287           sStateDesc := 'Error 902' ;
288
289           McHome [ 2 ] ( Axis := AxisACS [ 2 ] , Execute := FALSE ) ;
290           McReset [ 2 ] ( Axis := AxisACS [ 2 ] , Execute := TRUE ) ;
291           IF btReturn THEN
292               iState := 20 ;
293           END_IF
294
295     903 : // Error 3
296
297           sStateDesc := 'Error 903' ;
298
299           McHome [ 3 ] ( Axis := AxisACS [ 3 ] , Execute := FALSE ) ;
300           McReset [ 3 ] ( Axis := AxisACS [ 3 ] , Execute := TRUE ) ;

```

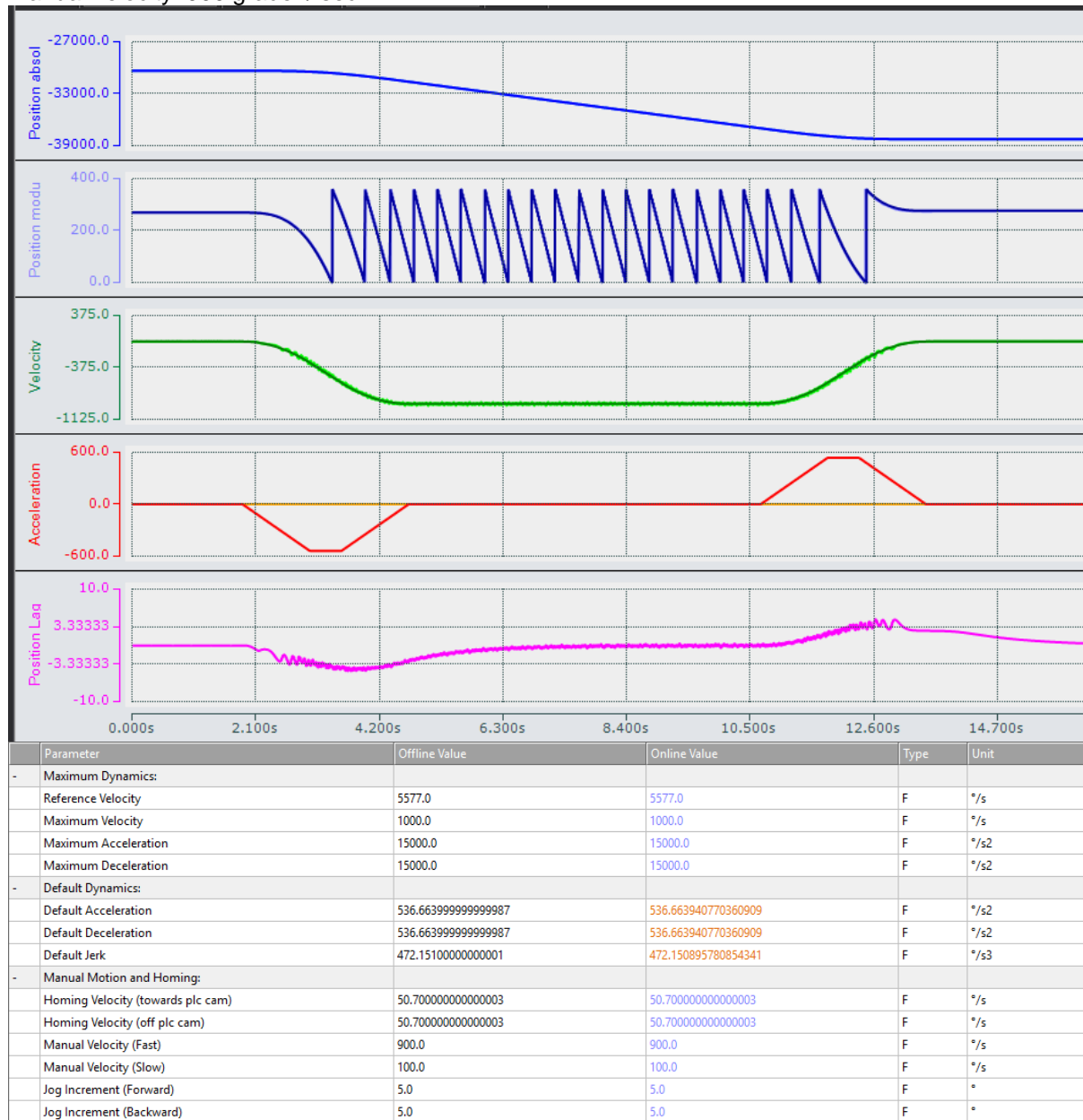
```
301     IF btReturn THEN
302         iState := 30 ;
303     END_IF
304
305     904 : // Error 4
306
307     sStateDesc := 'Error 904' ;
308
309     McHome [ 4 ] ( Axis := AxisACS [ 5 ], Execute := FALSE ) ;
310     McReset [ 13 ] ( Axis := AxisACS [ 5 ], Execute := TRUE ) ;
311     IF btReturn THEN
312         iState := 40 ;
313     END_IF
314
315     905 : // Error 5
316
317     sStateDesc := 'Error 905' ;
318
319     McHome [ 5 ] ( Axis := AxisA, Execute := FALSE ) ;
320     McReset [ 5 ] ( Axis := AxisA, Execute := TRUE ) ;
321     IF btReturn THEN
322         iState := 5 ;
323     END_IF
324
325     920 : // error kinematics
326     sStateDesc := 'Error 920: kinematics failure' ;
327     IF btReturn THEN
328         iState := 100 ;
329         btReturn := FALSE ;
330     END_IF
331
332     930 : // nc group error
333     sStateDesc := 'Error 930: nc group error' ;
334     IF btReturn THEN
335         iState := 100 ;
336         btReturn := FALSE ;
337     END_IF
338
339     960 : // load file error
340     sStateDesc := 'Error 960: file load error' ;
341     IF btReturn THEN
342         iState := 260 ;
343         btReturn := FALSE ;
344     END_IF
345
346     999 : //ERROR HANDLING TO DO!!
347     sStateDesc := '999, Error.' ;
348     IF btReturn THEN
349         iState := 0 ;
350         btReturn := FALSE ;
351
```

```
352         END_IF
353     ELSE
354         sStateDesc := 'Invalid State.' ;
355     END_CASE
356 END_CASE
357
```

## Bijlage E Plots motion

In deze bijlage zijn plots te zien. Met deze plots is er gezocht naar de perfecte beweging en naar een goede position lag voor het systeem. In deze plots zijn 4 verschillende waardes weergegeven. Dit zijn respectievelijk: De absolute positie, de snelheid, acceleratie en de position lag. Daarna volgen diverse parameters die bij deze plots horen.

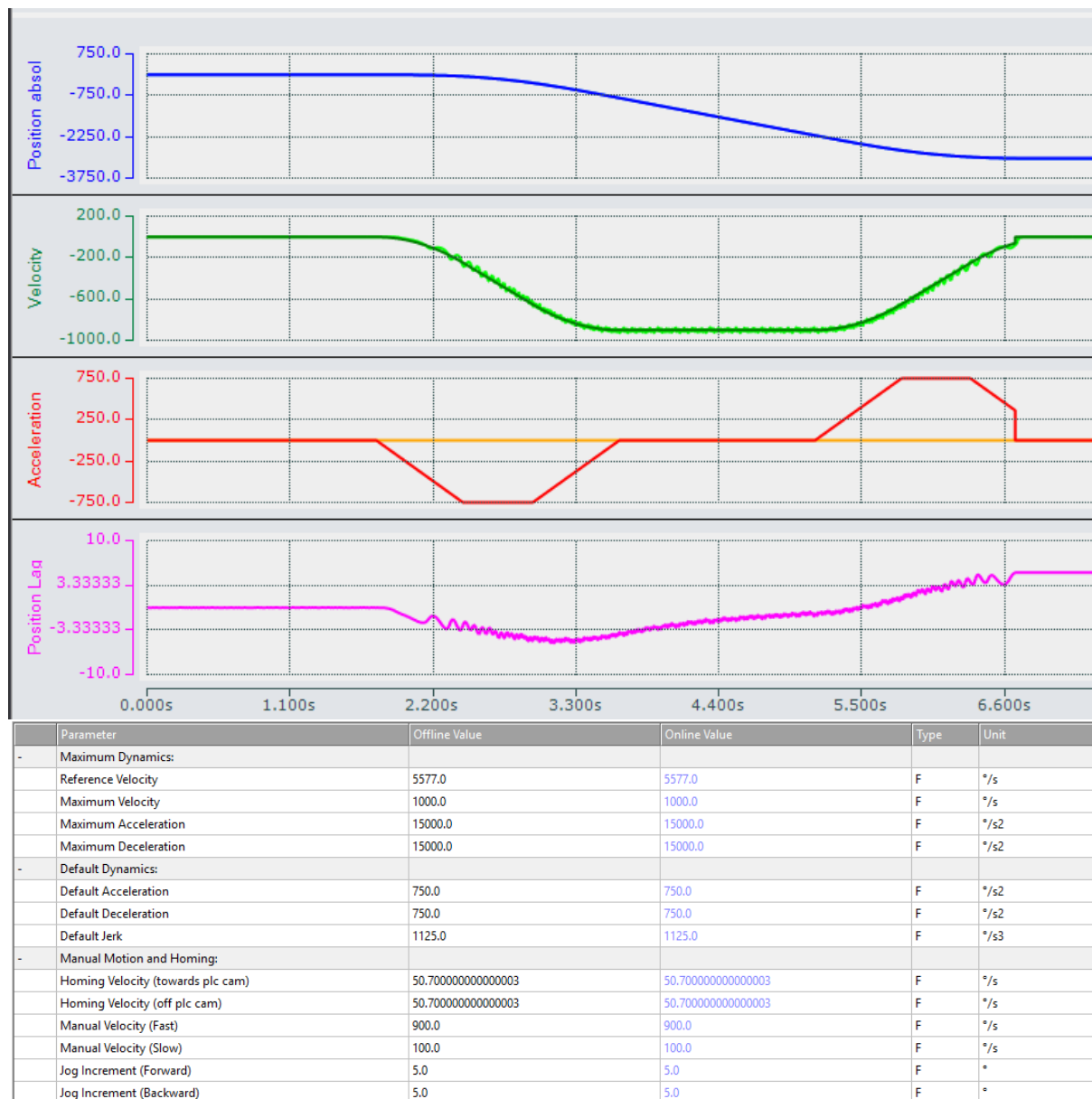
Manual velocity: 900 graden/ sec



acceleratie tijd: 3 Seconden

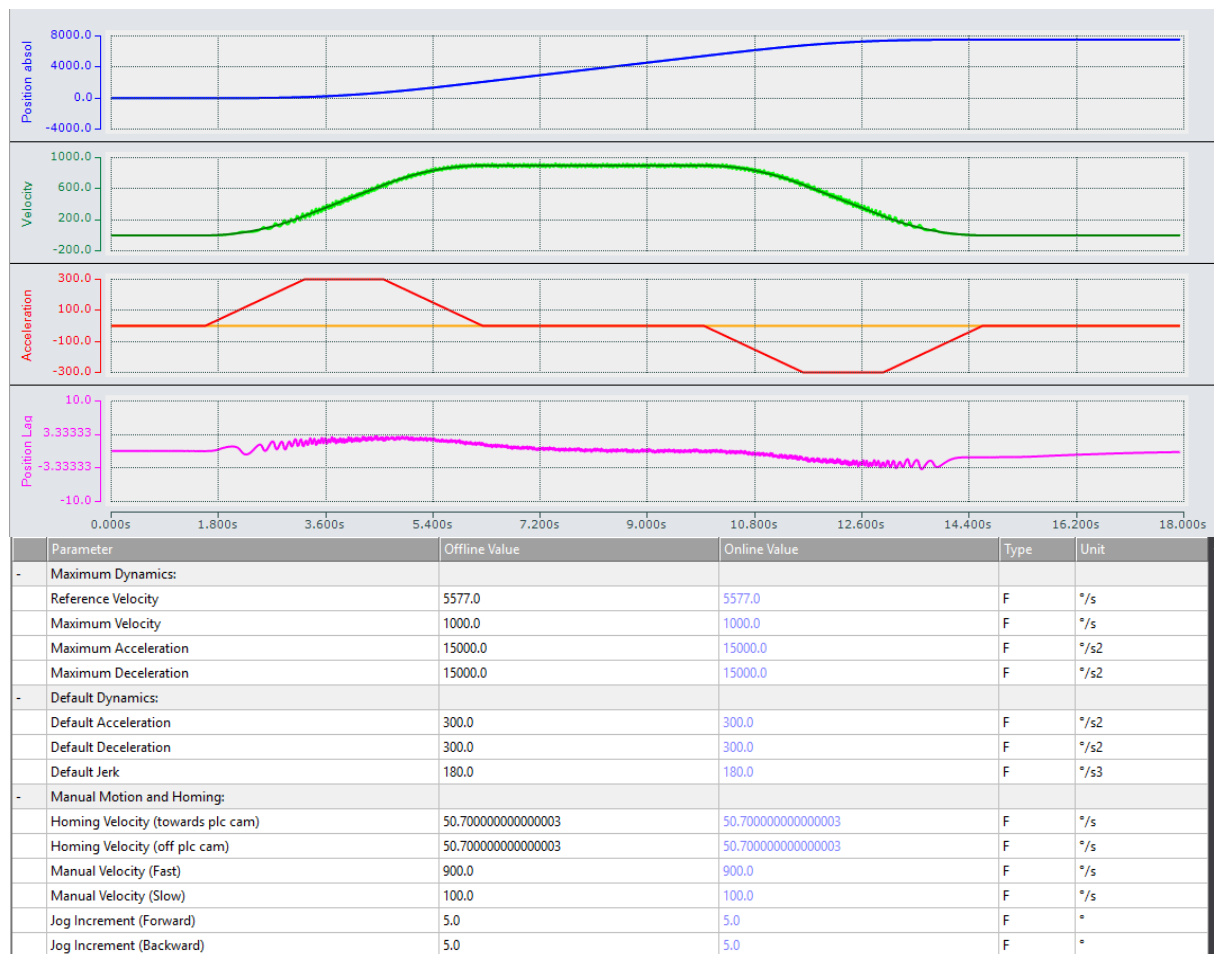
max speed: 1000 graden per seconden





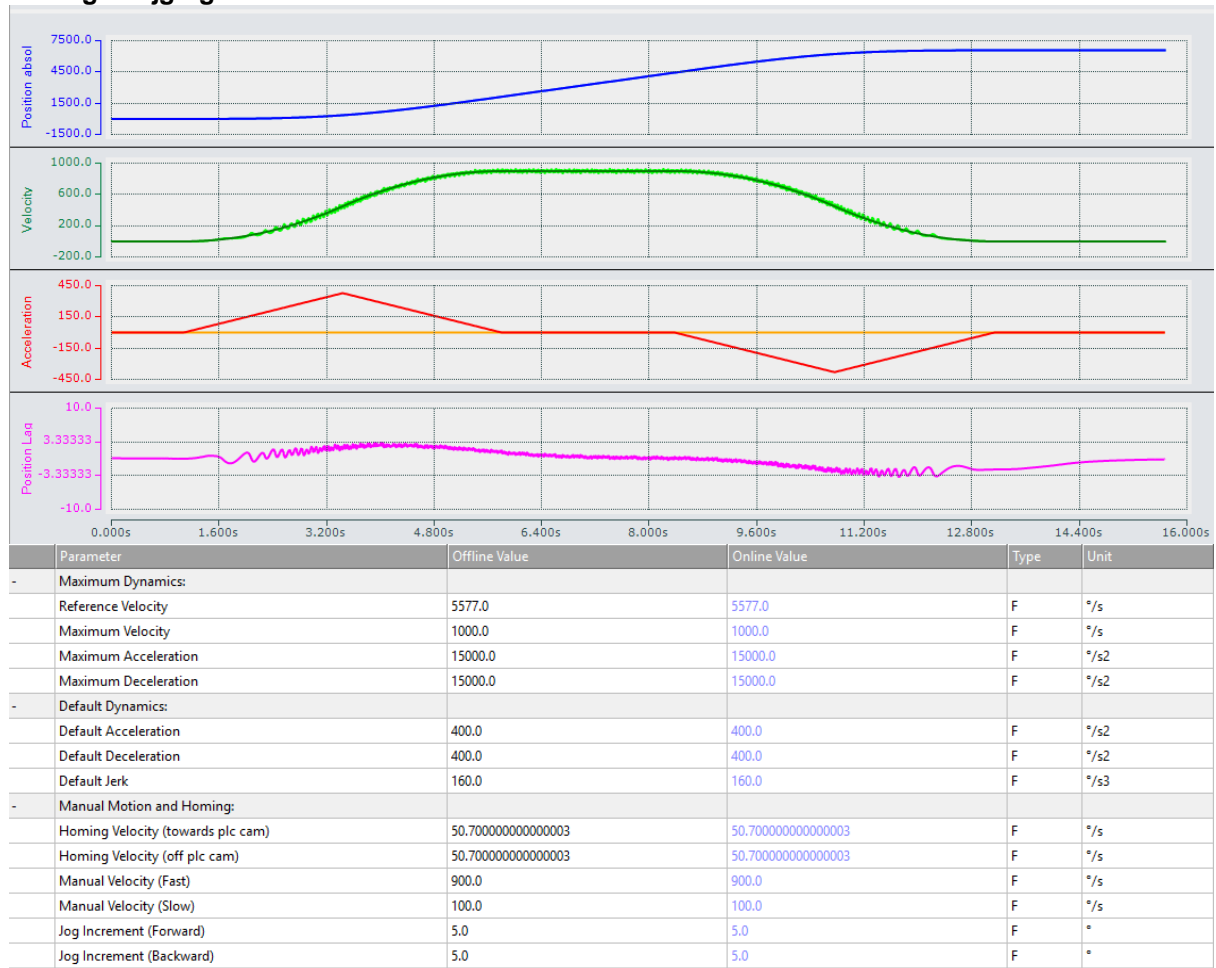
acceleratietijd: 2 Seconden

max speed: 1000 graden per seconden



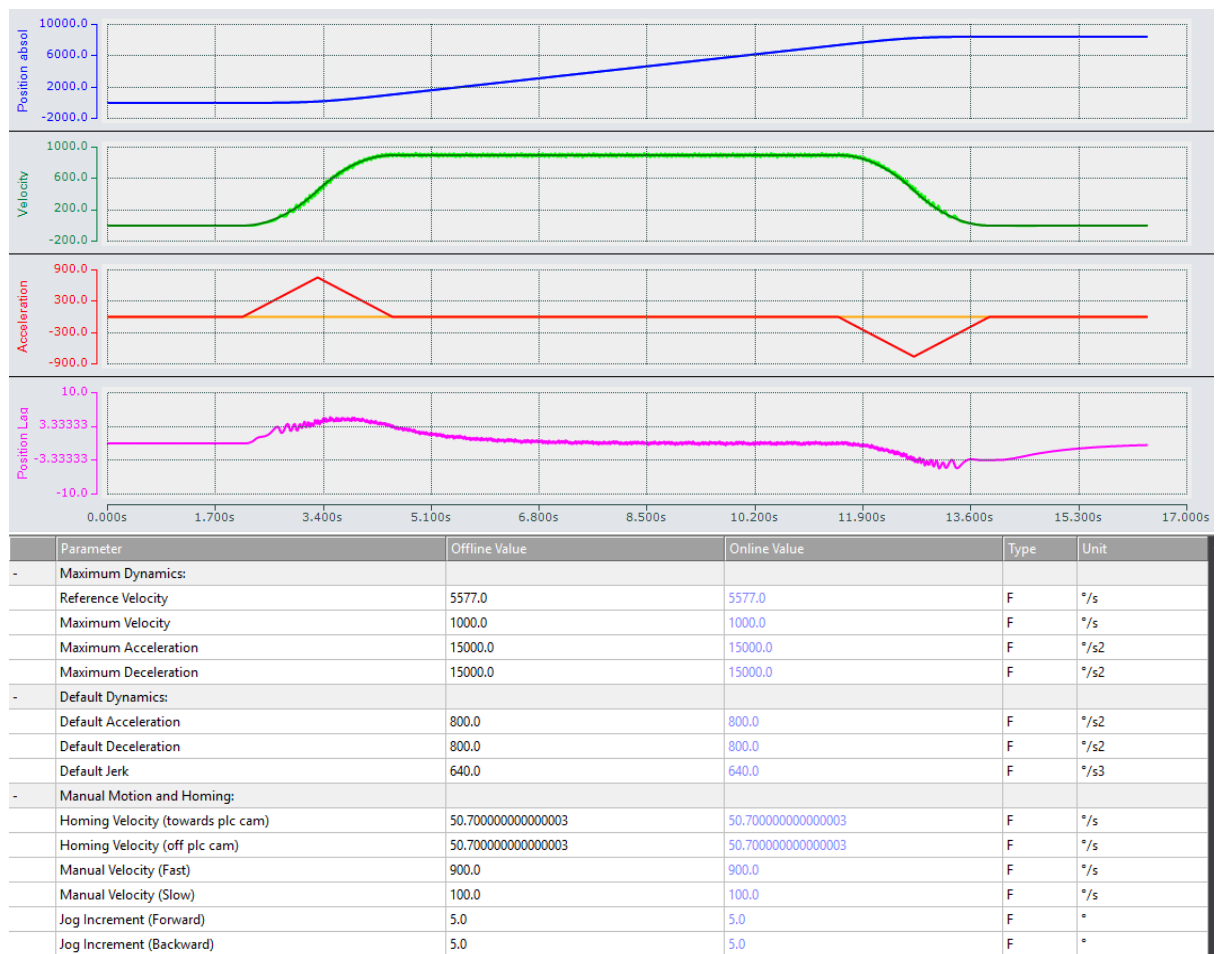
acceleratietijd: 5 Seconden  
 max speed: 1000 graden per seconden

## Hoekige stijging



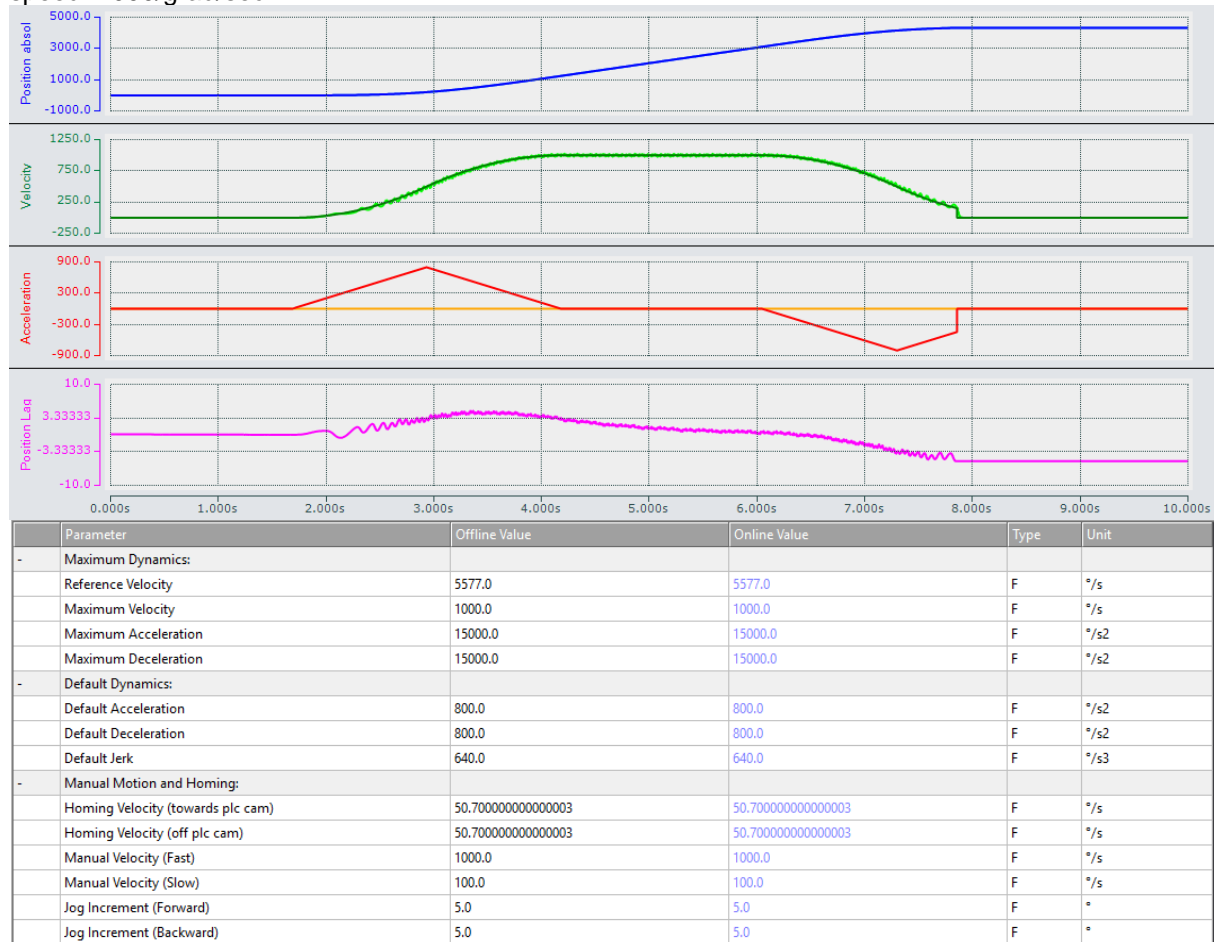
acceleratietijd: 5 Seconden

max speed: 1000 graden per seconden



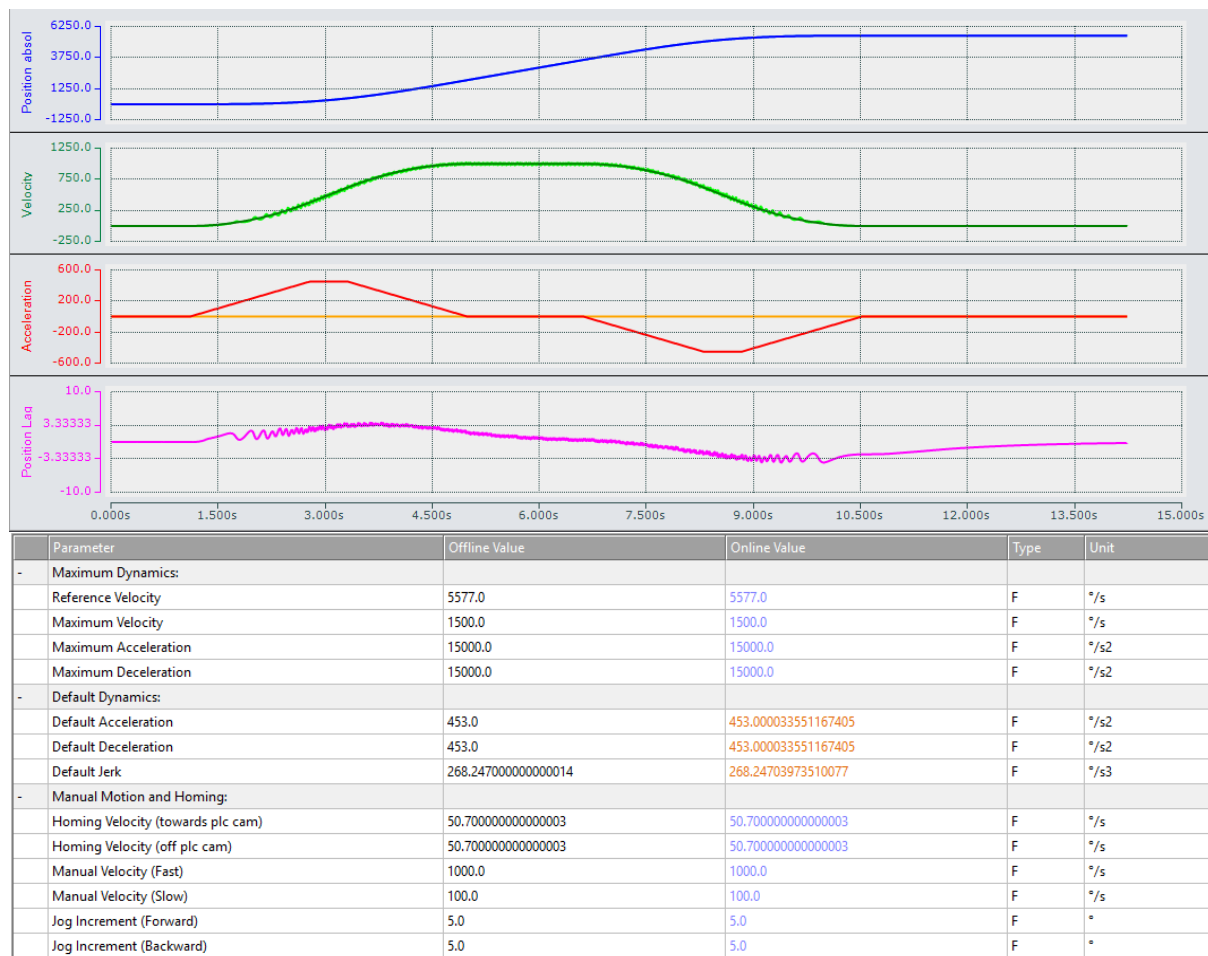
acceleratietijd: 2.5 Seconden  
 max speed: 1000 graden per seconden

speed: 1000/grad/sec



acceleratietijd: 2.5 Seconden

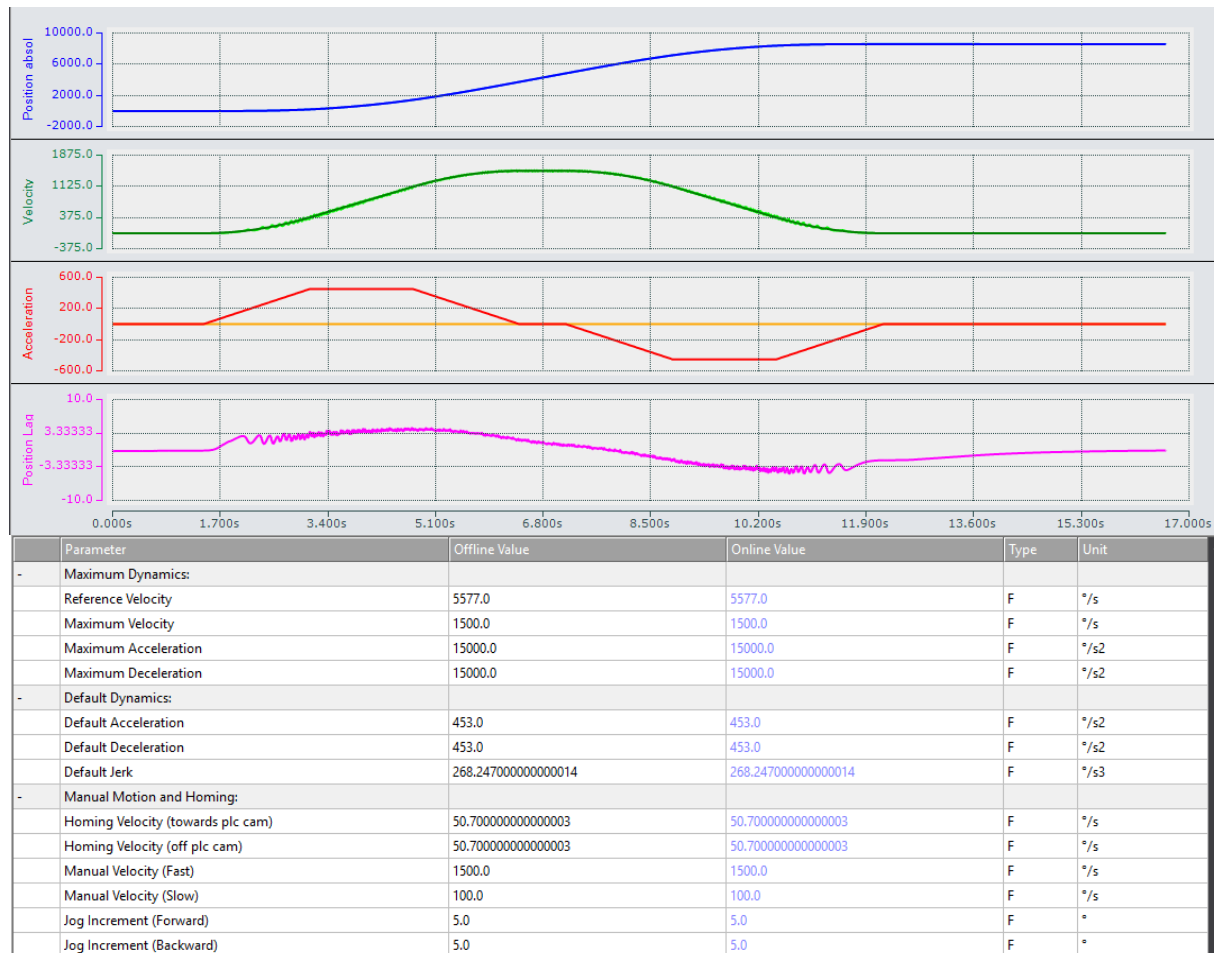
max speed: 1000 graden per seconden



acceleratie tijd: 5 Seconden

max speed: 1500 graden per seconden

speed: 1500grad/sec



acceleratietijd: 5 Seconden

max speed: 1500 graden per seconden