

Muës Joey
Wilco Waaijer
3EAIA

Documentatie Project: Deltarobot



Inhoudsopgave

[1. Beschrijving](#)

[2. Software](#)

[2.1 LabVIEW](#)

[2.1.1 Modules](#)

[2.2 Solid Works](#)

[2.3 LabVIEW code](#)

[2.3.1 Main VI](#)

[2.3.2 SubVI](#)

[2.3.3 Front Panels](#)

[3. Hardware](#)

[4. Hardware & Software combined](#)

[5. Projectverloop](#)

[6. Toekomst](#)

[7. Resources](#)

1. Beschrijving

De bedoeling van het project is om op een Delta Robot die ons ter beschikking is gesteld een demo te laten zien. Dit zal gebeuren met een sturing die geprogrammeerd is in LabVIEW. Doordat de Delta Robot een duur apparaat is, hebben we eerst onze code getest met een tekening die we gekregen hadden, pas als deze werkte zoals verwacht zal het onder toezicht op de robot zelf aangestuurd worden.

2. Software

2.1 LabVIEW

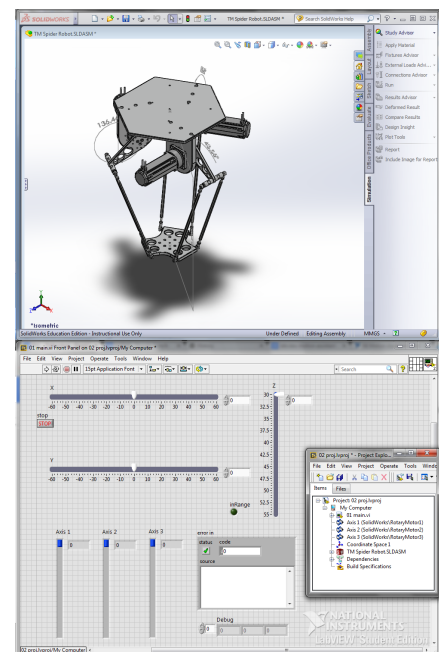
Om de robot te kunnen programmeren gebruiken we Labview, de eerste stap was om enkele extra modules te installeren, die hier onder vermeld zijn en uitgelegd zullen worden.

2.1.1 Modules

Om de SolidWorks tekening te laten werken met de LabVIEW code zijn er enkele modules gedownload die ons ter beschikking gesteld werden. Zo is er gebruik gemaakt van de Real Time module om real time applications aan te sturen en de SoftMotion module voor de motion control die we zullen gebruiken. Als laatste is er nog de NI-RIO module voor het besturen van de actuele motoren en de Motion Assistant om de LabVIEW code te koppelen met een SolidWorks tekening.

Het gebruiken van de SolidWorks tekening in de LabVIEW VI was zeer simpel aangezien er een stap per stap guide online stond die we gebruikt hebben. De guide zelf is de NI-Tutorial-9566 en is makkelijk terug te vinden op de website van National Instruments.

De guide bestaat uit 15 makkelijk te volgen stappen en binnen het half uurtje weet je de basis over het besturen van een tekening in je VI.



2.2 Solid Works

Solid Works is een CAD programma waarin we een LabVIEW VI verbinden met de tekening. De tekening van de Delta Robot werd ons ter beschikking gesteld. Aan de tekening zelf is er niets mogen veranderen, wel zijn hier maten genomen die nodig waren om de (inverse) kinematica te kunnen berekenen die in de code zal gebruikt worden.

2.3 LabVIEW code

De LabVIEW code die geschreven werd bestaat uit een VI en één enkele SubVI. In de SubVI worden alle berekeningen gedaan die voor de beweging van het platform zullen zorgen terwijl in de main VI de motoren zelf aangestuurd zullen worden.

2.3.1 Main VI

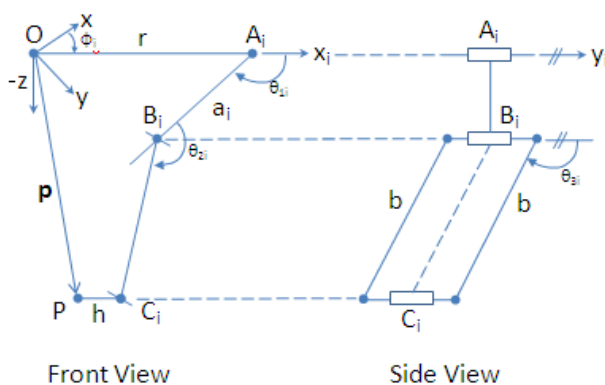
de Main VI kan opgedeeld worden in 3 grote secties, namelijk de input sectie, de berekening en het bewegen van de motor.

Allereerst bespreken we de input van de main VI, deze zijn constanten en zullen maar één maal ingegeven worden.

Om alles te kunnen berekenen zijn er 4 lengtes, 3 hoeken en 2 limieten nodig. De lengtes die we nodig hebben zijn "a", deze is de lengte van de motor naar het scharnierpunt van de tweede arm, en "b", die van het scharnierpunt naar de basis van het platform gaat.

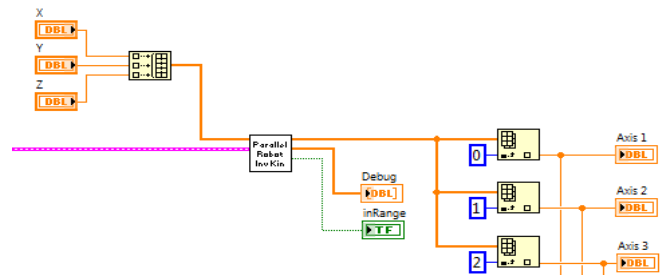
Daarbuiten hebben de maten van het platform en de basis nodig. Het platform word gemeten van het scharnierpunt met lengte b tot het midden van het platform, de basis word gemeten van de motor tot het midden. Aangezien deze maten niet zullen veranderen zijn ze niet variabel gemaakt.

De waarden Theta1, 2 en 3 zijn de hoeken van de motoren ten opzichte van het nulpunt op de x-as, deze zijn ook vaste waarden en hebben deze niet variabel gemaakt. De upper en lower limit zijn ingesteld zodat de Delta Robot geen bewegingen kan maken die niet gaan, zo word er gezorgd dat de motoren niet beschadigd raken.



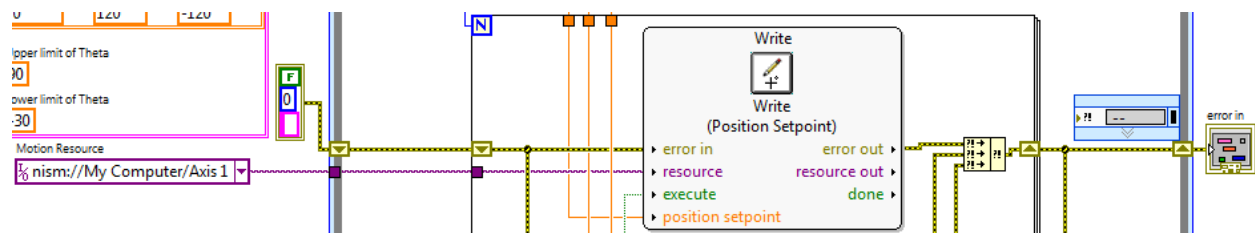
Length of arms "a" (cm)	13
Length of arms "b" (cm)	40
Length of base "r" (cm)	10
Length of platform "h" (cm)	8
Phi (deg)	
Theta 1 (deg) 2 Theta 1 (deg) 2 Theta 1 (deg) 2	0 120 -120
Upper limit of Theta	90
Lower limit of Theta	-30

Deze waarden worden doorgegeven naar het volgende deel. Dit gedeelte bekijkt de x-, y- en z-waarden die ingegeven worden in het front panel en zet ze in een array. De waarden van de array worden gestuurd naar de SubVI, samen met de input die eerder vermeld werd.



Eenmaal als de SubVI alles heeft berekend zal hij de waarden op het Front Panel weergeven en doorsturen naar servosturing.

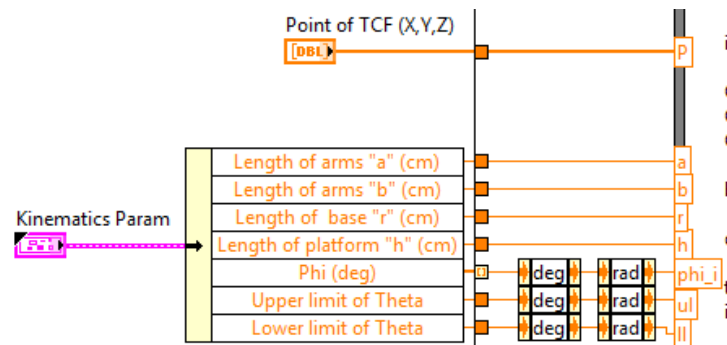
De motoren worden door middel van de motion resource gekoppeld aan een motor in het project, die op zijn beurt aan de motor van de Solid Works tekening gekoppeld is. Elke motor krijgt een waarde doorgestuurd door de SubVI en zal bewegen naar desbetreffende coördinaten.



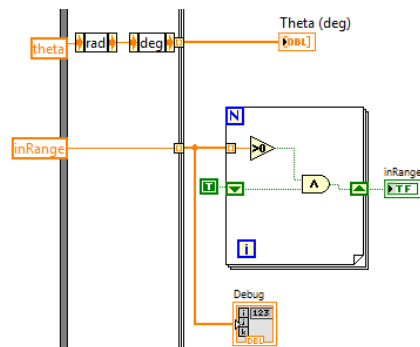
2.3.2 SubVI

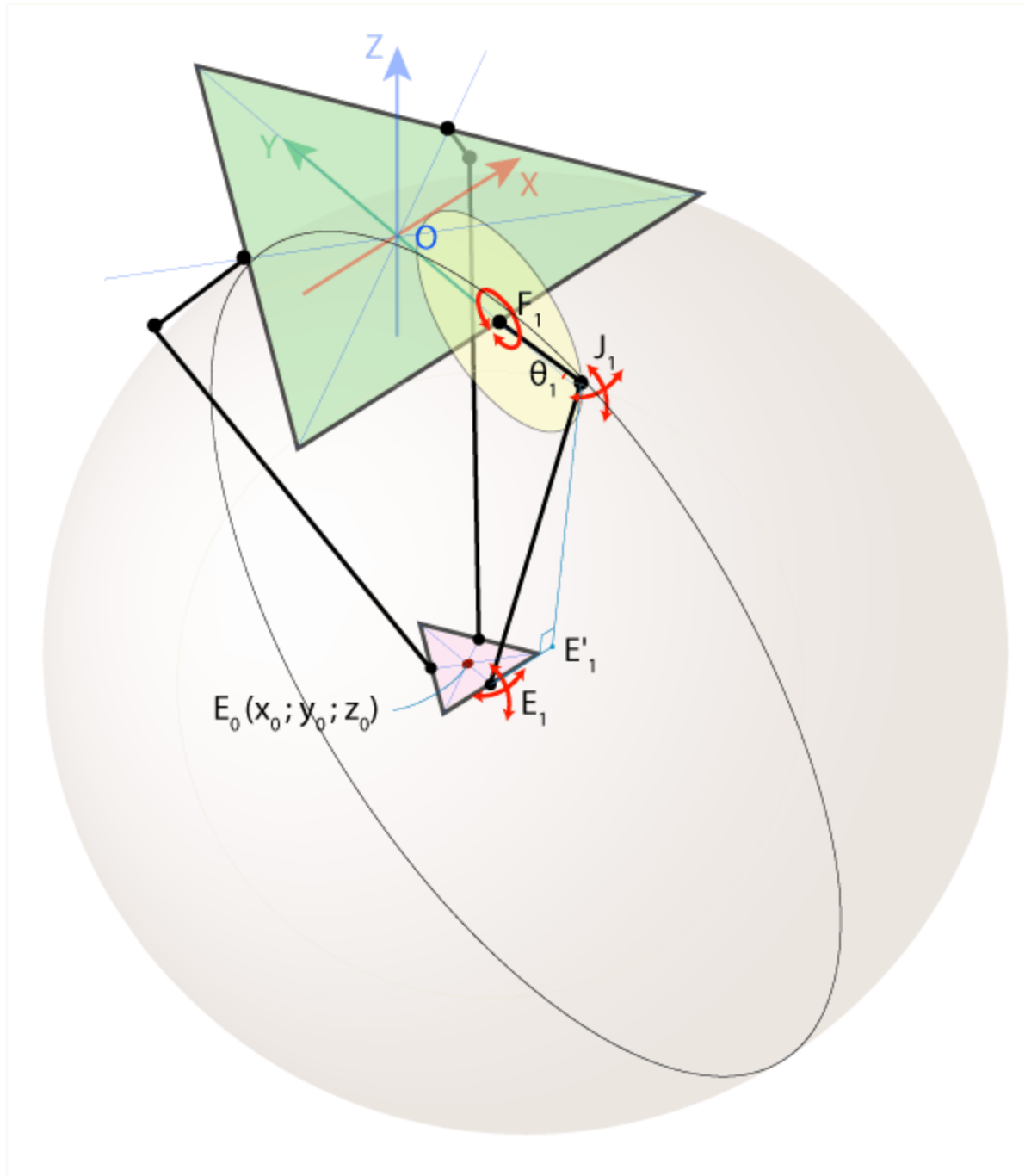
De SubVI bestaat zelf uit 3 delen, waarvan de input, die word ingelezen, de actuele code voor het bereken van de inverse kinematica en een klein stukje waarin onder andere de array die de waarden naar de motoren stuurt staat.

De inputs komen rechtstreeks van de main VI en worden zo gedeclareerd in de code. De hoeken worden ingelezen in graden en daarna geconverteerd naar radialen, dit omdat de code met radialen werkt.



Het laatste deel is de uitkomst Theta, een array die de waarden bevat om de motoren aan te sturen, samen met een klein debug command en een indicator die aangeeft of de motor nog tussen de limieten bevindt.





Het grootste en meest complexe gedeelte is de code achter de kinematica. Er waren veel online voorbeelden te vinden, maar geen enkele die werkte zoals het zou moeten, daarom is er helemaal vanaf nul begonnen. Dit zorgde er voor dat we de berekeningen volledig begrepen. De berekeningen zijn vrij ingewikkeld omdat je de snijpunten van een sphere met een vlakke cirkel moet berekenen. Dit is uiteindelijk gelukt en dit is de uitkomst:

```
//Define variables
float64 Cx; float64 Cy; float64 Cz; float64 theta1; float64 theta2; float64 b2; float64 d;
```

```
inRange = 1;
```

```
Cx = P[0]*cos(phi_i) + P[1]*sin(phi_i) + h - r;
Cy = P[0]*-sin(phi_i) + P[1]*cos(phi_i);
Cz = P[2];
```

```
b2 = sqrt( b**2 - Cy**2 );
```

```
d = sqrt( Cz**2 + Cx**2 );
```

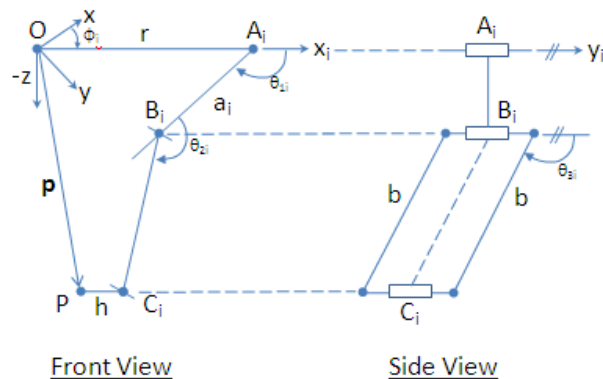
```
theta1 = atan( Cz/ Cx );
if ( Cx < 0 ) theta1 += pi;
```

```
theta2 = acos( ( ( d**2 - b2**2 + a**2 ) / ( 2 * d ) ) / a );
```

```
theta = theta1 - theta2;
```

```
if ( abs(Cy) > b ) {
    theta = ul;
    inRange = -3;
}
else if ( d > ( a + b2 ) ) {
    theta = ul;
    inRange = -1;
}
else if ( d < a ) {
    theta = ll;
    inRange = -2;
}
}
```

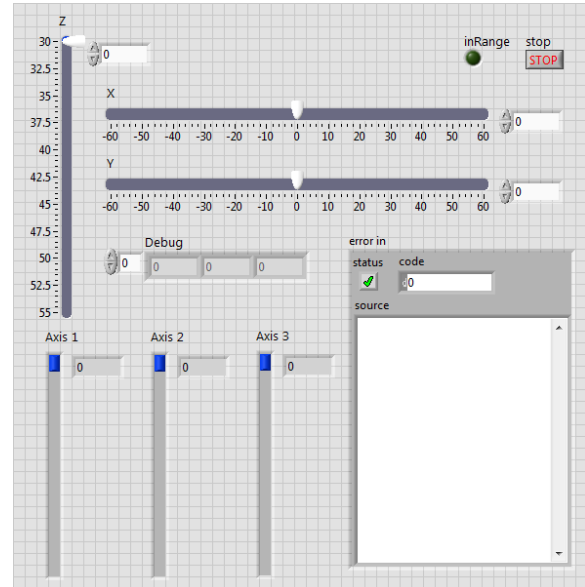
```
if ( theta > ul ) {
    theta = ul;
    inRange = -4;
}
else if ( theta < ll ) {
    theta = ll;
    inRange = -5;
}
}
```



2.3.3 Front Panels

De Front Panel is zo gebruiksvriendelijk mogelijk gemaakt, zo zijn er enkel volgende onderdelen op te zien:

- een slider voor de x-, y- en z-as voor de plaatsbepaling van het platform;
- een inRange indicator om weer te geven of de robot nog binnen zijn limieten zit;
- een debug array dat vooral gebruikt werd om fouten op te sporen;
- Axis 1, 2 en 3 zijn de waarden van de motoren om de juiste plaatsbepaling te verkrijgen;
- error in geeft weer als er foutmeldingen zouden optreden;
- de Stop knop is om het project uit te schakelen;



Op deze manier zijn enkel de belangrijkste gegevens nodig en heeft de gebruiker een visuele representatie van de waarden die naar de robot worden gestuurd.

3. Hardware

Voor de sturing van de Delta Robot zijn volgende hardware componenten nodig:

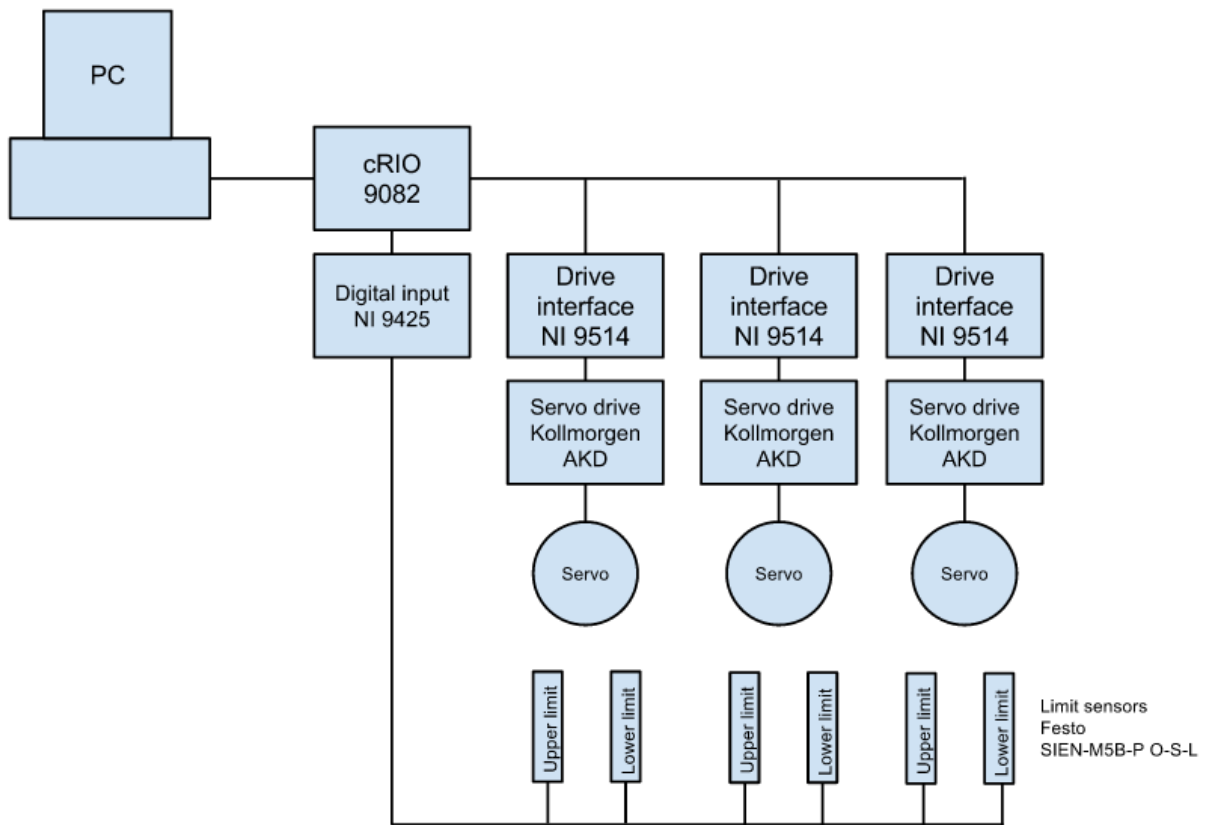
- NI CompactRIO controller en chassis om de RIO Scan interface te ondersteunen

Of

- NI 9144 8-Slot EtherCat Slave chassis
- Drie NI 9514 C Series Stepper Drive Interface met Encoder Feedback
- Drie Kollmorgen ADK servo drivers
- Power supply voor de controller
- Apparte power supply voor de modules
- Ethernet connectie en kabel

In dit project werd er de tweede optie gebruikt.





4. Hardware & Software combined

Om de LabVIEW code naar de Delta Robot te uploaden moest de robot worden geupdate. Deze stap was zeer makkelijk en snel gedaan binnen LabVIEW. Na deze stap werd er een nieuw project aangemaakt en werd de robot toegevoegd op dezelfde manier als de Solid Works tekening.

Alle stappen verliepen goed, enkel kon er geen beweging worden gebracht in de motoren. Na een hele tijd van problem solving werd er achterhaald dat de NI 9514 geen spanning gaf aan de ADK Servo Drivers. Door dit probleem kon de ADK ook geen motoren aansturen.

5. Projectverloop

Tijdens het project zijn er een aantal struikelblokken geweest, zo was de eerste les verloren gegaan naar het opzoeken en het installeren van de nodige modules. Eens dat uit de weg was en we beweging in de Solid Works tekening kregen besloot het tekenprogramma enkele keren raar te doen en alle onderdelen anders te plaatsen, dit probleem hebben we kunnen voorkomen door de simulatie niet in een onmogelijke positie te sturen.

Nu de motoren apart bewogen konden worden wilden we coördinaten ingeven, maar de kinematica die gebruikt werd was incorrect en is er veel tijd besteed aan het uitzoeken hoe deze juist in elkaar zat.

Nadat dat probleem uit de weg was was het tijd om de effectieve robot aan te sturen, ook dit verliep niet van een leien dakje. Na het uploaden van het programma zou het moeten werken, maar dat deed het niet. Na een grote speurtocht naar fouten werd er ondervonden dat de displays op de ADK Servo drivers niets weergaven. Alle spanningen werden hierdoor nagemeten en al gauw kon er geconcludeerd worden dat het probleem bij de NI 9514 lag, deze was niet verbonden aan de ADK, waardoor deze niet in werking trad. Het probleem kon echter te laat verholpen worden, waardoor er geen tests meer konden gedaan worden.

6. Toekomst

Nu gebruiken we de upper en lower sensoren niet, dit is echter wel aan te raden om zeker te zijn dat de robot niet in een ongewenste toestand komt. Verder kan de code uitgebreid worden door gebruik te maken van een gesimuleerde coördinate space om de besturing makkelijker te maken. Ook kan de code geïmplementeerd worden op de hardware.

7. Resources

Introduction to NI SoftMotion for SolidWorks

<http://www.ni.com/white-paper/9566/en/>

Getting Started with CompactRIO and LabVIEW

<http://www.ni.com/pdf/manuals/372596b.pdf>

Getting Started with NI 9514/16 C Series Modules and AKD Servo Drives

<http://www.ni.com/pdf/manuals/375516b.pdf>

Delta robot kinematics

<http://forums.trossenrobotics.com/tutorials/introduction-129/delta-robot-kinematics-3276/>

Circle-Circle Intersection

<http://mathworld.wolfram.com/Circle-CircleIntersection.html>