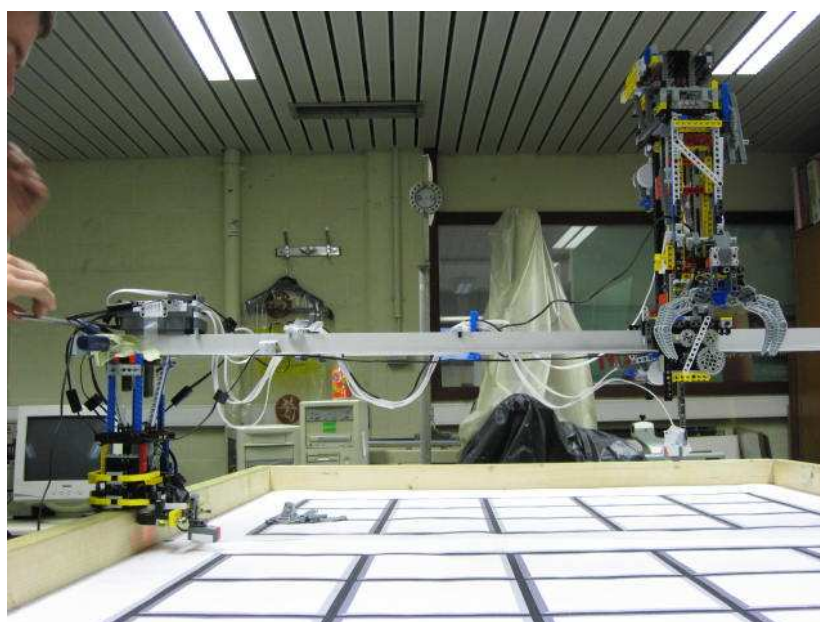


Verslag project Werktuigkunde- Elektrotechniek: Robot



Technisch verslag

Inleiding	4
Materiaal, Sensoren en Software.....	6
Materiaal.....	6
Sensoren	13
Software	16
Conceptvoorstellen.....	23
Voortbewegingsystemen	23
Detectiemethodes	30
Navigatiesystemen.....	33
Grijpsystemen.....	38
Opslagsystemen.....	43
Mogelijke combinaties	48
Argumentatie keuze kraanmodel.....	49
Pro	49
Contra.....	52
Samenvatting.....	52
Ontwerp en Bouw.....	53
Bouw van verschillende onderdelen.....	53
Toegevoegd materiaal	65
Het Programma.....	68
Besluit.....	72

[Boekhouding.....74](#)Reflectie

Goedele Van kerkhoven	78
Avdyl Shala	80
Pieter Ruelens.....	82
Zjef Van De Poel.....	83

Maarten Weckx	84
Hannes Maes	85
Adam Cooman.....	87

Bijlagen

Bijlage 1: Vakwerken.....	88
Bijlage 2: Onrealistische grijpsystemen	89
Bijlage 3: Source code van het programma.....	92
Bijlage 4: Berekening scheef trekken.....	98
Bijlage 5: Tandwielen	100

Technisch verslag

Inleiding

Tijdens de eerste zeven weken van het tweede semester van het tweede jaar Bachelor in de ingenieurswetenschappen werd er met een groep van 7 studenten gewerkt aan een project dat diende als praktische toelichting bij de studierichting Werktuigkunde-Elektrotechniek. De desbetreffende groep had gekozen voor het project 'Bouw een geautomatiseerde machine' dat als doel had dat de studenten praktische ervaring zouden opdoen inzake robotica en mechatronica. Andere keuzemogelijkheden waren het bouwen van een kart of een windturbine. De groep bestond uit de volgende studenten: Zjef Van de Poel, Goedele Van kerkhoven, Pieter Ruelens, Avdyl Shala, Maarten Weckx, Adam Cooman en Hannes Maes.

Voor de bouw van de robot mocht er gebruik gemaakt worden van een Lego Mindstorms NXT bouw pakket, indien nodig mochten ook andere bouwelementen worden toegevoegd. Om de robot op een autonome wijze de gevraagde opdracht te laten uitvoeren mocht er gebruik gemaakt worden van verschillende sensoren. De sturing van de robot gebeurde aan de hand van een controller die men op voorhand kon programmeren aan de hand van verschillende mogelijke programma's, ook deze worden verderop toegelicht. Om een duidelijk doel te hebben bij de ontwikkeling van de robot was er op het einde van de werkperiode een ballenspel voorzien waaraan de robot uiteindelijk moest kunnen deelnemen tegen de robot van een concurrerende groep.

Dit spel speelde zich af op een veld van ca 1.2 bij 1.5 meter waarop aan beide zijden een raster van vijftien (vijf bij drie) vakken was getekend. De lijnen van dit raster bestonden uit 3 verschillende grijswaarden. Het speelveld was bezaaid met ballen die langs de onderzijde waren afgeplat. De plaats van deze ballen was op voorhand niet gekend, enkel dat ze telkens op een snijpunt van twee lijnen lagen en dat ze aan beide zijden symmetrisch geplaatst zouden zijn lag vast. Het speelveld was omringd door een houten rand van ca. 10 cm. Beide robots startten in een hoek van hun eigen zijde en moesten gedurende vijftien minuten zoveel mogelijk ballen grijpen en in een gootje achter hun eigen zijde deponeren. Er lagen zowel rode als blauwe ballen op het veld, de blauwe ballen leverden één punt op, rode twee voor de blauwe ploeg en omgekeerd voor de rode ploeg. Wanneer een robot moest gereset worden, kostte dat de ploeg 2 punten. De ploeg die op het einde van het spel het meeste punten had, won. Een laatste regel was dat men niet bewust de robot van de tegenstander mocht beschadigen.

Het doel van het project was dus op een onderbouwde manier een zo efficiënt mogelijke robot te creëren. Het ballenspel was een manier om bepaalde concrete richtlijnen en vereisten te hebben voor de robot. Om aan de hand van dit spel een goede robot te maken was het dus belangrijk dat men goed besepte welke de specifieke vereisten waren:

- Stabiel ontwerp opdat de robot niet stuk zou gaan tijdens het spel.
- Efficiënt voortbewegingssysteem voor vlotte verplaatsing van de robot.

- Accuraat navigatiesysteem voor een goede coördinatie van de robot.
- Goede detectiemethode voor het vinden van de ballen.
- Accuraat grijpsysteem voor het opnemen van de ballen.
- Indien nodig een opslagsysteem voor het bijhouden van de ballen tijdens het spel.
- Een vlot werkend programma dat de verschillende commando's van de robot ten gepaste tijde zou doorsturen.

Om zo goed mogelijk aan deze vereisten te voldoen heeft men de volgende aanpak gebruikt:

Ten eerste heeft men een grondige studie van het beschikbare materiaal gedaan om te zien wat de mogelijkheden en beperkingen van het Lego Mindstorms NXT bouwpakket, de verschillende sensoren en de beschikbare software waren.

Daarna heeft men verschillende concepten bedacht op vlak van voortbewegingsystemen, navigatietechnieken, detectiemethodes, grijpsystemen, opslagsystemen en programmamogelijkheden. Deze zijn dan elk op een zeer grondige wijze bestudeerd. Uit de conclusies van deze verschillende conceptafwegingen heeft men dan de mogelijke combinaties van concepten beschreven die goed aan de besproken vereisten zouden voldoen.

Op basis van de studie van deze concepten en de mogelijke combinaties heeft men dan een beslissing gemaakt voor het ontwerp.

Binnen de uitwerking van dit ontwerp is men dan nog op ettelijke problemen gestoten. Voor de meeste hiervan is een oplossing gevonden.

Uiteindelijk heeft men dan een besluit getrokken in zake ontwerp en werkmethodes.

Aanvullend bij het technische verslag vindt de lezer de boekhouding van het gedane werk en de reflectie van de verschillende ontwerpers op mekaar en op het gedane werk.

Het team hoopt dat dit verslag de lezer duidelijkheid zal scheppen over de ontwikkeling van de ontworpen robot.

Materiaal, Sensoren en Software

Dit deel is de eerste stap in de ontwikkeling van de robot. Het is namelijk onzinnig om te spreken van vorm, constructie, besturing enz. indien men niet weet over welke middelen men beschikt. Daarom wordt er in dit deel een overzicht gegeven van de drie grote onderdelen van het beschikbare materiaal: materiaal, software en sensoren. Er werd voor deze indeling gekozen omdat deze drie soorten middelen ieder van zeer groot belang zijn in de robot en toch onafhankelijk van elkaar bestudeerd kunnen worden.

Onder materiaal verstaat men de beschikbare Lego en de mogelijke combinaties die hiermee gevormd kunnen worden. Hierin worden nog geen specifieke constructies of robotvormen besproken, dit behoort namelijk tot de conceptbespreking die nog niet aan de orde is. De bedoeling is aan te tonen wat de algemene mogelijkheden zijn van het Lego Mindstorms NXT bouwpakket.

Vervolgens worden de verschillende sensoren besproken omdat deze zullen dienen als de zintuigen van de robot. Aan een uitwerking van de robot beginnen zonder de eigenschappen van de sensoren te kennen, zou onzinnig zijn.

Het laatste deel van dit stuk bestaat uit een bespreking van de beschikbare software. Omdat men hier tussen verschillende programma's kan kiezen is het belangrijk dat men weet wat de verschillen zijn alvorens men een keuze maakt van het te gebruiken programma.

Materiaal

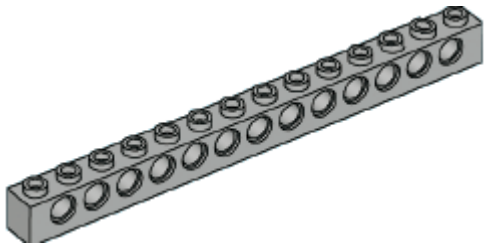
Wanneer men voor een project iets moet bouwen is een studie van het materiaal noodzakelijk. Voor dit project is dit hoofdzakelijk Lego, maar er mogen ook andere materialen gebruikt worden. Hierna volgt een gedetailleerde beschrijving van de meest voorkomende Lego-onderdelen en hun nut.

Afmetingen en afstanden

Afstanden in Lego zijn moeilijk uit te drukken in cm of zelfs in inch. De gebruikte afstand is de fundamentele Lego unit (FLU), dit is de breedte van een blokje met één nop op. 1 FLU is 5/16 inch of 0.79375 cm. De hoogte van een blokje is 3/8 inch of 6/5 FLU terwijl een plaatje een hoogte heeft van 1/8 inch of 2/5 FLU. Dit heeft als gevolg dat 3 plaatjes die opeen gestapeld zijn even hoog zijn als een blokje. Men kan ook een hoogte creëren die een geheel veelvoud is van de breedte van een blokje. Indien men een blokje en 2 plaatjes opeen stapelt is dit 2 FLU.

Balken, verbindingspinnen

Het basis legoblokje is de balk. De balk is een lang blokje met gaten in de zijkant. Deze gaten staan exact tussen 2 noppen, dus de afstand tussen 2 gaten is 1 FLU. Het aantal noppen is altijd even en het aantal gaten 1 minder.



De verbindingspinnen passen in de gaten die in de balken zitten en kunnen gebruikt worden om de balken zij aan zij te verbinden. Er zijn 2 soorten pinnen, zwarte en grijze.



De zwarte pinnen zijn iets groter en zitten steviger in de gaten. De grijze pinnen zitten er losser in en moeten dus gebruikt worden wanneer de 2 delen moeten kunnen roteren. Er bestaan ook langere verbindingspinnen waarop men 3 balken kan bevestigen i.p.v. 2.

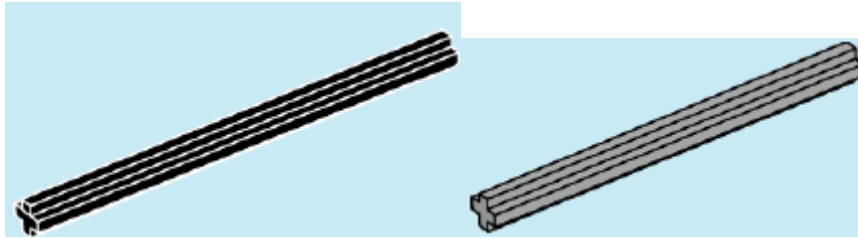


Verder zijn er nog pinnen met aan een kant een rond uiteinde dat in een gat past, en aan de andere kant een kruis zoals de staafjes.



Staafjes

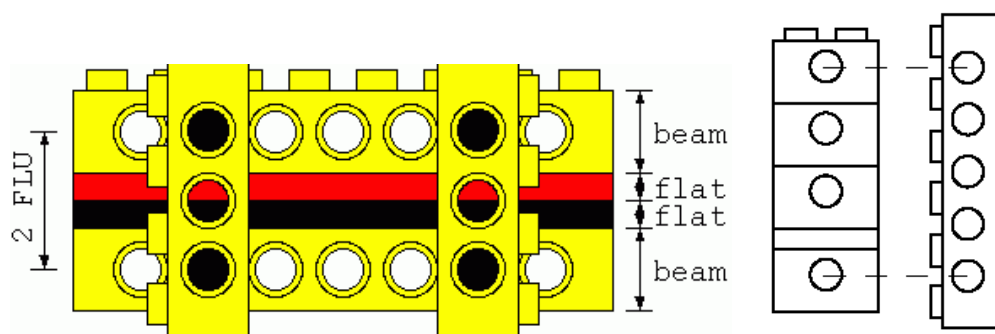
Legostaafjes zijn kruisvormig, dit heeft als gevolg dat men ze kan gebruiken om tandwielen aan te drijven, de staafjes kunnen niet vrij roteren in een kruisvormig gat. Er zijn grijze en zwarte staafjes.



De zwarte staafjes hebben een lengte van een even aantal FLU, de grijze een lengte van een oneven aantal FLU. Ook zijn de grijze staafjes iets dikker waardoor ze indien ze in een rond gat zitten, stroever draaien en dus voor meer wrijving zorgen.

Verstevingen

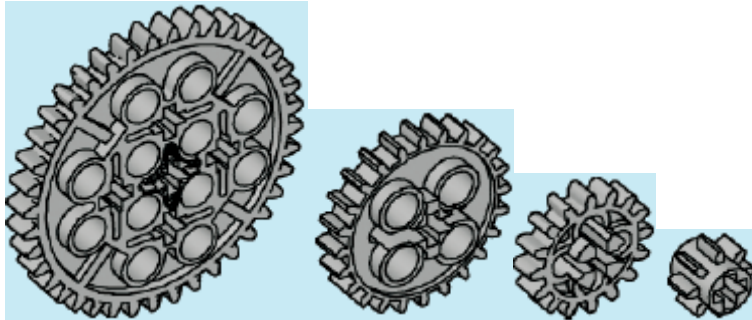
Als men de blokken gewoon opeen stapelt met behulp van de noppen, is de constructie niet echt stevig. Indien men er meer kracht op wil zetten zonder dat de blokken los komen moet men verstevingen aanbrengen. De meest gebruikte techniek is om de blokken zo te stapelen dat de afstand tussen 2 verticale gaten een geheel aantal FLU is, zodat als er verticaal een balkje wordt tegengehouden, de gaten exact overeenkomen en ze dus kunnen verbonden worden met pinnen.



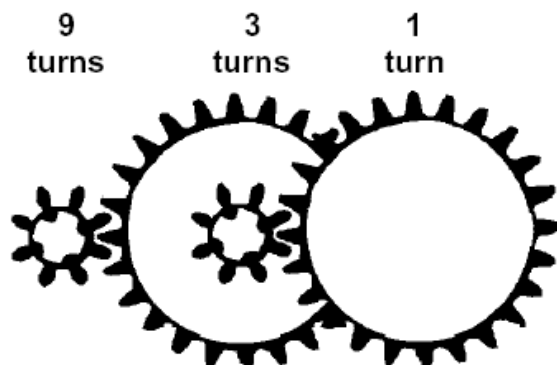
Een andere manier bestaat erin om de verstevingbalken schuin te hangen, dit biedt meer mogelijkheden, maar is vooral natte vingerwerk. Omdat hierbij de gaten soms niet exact overeenkomen moet men een beetje wringen

Tandwielen

In Lego zijn verschillende soorten tandwielen beschikbaar. De basis tandwielen zijn enkel geschikt om in een vlak te gebruiken, ze dienen om overbrengingen te creëren. Er zijn tandwielen met 40, 24, 16 en 8 tanden.



Hiermee kunnen verhoudingen gemaakt worden van 5:1, 3:1, 5:2, 2:1, 5:3, 3:2, 2:3, 3:5, 1:2, 2:5, 1:3 of 1:5. Men kan ook een verhouding maken dat een product is van deze verhoudingen door een groot en een klein tandwiel op hetzelfde staafje te monteren. In het geval er dus een fout zit op het aantal rotaties van een tandwiel door een fout op de aandrijving van de motor kan men deze zo beperken.



Er zijn ook platte tandwielen, deze zijn perfect om de rotatieas 90° te verdraaien. Het is wel onmogelijk om met deze tandwielen van ratio te veranderen.



De ronde tandwielen zijn iets slechter voor rechte hoeken, maar kunnen wel gebruikt worden om de rotatieas over een willekeurige hoek te draaien.



Deze tandwielen bestaan ook in verschillende formaten zodat men verhoudingen kan maken.

Verandering van richting

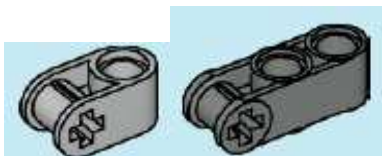
Het is in Lego niet simpel om van bouwrichting te veranderen. Hiermee wordt bedoeld een horizontaal blokje aan een verticaal te bevestigen. Een van de simpelste methodes is gewoon de noppen van de ene balk in de gaten van de andere te duwen.



Het probleem hiermee is dat de noppen iets te groot zijn voor de gaten en dit na een tijd vervorming van de Lego kan teweegbrengen. Een andere manier bestaat erin om de 'pin met nop' te gebruiken, maar deze manier is niet echt stevig.



Een betere manier bestaat erin om speciale blokjes te gebruiken, bijvoorbeeld één waar een kruisvormig gat in zit en daaronder, haaks op de richting van het kruisvormig gat, een of twee ronde gaten.



Een andere mogelijkheid is een blokje gebruiken waar 3 ronde gaten in zitten, 1 in elke richting.



Sensoren

Voor dit project kon gebruik gemaakt worden van een aantal sensoren:

- druksensor:



- afstandssensor:



- lichtsensor:



- geluidssensor:



- kompassensor:



Deze sensoren worden in een later deel van het verslag uitgebreider besproken.

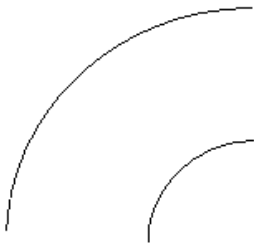
Motoren

Er waren drie servomotoren, en twee 9V motoren beschikbaar gesteld voor dit project.



Met deze motoren moest heel de robot worden aangedreven worden. Een van de problemen met de servomotoren is dat ze een vrij vervelende vorm hebben om in een constructie te bevestigen. Een ander probleem is dat er een fout op de motor zit en deze vrij groot is indien men de motor een korte beweging met weinig last laat doen. Deze fout kan kleiner gemaakt worden d.m.v. tandwieloverbrengingen zoals besproken in de paragraaf over tandwielen.

In dit verslag zal regelmatig gesproken worden over een differentieel. Een differentieel is een tandwielschakeling die toelaat om van één motor twee assen af te takken die met verschillende snelheid kunnen draaien. In wagens wordt een differentieel bvb. gebruikt om de wielen niet te laten doorslippen in een bocht.

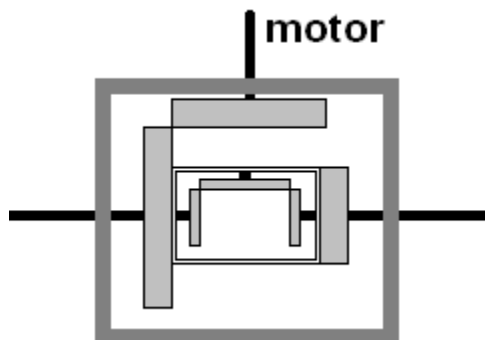


De weg die het wiel aan de binnenkant van de bocht moet afleggen is korter dan het wiel aan de buitenkant, het moet dus minder snel draaien. De differentieel laat dit toe.

De differentieel zal er voor zorgen dat de uitgang die het minst belast is roteert, dit zal van essentieel belang blijken bij o.a. grijpsystemen. Als een van de assen geblokkeerd is drijft de motor enkel de andere as aan, op deze manier kan men, door de assen op het juiste moment te blokkeren, één motor gebruiken voor twee bewegingen. In Lego ziet een differentieel er als volgt uit:



De twee assen die eruit lopen zijn duidelijk zichtbaar het is de bedoeling dat de motor het grijze omhulsel aandrijft. Dit kan gebeuren met nog een tandwiel:



Sensoren

Voor het testen van de beschikbare sensoren werden de waarden nagegaan die worden gegeven bij het scannen van de rand van het speelveld, de vloer van het speelveld en de twee verschillend gekleurde ballen. Hierbij werd vooral gelet op de afstand tot het te scannen object en de mate van variatie in waarden.

Lichtsensoren (stand 1)

Waarde bij contact:

- Blauwe bal: $\pm 18\%$
- Rode bal: $\pm 41\%$
- Rand: $\pm 6\%$

De waarde bij 'in het wilde weg' scannen: $50\% < x < 65\%$ (evenwichtswaarde)

We zien dat er een duidelijk verschil is tussen de evenwichtswaarde en de waarde voor de te scannen objecten. Voor de rode ballen ligt de waarde weliswaar dicht bij de evenwichtswaarde, maar in principe zou een daling van circa 10% duidelijk herkenbaar moeten zijn voor het besturingsprogramma.

Het probleem echter is dat de lichtsensoren in stand 1 pas van op korte afstanden een duidelijke, voor het programma herkenbare, daling weergeeft. Voor blauwe ballen begint hij pas aanzienlijk in waarde te dalen als hij in een straal van 7cm, om de bal, loodrecht op de bal is gericht, voor rode ballen is dit een straal van 1cm. De rand geeft pas een waarde van 6% indien de sensor hem raakt, een beetje minder contact kan al een veel hogere waarde geven.

Besluit:

- Op lange afstanden moeilijk als herkenningssensor te gebruiken
- Uitstekend als korte afstandsensoren
- Aangeraden om op lange afstand een andere sensor te gebruiken om objecten te detecteren en de lichtsensoren als bevestiging te gebruiken bij het benaderen van de gevonden objecten
- Botsing met de rand detecteren is mogelijk, indien loodrechte botsing gegarandeerd kan worden

Lichtsensoren (stand 2)

De tweede stand van de lichtsensoren heet 'reflected', hierbij wordt door de sensor licht uitgezonden en het percentage gereflecteerd licht gemeten.

Waarde bij contact:

- Blauwe bal: $\pm 21\%$
- Rode bal: $\pm 49\%$
- Rand: $\pm 37\%$

Evenwichtswaarde: $30\% < x < 40\%$

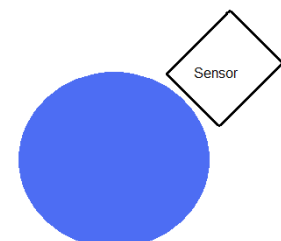
Het is misschien interessant om op te merken dat bij deze stand de waarden voor rode en blauwe ballen niet allebei onder de evenwichtswaarde liggen. Zo kan het stuurprogramma misschien geprogrammeerd worden om sneller beslissingen te nemen. Indien de gemeten waarden onder de evenwichtswaarde zakt, kan namelijk meteen worden aangenomen dat een blauwe bal gevonden werd. Bij stand 1 moet eerst worden gewacht op een stabiele waarde, want zowel de rand als de twee soorten ballen kunnen gevonden zijn bij een daling in waarde.

Het is ook opmerkelijk dat bij detectie van de rand een waarde van ongeveer 37% wordt gegeven, terwijl een waarde van 60% gelezen werd op enkele millimeters van de rand. Dit sluit betrouwbare detectie van de rand met de lichtsensoren in stand 2 uit.

De sensor geeft waarden die verder van de evenwichtswaarde liggen indien er in een hoek wordt gescand. Hiermee wordt het probleem van afstand echter niet uit de weg gegaan.

Besluit:

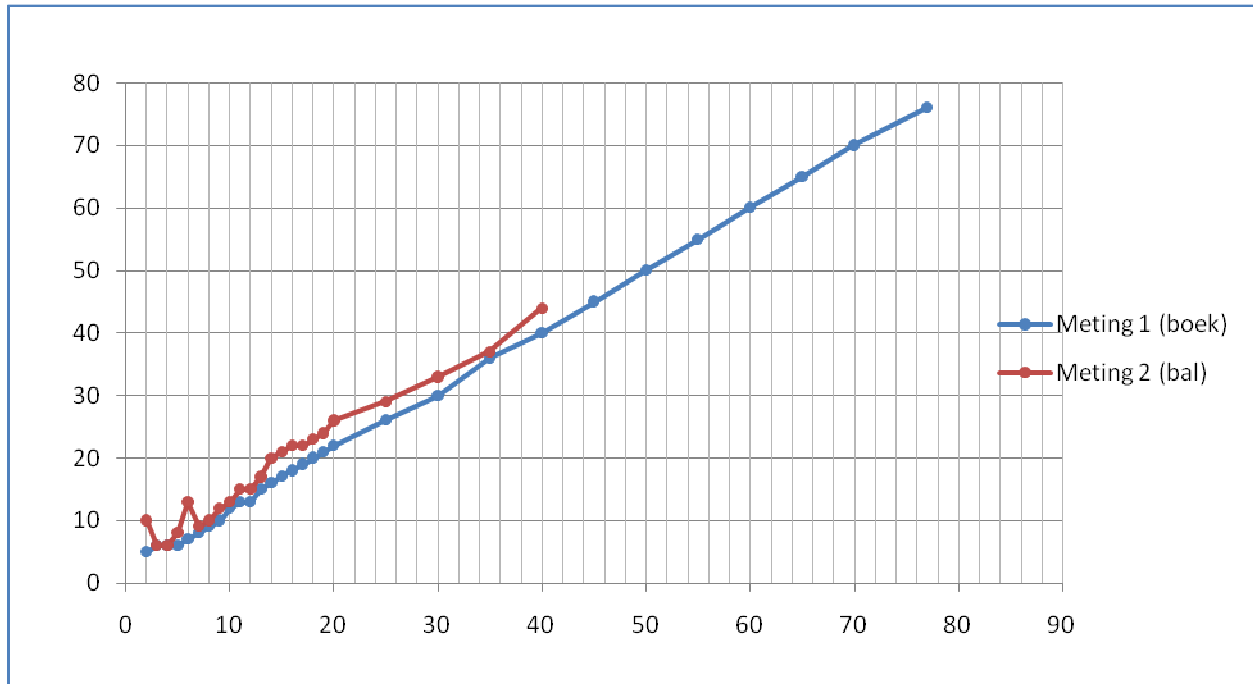
- Net zoals bij stand 1 is er het probleem van afstand
- De sensor in een hoek monteren geeft betere waarden
- De waarden voor blauwe en rode ballen liggen niet allebei aan dezelfde kant van de evenwichtswaarde



- Detectie van de rand met deze sensor is wellicht niet betrouwbaar

Afstandsensor

Voor het testen van de afstandsensor heeft men de gemeten waarden in functie van de afstand tot het te meten object in grafieken gegoten. De metingen gebeurden gewoon door de afstandsensor horizontaal te houden en recht op het te detecteren object te richten. De eerste metingen werden op een vlak object (een boek) uitgevoerd, de tweede op een bal.



Het eerste dat opvalt is dat onder 5cm de afstandsensor helemaal niet betrouwbaar is. Volgens de beschrijving van de sensor in de handleiding van Lego, werkt de sensor optimaal bij het detecteren van vlakke objecten (bv. muren, de rand en het speelveld) tot op een afstand van 255cm. De metingen bevestigen dit. Dit betekent dat het moeilijker is om ballen met deze sensor te detecteren. Voor de detectie van ballen is deze sensor echter wel betrouwbaar op een afstand tussen 5 en 20cm.

Daarbij komt nog de werking van de sensor. De sensor stuurt ultrasone golven uit en meet de tijd die deze nodig hebben om de sensor opnieuw te berekenen. Net zoals het navigatiesysteem van een vleermuis. Dit doet hij door middel van twee 'ogen', waarmee hij golven uitzendt en terug opvangt. Het probleem hierbij is dat het gemiddelde van de twee ogen wordt genomen, dus als enkel 1 oog op een bal gericht is, gaat een foute waarde worden uitgelezen. Het is dus belangrijk dat de sensor goed gepositioneerd wordt. Bij het programmeren van het stuurprogramma kan wel gezocht worden naar minima bij het uitlezen van de gemeten waarden.

Besluit:

- De sensor is uitstekend voor randdetectie

- Voor baldetectie best in samenwerking met een sensor die op kortere afstanden (<5cm) goed werkt (lichtsensor)

Druksensor

De druksensor geeft ofwel een waarde 0 (geen druk) ofwel een waarde 1. Om een bal te detecteren is de druksensor niet gevoelig genoeg, hij zal de bal gewoon wegduwen, tenzij er een tegengestelde kracht op de bal wordt uitgeoefend. De rand detecteren is geen probleem.

Besluit:

- Uitstekend voor randdetectie
- Niet bruikbaar voor baldetectie

Kompassensor

De kompassensor dient te worden gebruikt in de stand van de afstandsensor, dit betekent dat hij een waarde in cm geeft. Deze waarde is eigenlijk het aantal graden dat de sensor ten opzichte van het noorden is verdraaid. De waardes nemen toe in wijzerzin. Gericht naar het noorden geeft de sensor echter de waarde '?????', dit kan misschien problemen geven bij de programmatie, maar is zeker niet onoverkomelijk.

De sensor reageert vrij snel op een verandering van richting en het is mogelijk om de sensor 360° te laten roteren, zonder dat er waardes worden overgeslagen. Hij blijft ook consequent: indien hij een waarde x geeft voor een bepaalde richting, zal hij diezelfde waarde ook geven bij het terugkeren naar die positie. De kompassensor is een uitstekend oriëntatiemiddel, indien goed gekalibreerd ten opzichte van de muren van het speelveld.

Besluit:

- Reageert snel, toch snelheid rotatie robot hierop afstemmen
- Na kalibreren uitstekend oriëntatiemiddel
- Graden = cm
- Het noorden wordt aangeduid door '??????'

Software

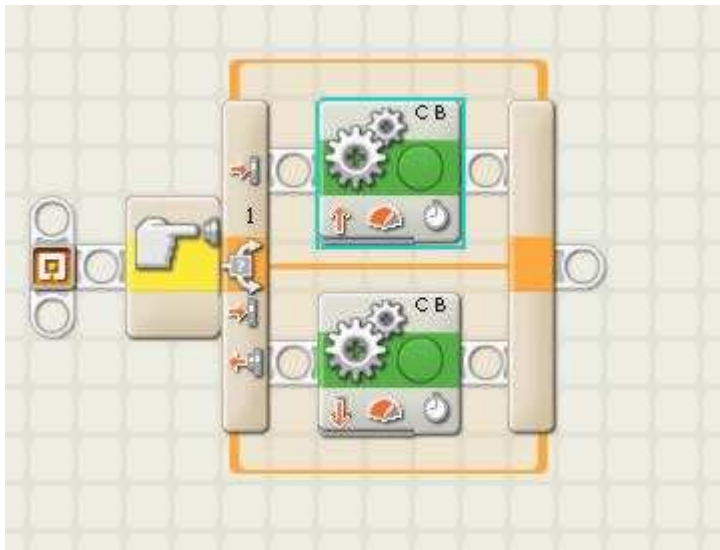
De code van het programma is voor de robot wat de hersenen zijn voor de mens; het moet 'nadenken' wat er gedaan moet worden, zet dit daarna om in commando's die doorgestuurd moeten worden naar de motoren en sensoren, en zorgt ervoor dat dit op een gecontroleerde manier gebeurt.

Hiertoe zijn er verschillende mogelijkheden om dit te verwezenlijken; er zijn verschillende programmeertalen en -omgevingen waarmee men het programma kan verwezenlijken, waaronder:

- -NXT-G
- -MindSqualls in Microsoft Visual Studio (C#)
- -NxtSharp in Microsoft Visual Studio (C#)

Keuze van de taal

NXT-G:



Deze is wezenlijk verschillend van de twee andere opties. Het geschreven programma wordt geuploadet in de Nxt-brick, waar het wordt uitgevoerd, in tegenstelling tot de andere methodes waar het programma op de computer uitgevoerd moet worden.

Het is duidelijk dat dit een snellere manier is om de robot te sturen; doordat de code lokaal in de Nxt-module uitgevoerd wordt, heeft men geen bluetoothconnectie nodig.

Doordat deze software van Lego zelf komt, is er een betere aansturing van wat de motoren betreft: een rotatie van X graden zal nauwkeuriger uitgevoerd worden.

Echter, doordat deze software duidelijk gericht is op een doelgroep die ettelijke jaren jonger is dan de personen van dit project, is er een grote vereenvoudiging gemaakt van de programmeer omgeving, waardoor men wel gemakkelijker simpele programmaatjes kan schrijven (zonder in staat hoeven te zijn om te kunnen programmeren), maar het moeilijk wordt om complexere zaken te gaan implementeren.

Voordelen:

- ✓ snellere uitvoering van programma
- ✓ nauwkeurigere bewegingen

Nadelen:

- moeilijk complexere dingen te programmeren
- niet vrij verkrijgbaar om ook na de uren aan het programma verder te kunnen werken

MindSqualls/NxtSharp

Beide zijn gelijkaardig.

Zoals eerder vermeld, wordt de code hier in C# geschreven met behulp van Microsoft Visual Studio. De gecompileerde code wordt dus op de computer uitgevoerd, en de commando's via bluetooth doorgestuurd, wat een vertraging oplevert. Deze kan ervoor zorgen dat de motoren verder draaien dan gewenst (De grootte van deze fout is niet erg constant, zodat het onmogelijk is om dit softwarematig te corrigeren).

Voordelen:

- ✓ Visual Studio is thuis gratis downloadbaar
- ✓ beter programmeerbaar

Nadelen:

- via bluetooth
- problemen met aandrijving motoren (zie later)

Na afwegen van deze voor- en nadelen is de keuze gevallen voor te programmeren in Visual Basic.

De keuze voor NxtSharp boven MindSqualls wordt duidelijker wanneer men de sensoren gaat testen.

Men neemt het volgende testsysteem: 1 motor en 1 sensor (in volgend voorbeeld de druksensor) worden bevestigd aan de brick. Het programma laat de motor continu roteren, terwijl de druksensor een commando geeft om te remmen wanneer deze wordt ingedrukt.

Wat men waarneemt is dat bij het programma geschreven met behulp van NxtSharp de motor quasi instantaan stopt met roteren nadat men de sensor heeft ingedrukt, in tegenstelling tot MindSqualls waarbij men een vertraging kan vaststellen van om en bij de 250-500 ms.

In principe is dit eenvoudig softwarematig op te lossen door de motoren, na elke remming opgeroepen door de sensoren, een kleine rotatie te laten ondergaan in de omgekeerde richting, om zo de vertraging van het remmen, en dus het te ver rijden van de robot, te compenseren. Maar aangezien dit quasi volledig automatisch opgelost wordt in NxtSharp, en voor de rest beide identiek schijnen, is de keuze voor NxtSharp evident

Echter al snel bleek er toch een verschil op te duiken tussen de twee, in het nadeel van NxtSharp; de tachocount, een variabele die het aantal rotaties van de motor bijhoudt, een soort kilometerteller van een auto, kan hier niet uitgelezen worden. Echter door wat onderzoek in de code werd duidelijk dat intern er wel degelijk een tachocount werd bijgehouden door het programma, maar deze nergens van buitenaf toegankelijk was. Maar door de eigenlijke code van NxtSharp wat aan te passen, kan deze variabele wel uitgelezen worden. De manier waarop is te vinden in het deel 'Vastgestelde Problemen'.

De mogelijkheden van het programma, uitlezen van motoren en sensoren

Hier volgt een summiere opsomming van de verschillende tools, betreffende de aansturing van de robot, beschikbaar gesteld door NxtSharp.

Motoren:

- laten roteren over een hoek, met een gegeven vermogen *
- oneindig laten roteren
- uitlezen van het aantal graden dat een motor heeft afgelegd, de tachocount (na kleine aanpassing code in het geval van NxtSharp)*
- laten remmen op twee verschillende manieren: de motoren blokkeren (Brake), of uit laten draaien (Coast)

Sensoren:

- de waarde van de sensor uitlezen. Opmerking: dit is niet de huidige waarde, maar de waarde die de sensor bij de laatste uitlezing heeft teruggegeven
- de sensor eenmalig zijn waarde laten updaten
- automatische uitlezing met instelbare frequentie
- functie oproepen wanneer de sensor uitgelezen wordt
- functie oproepen wanneer de waarde veranderd is.

*enkel mogelijk met de servomotoren, niet de 9V motoren

Vastgestelde Problemen/Tekortkomingen en oplossingen

Onnauwkeurige rotaties

Wanneer men als test de motor tweemaal over 360° laat roteren met het vermogen op 100%, dan merkt men dat dit commando niet uitgevoerd wordt zoals men zou verwachten. Wat men vaststelt is dat er zich een grote afwijking voordoet op beide rotaties. De eerste rotatie is te groot, ongeveer 650° , te wijten aan het grote vermogen, terwijl men een kleine rotatie oplegt en geen belasting aan de motor hangt; de motor roteert te ver.

De interne werking van de motoren tracht dit te corrigeren bij de tweede rotatie; er wordt bijgehouden hoeveel graden de motor gedraaid zou moeten hebben. Toegepast op de test levert dit de volgende situatie:

Het eerste commando is een rotatie over 360° , maar de motor roteert 650° . Het tweede commando is weer een rotatie van 360° . De totale gewenste rotatie is nu 720° . Dit is het punt dat de motor wil bereiken. De motor zal nu dus geen 360° (of de overshoot van 650°) draaien, maar ongeveer 70° om zo 720° te bereiken.

Dit vormt een probleem wanneer men een enkele, nauwkeurige rotatie wil opleggen; het omhoog heffen van de grijper van de robot bijvoorbeeld.

Echter blijkt het probleem niet zo'n echt probleem te zijn als ingeschat. Het is grotendeels het resultaat van een slecht geconditioneerde test opstelling. Een kleine rotatie, waarvoor men veel vermogen geeft, en bovendien geen belasting aan de motor hangt, is een weinig voorkomende reële situatie. Doordat in de werkelijkheid een belasting op de motor bevestigd wordt, is de werkelijke overshoot kleiner. Een andere vaststelling is ook dat deze onnauwkeurigheid verkleint naarmate men het aantal rotaties verhoogt; 20 volledige omwentelingen, zelfs zonder belasting, resulteert in ongeveer 19.5 omwentelingen, een relatieve fout van 2.5%.

Geen tachocount

Zoals eerder gesteld lijkt er geen tachocount, waarvan kennis een beter zicht kan geven op de precieze locatie van de robot, te zijn in NxtSharp, maar op volgende manier is deze toch beschikbaar voor de programmeur:

In Bram.NxtSharp opent men de klasse NxtCommunicator

Men zoekt de functie GetOutputState

Hier valt meteen op dat tachocount hier toegewezen wordt

Kijkt men nu naar de methode openSerialPorts in NxtBrick, ziet men dat hier een object van

NxtCommunicator wordt aangemaakt

Men verandert private void openSerialPorts naar

Private NxtCommunicator openSerialPorts en men geeft het aangemaakt object mee

In dezelfde klasse gaat men naar Connect waar bovenstaande functie wordt opgeroepen,

waar men het object dat nu meegestuurd wordt met openSerialPorts op zijn beurt

meegeeft met Connect

In het zelf geschreven programma stockeert men dit object, waardoor de methode

GetOutputState, en dus meteen ook de tachocount van de motor, van hieruit aanspreekbaar is.

Problemen met autopoll sensors

De sensoren kunnen softwarematig met een bepaalde frequentie uitgelezen worden met behulp van 'autopoll'. Er traden echter problemen op in een testprogramma waarbij de sensors niet automatisch uitgelezen werden. Dit deed zich voornamelijk voor wanneer men de genoemde autopoll in- en uitschakelde.

Een eenvoudige oplossing bestaat erin om niet te vertrouwen op de autopoll verschaft door NxtSharp, maar deze in de geschreven software te incorporeren.

Men creëert op volgende wijze een autopoll die gestopt kan worden wanneer aan een bepaalde voorwaarde voldaan wordt.

```
...
sensor.Polled += new SensorEvent(Method);1
...
sensor.Poll();2
...
void Method(NxtSensor sensor)
{
    if (...)3
    {
        ...
    }
    else
    {
        sensor.Poll();4
    }
}
```

1. De methode die opgeroepen moet worden bij de uitlezing van de sensor
2. Eenmalig uitlezen van de sensor om de autopoll in te schakelen
3. Vergelijking die de autopoll uitschakelt
4. De sensor opnieuw uitlezen, men belandt dus opnieuw in de methode 'Method'

Wachten tot motor stilstaat

NxtSharp bevat geen mogelijkheid om te controleren of een motor aan het roteren is of niet. Dit is echter nodig wanneer men een motor laat draaien, en nadat deze uitgevoerd is pas een ander commando wil sturen.

Wanneer men volgend voorbeeld in code zou omzetten en zou uitvoeren,

```
Motor.turn(power,angle);
```

```
Other Command;
```

merkt men dat beide tegelijk uitgevoerd zullen worden.

Door echter gebruik te maken van de eerder vermelde manier om de tachocount van een motor uit te lezen, kan men een methode schrijven die wacht totdat de motor stilstaat. Wanneer de motor nog roteert, zal de tachocount veranderen. Men dient te wachten totdat deze teller niet meer verandert.

```
private void WaitWhileRunning(NxtMotor motor)
{
    int tacho,prevtacho=0;

    while (true)
    {
        tacho = GetTacho(motor);
        Thread.Sleep(300);
        if (tacho == prevtacho)
        {
            break;
        }
        prevtacho = tacho;
    }
}
```

Conceptvoorstellen

De volgende belangrijke stap in de ontwikkeling is de concrete uitwerking van verschillende mogelijke concepten. In dit deel wordt er nog geen besluit genomen over het uiteindelijke ontwerp. In deze bespreking wordt de volgende volgorde gehanteerd:

1. Voortbewegingsystemen
2. Detectiemethodes
3. Navigatiesystemen
4. Grijsystemen
5. Opslagsystemen

Er werd bewust gekozen om programmatiemogelijkheden niet bij deze conceptbesprekingen te voegen omdat de programmatie net gebaseerd is op het gekozen concept. Ook zijn de mogelijkheden en verschillen in de software al grondig besproken in het vorige hoofdstuk. Op het einde worden enkele mogelijke vormen van robots, dit zijn dus combinaties van de besproken concepten, voorgesteld waaruit men dan een keuze kan maken voor het uiteindelijke ontwerp.

Voortbewegingsystemen

In dit stuk bespreekt men de mogelijke manieren voor het voortbewegen van de robot, onafhankelijk van de vorm en de navigatie en zonder rekening te houden met de ballen op het speelveld. Men gaat ervan uit dat men voor de voortbeweging van de robot maximaal twee motoren ter beschikking heeft. Voor de grijsystemen is er immers altijd minstens 1 motor vereist. Het beschikbare materiaal bestaat uit een rupsbandenpaar en een viertal wielensets. Omdat er maar twee motoren ter beschikking zijn zal men zich dus beperken tot enkel rijsystemen. Een andere mogelijkheid zou een stappende robot of een robot met vier poten zijn maar deze zijn veel moeilijker bestuurbaar en zijn ook niet nauwkeurig. Er wordt voor elk onderdeel een onderscheid maken tussen één- en tweemotorige systemen tenzij overbodig.

Rupsbanden

Een groot voordeel ten opzichte van de meeste wielsystemen is dat men met rupsbanden op een theoretisch eenvoudige manier ter plaatse kan roteren.



Voor de navigatie is het essentieel dat de robot ter plaatse kan roteren. Indien dit niet het geval is creëert men veel makkelijker fouten op de plaatsbepaling die niet in te schatten zijn.

Bij het uitdenken van dit idee botst men op twee essentiële vereisten:

- beide rupsbanden moeten zowel in dezelfde als in tegengestelde zin kunnen draaien aan dezelfde snelheid
- Raakvlak van beide rupsbanden met de grond heeft een grote invloed op het rotatiepunt

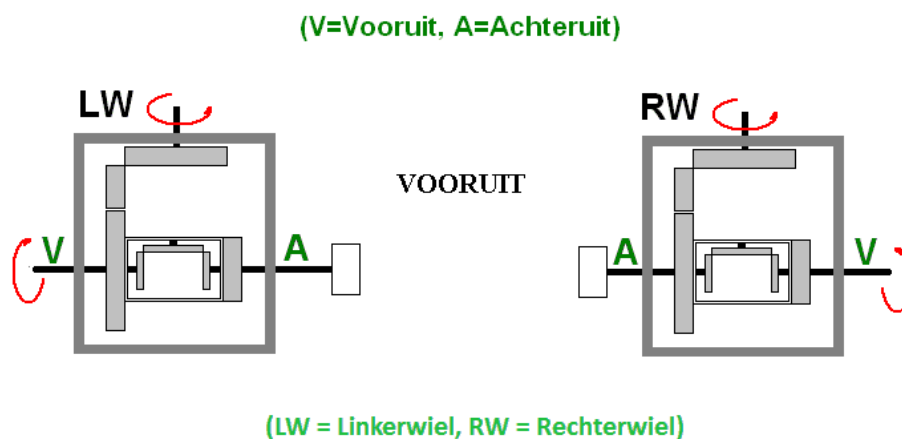
Praktische uitvoering

Moeilijkheid 1: Beide rupsbanden moeten zowel in dezelfde als in tegengestelde richting kunnen draaien aan dezelfde snelheid.

Eenmotorig systeem: Wanneer er maar één motor ter beschikking is, heeft men in het geval van rupsbanden altijd twee differentieelsystemen nodig. De belangrijkste moeilijkheid zit er namelijk in dat beide rupsbanden zowel in dezelfde als in tegengestelde richting moeten kunnen draaien. De twee differentieelsystemen zouden dan steeds gezamenlijk door die ene motor aangedreven worden, anders gezegd: wanneer één rupsband draait, kan de andere nooit stil staan. Dit is natuurlijk net wat gewenst is om ter plekke te roteren, maar het brengt ook wat complicaties met zich mee. Toelichting:

Elke band moet aan een differentieelsysteem gekoppeld worden waardoor hij zowel vooruit als achteruit kan draaien.

Stel dat de robot vooruit wil rijden.(achteruit is analoog)



Om de robot op rupsbanden rechtdoor te laten rijden moeten de linkerband en de rechterband vooruit draaien. Daarom moet zowel de as voor het achteruitdraaien van de linkerband als de as voor het achteruitdraaien van de rechterband geblokkeerd worden. In het geval van achteruitrijden moeten de assen voor het vooruitrijden van beide banden geblokkeerd worden.

Stel nu dat de robot naar links wil draaien. (rechts is analoog)

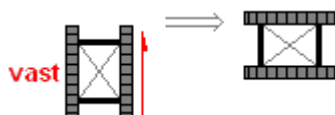


Wanneer men de robot in tegenwijzerzin wil laten roteren moet de linkerband achterwaarts en de rechterband voorwaarts draaien. In dit geval moet dus de as voor het vooruitdraaien van de linkerband en de as voor het achteruitdraaien van de rechterband geblokkeerd worden. Omgekeerd voor het roteren in wijzerzin.

Men ziet dus dat voor voorwaartse en achterwaartse beweging en rotaties in wijzer –en tegenwijzerzin telkens andere belastingscombinaties nodig zijn. Aangezien met het programma enkel de motor kan bestuurd worden die beide differentieelsystemen aandrijft is het dus onmogelijk om deze verschillende belastingscombinaties op het juiste moment te verwezenlijken.

(Wanneer eenmotorige voortbewegingsystemen in verder onderdelen van deze bespreking worden ‘overgeslagen’ is dat omdat het hierboven vermelde probleem ook daar van toepassing is.)

Tweemotorig systeem: Wanneer men twee motoren ter beschikking heeft wordt alles veel eenvoudiger, men heeft namelijk geen differentieelsystemen meer nodig. In dit geval zou men aan elke rupsband één motor koppelen. Deze zouden dan afzonderlijk bestuurd kunnen worden door het programma waardoor de voordien besproken bewegingen zeer eenvoudig zouden zijn te verwezenlijken. Verder zou het ook mogelijk zijn om een niet-lokale rotatie uit te voeren.



Hierbij zal één wiel wel steeds voor een kleine blokkering zorgen wat de beweging vrij stroef maakt. Zoals hierboven vermeld is een niet-lokale rotatie ook af te raden.

Conclusie moeilijkheid 1: Een vlotte beweging op rupsbanden is enkel mogelijk wanneer men beschikt over twee motoren.

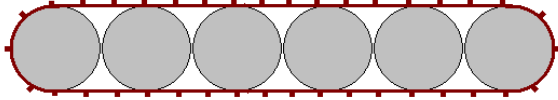
Moeilijkheid 2: Raakvlak van beide rupsbanden met de grond heeft een grote invloed op het rotatiepunt.

Een tweede grote moeilijkheid bestaat erin dat bij het gebruik van rupsbanden het rotatiepunt nooit exact gekend is bij een imperfect vlak oppervlak. Hierdoor zal er steeds een grote fout zitten op het ter plaatse roteren wat zeer ongunstig is voor de navigatie. Zonder bestuurder die deze rotatiefout kan corrigeren is een robot op rupsbanden dus zeer moeilijk te programmeren omdat de fout volledig afhankelijk is van de oneffenheden in het oppervlak en dus onmogelijk is te voorspellen.

Conclusie moeilijkheid 2: Rupsbanden zullen steeds zeer ongunstig zijn voor de navigatie.

Bijkomende nadelen

Een eerste bijkomend nadeel is het feit dat de rupsbanden voorzien zijn van ‘tandjes’. Hierdoor gaat de robot trillen wat in de eerste plaats nadelig is voor de sensoren. Ook kunnen er delen van de robot losstrillen.



Het tweede nadeel zit hem in de snelheid van een robot op rupsbanden. De rupsbanden kunnen wel veel meer massa verdragen dan de meeste wielconstructies maar vertragen de beweging zeer sterk.

Conclusie: Het grote nadeel van een robot op rupsbanden is dat er steeds grote fouten ontstaan bij rotaties, hierdoor zijn ze zeer ongunstig voor de navigatie. Verder zijn er altijd twee motoren nodig voor de aandrijving. Een voordeel is dat wanneer de massa van de robot zeer groot dreigt te worden rupsbanden een goede redding zullen zijn, hiertegenover staat wel snelheidsverlies.

Wielen

De vuistregel bij de variatie in wiel aantallen is dat met een grotere hoeveelheid wielen een grotere stabiliteit en met een lagere hoeveelheid wielen een grotere wendbaarheid gepaard gaat. De uitdaging bevindt zich dus in het vinden van een compromis.

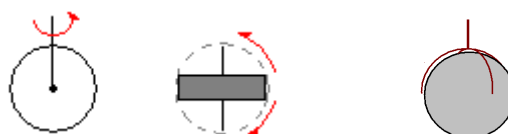
In wat volgt zal men de verschillende mogelijke wiel aantallen bespreken en hierbinnen telkens ingaan op positie en aandrijving. Op deze manier zal men het beste compromis proberen vast te stellen. Over de variatie in grootte wordt hier niet verder uitgeweid, de besproken systemen zijn zowel mogelijk voor kleine als voor grote wielen.

2+steunpunt

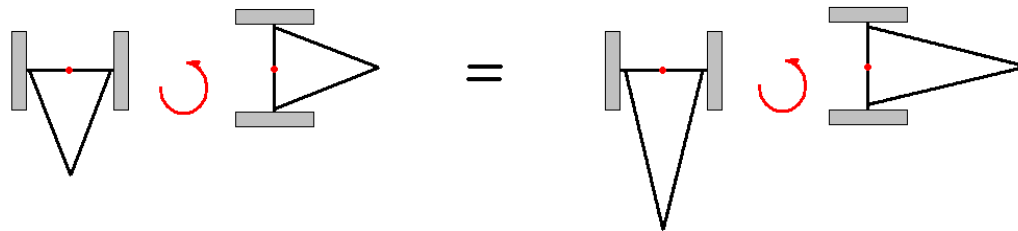
De meest eenvoudige maar in het dagelijkse leven niet vaak gebruikte wielconstructie is die bestaande uit twee wielen. Voor het bewaren van het evenwicht moet er dan een derde steunpunt worden gezocht. Men kan dit beschouwen als een alternatief voor een driewieler zonder stuurman.

Het steunpunt kan bestaan uit een legoblokje, hiermee gaat bij het roteren wel slip gepaard gaan, of een dummywiel.

Dit dummywiel zou men kunnen maken door een wiel gemonteerd op een roterende as, men zou als alternatief een balletje kunnen nemen dat men ‘vast’ houdt zodanig dat het wel nog kan rollen:



Om dezelfde reden als deze besproken bij het gebruik van rupsbanden zullen hier ook steeds twee motoren nodig zijn. Hierdoor is het dus wel, op dezelfde manier als bij rupsbanden, mogelijk om ter plaatse te roteren wat zeer gunstig is voor de navigatie. Een ander voordeel aan dit systeem is dat bij het maken van een rotatie volgens het rupsbandensysteem, al dan niet lokaal, het rotatiepunt nu wel duidelijk bepaald is, onafhankelijk van de massaverdeling.



De grootte van de driehoeken duiden op de massaverdeling.

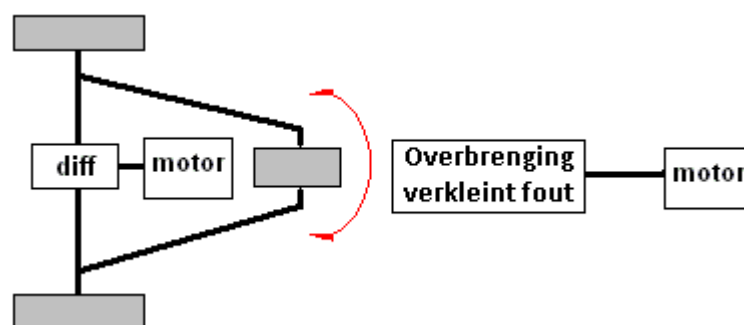
Omdat men de massaverdeling nu vrij kan kiezen, heeft men meer vrijheid bij het creëren van stabiliteit.

Besluit: De voortbeweging op basis van het gebruik van twee wielen en een steunpunt is een zeer goede methode omdat dit veel voordelen geeft op vlak van wendbaarheid, stabiliteit en navigatie.

3 wielen

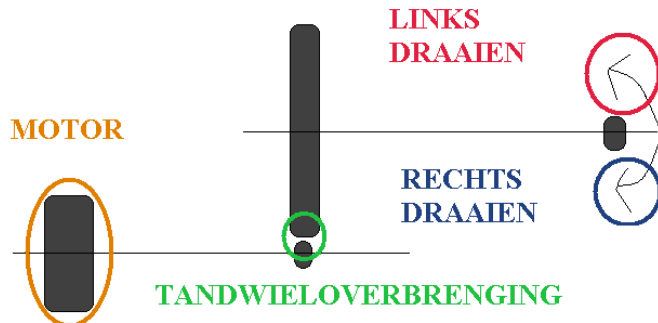
2 + stuurwiel:

Een andere voortbewegingsmethode, vergelijkbaar met de vorige, is een methode waarbij men gaat trachten de bestuurder van een driewieler na te bootsen:



Zoals te zien op de tekening is dit een vrij eenvoudige constructie maar er treden wel enkele nadelen op. Wanneer men slechts één motor gebruikt voor de achterwielaandrijving is de mogelijkheid van het lokale roteren al direct over boord geworpen, wat net een essentieel voordeel was voor de navigatie. Verder moet men nog een extra differentieel bouwen voor de achterste as omdat men anders te maken heeft met een starre as waardoor de achterste wielen gaan slippen en er dus een fout ontstaat op de navigatie. Een ander nadeel is dat men met behulp van het programma het voorste wiel in een juiste

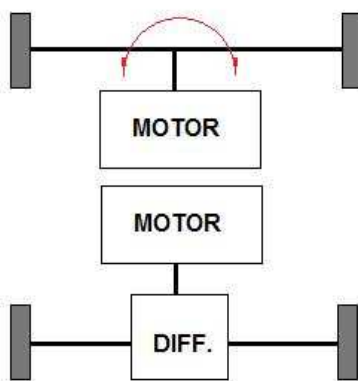
hoek gaat moeten zetten wanneer men de driewieler een bocht wil laten nemen. Hierop is het gevaarlijker om fouten te maken maar met enkele tandwieloverbrengingen zou deze fout wel kunnen verminderd worden (zie 'materiaal'). Deze tandwieloverbrenging zou er dan bvb. als volgt uitzien:



Mits de juiste berekeningen van de tandwieloverbrengingen (zie 'materiaal') kan men dan bepalen hoe groot de hoekverandering is die overeenkomt met het draaien van de motor. Hoe meer men de fout wil verminderen hoe trager het verdraaien van het wiel dus gaat verlopen. Buiten het feit dat de massa wat eerlijker verdeeld is door het gebruik van de twee motoren biedt deze methode niet veel voordelen ten opzichte van de vorige. Er zijn evenveel motoren vereist maar het systeem is trager en men verliest een grote troef bij de navigatie.

Besluit: Buiten het feit dat de massa wat eerlijker verdeeld is door het gebruik van de twee motoren biedt deze methode niet veel voordelen ten opzichte van de vorige. Er zijn evenveel motoren vereist maar het systeem is trager en men verliest een grote troef bij de navigatie.

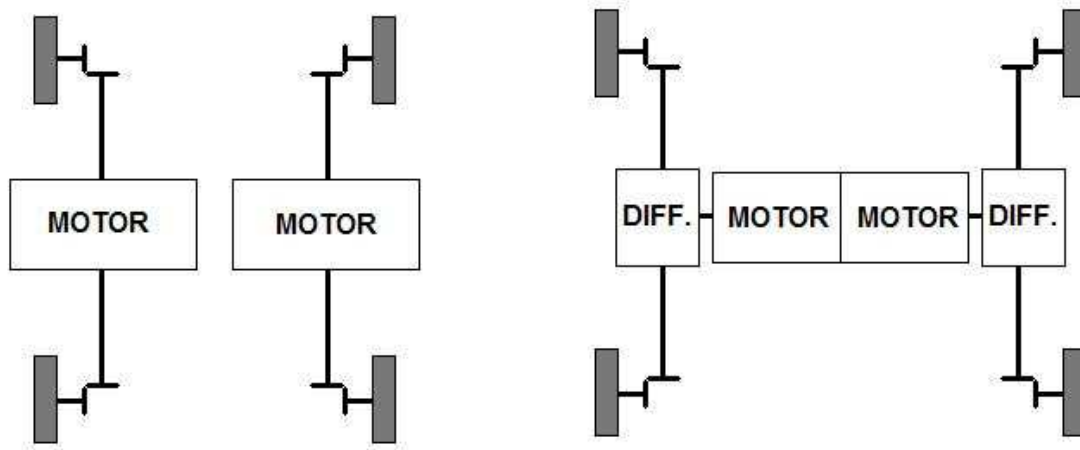
4 wielen



In de tekening hierboven zien we een methode die valt samen te vatten als "het vierwielige equivalent van de methode besproken in '2+sturing'". Opnieuw is er het nadeel dat er niet ter plaatse geroteerd kan worden. Het systeem dat eerder werd vermeld om de fout op de bocht te verkleinen is in dit geval compleet analoog alleen zou men nu in plaats van een tandwieloverbrenging voor elk van beide wielen een tandwieloverbrenging voor de hele as moeten maken. Om slip te voorkomen zou men nu op zowel

de voorste als de achterste as een differentieel moeten monteren. Het enige logische voordeel is dat een robot op vier wielen minder makkelijk omvalt dan een robot op drie wielen. Moest evenwicht dus een groot probleem worden dan zou dit een goed alternatief zijn voor de methode besproken in '2 + stuurwiel'.

Besluit: Moeilijker equivalent van '2 + stuurwiel'. Voordeel: stabiliteit

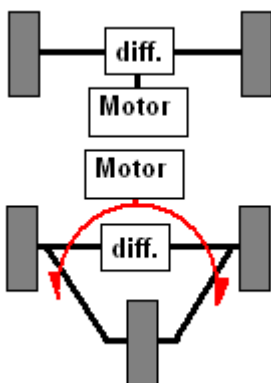


In de tekening hierboven zien we een methode die valt samen te vatten als "het vierwielige equivalent van rupsbanden". Net zoals bij de rupsbanden is het met deze configuratie mogelijk om ter plaatse te roteren. Het nadeel is net als bij de rupsbanden dat er wielen doorslippen. De ideale oplossing zou in dit geval zijn dat alle wielen kunnen draaien, maar er zijn niet genoeg motoren om dit te verwezenlijken. Het is ook mogelijk de slip te beperken door gebruik te maken van differentiëlen.

Besluit: Niet te verwezenlijken door te weinig motoren

5, 6, 7... wielen

Wanneer men naar grotere wiel aantallen gaat, hervalt men steeds in de voordien besproken methodes. Het verschil is dat er steeds meer stabiliteit gecreëerd wordt maar steeds moeilijkere constructies moeten worden gemaakt. Meer wielen wil meestal ook zeggen meer kans op slippen. Als voorbeeld wordt kort een systeem met vijf wielen besproken.



Het systeem dat op de tekening getoond wordt komt eigenlijk neer op een 2+ stuurwiel systeem, de drie voorste wielen doen dienst als stuurwiel.

Het ander geval zou zijn 4+ stuurwiel, maar dan wordt er gewoon meer slip en ingewikkeldere systemen gecreëerd. De slip kan gedeeltelijk gecompenseerd worden door gebruik te maken van differentiëlen maar dit maakt de constructie veel ingewikkelder.

Besluit: Moeilijkere equivalenten van eerder besproken systemen.

Detectiemethodes

Deze paragraaf handelt over het autonoom detecteren van ballen door de robot. Ten eerste zal men zich afvragen in welke mate dit noodzakelijk is, en of ook de kleur van de ballen belang zal hebben. Daarna worden algemeen de mogelijkheden van detectie aan de hand van sensoren besproken.

Verder bespreekt men de technieken die mogelijk zijn met de basis sensoren van Lego. Enkele concepten worden tot op een semiconcreet niveau uitgewerkt om hun voor- en nadelen te kunnen afwegen.

Nut van het detecteren van de ballen

Vanuit de probleemstelling blijkt dat het noodzakelijk is dat de robot detecteert waar de ballen zich bevinden. Het is echter niet strikt noodzakelijk om de ballen te detecteren als gebruikt gemaakt wordt van een grijpsysteem dat meerdere ballen tegelijk voortsleept of voortduwt. In het hoofdstuk over grijpsystemen worden enkele van zulke methodes besproken.

Omdat de robot op een relatief klein terrein moet werken kan het efficiënt zijn om een route te programmeren die de robot aflegt en door een V-vormige grijper de ballen te verzamelen (zoals uitgelegd bij de grijpsystemen). Hierbij is enkel een sensor nodig die zegt dat er een bal in de grijper terecht gekomen is, zodat de grijper in actie gezet kan worden.

Algemene detectie aan de hand van sensoren

Vooraleer men zich beperkt tot het gebruik van de basis sensoren van Lego, wordt eerst even onderzocht wat de mogelijkheden zijn om een machine een voorwerp te laten detecteren aan de hand van sensoren.

Men kan bestaande technieken zoeken die worden toegepast in situaties die gelijkenissen hebben met deze opdracht. Als voorbeeld denken we aan het systeem dat vaak wordt toegepast als bewakingssysteem, waarbij een lichtstraal kan onderbroken worden en zo informatie geeft aan een sensor. Als men deze techniek toepast op de probleemstelling van dit project, kan men het mogelijk maken om de aanwezigheid van een bal te detecteren tijdens het voorbijrijden van de robot.

Een andere optie zou het gebruik van een webcam kunnen zijn. Hierbij wordt eigenlijk een beetje het menselijke zicht geïmiteerd. Hierbij moet dan de juiste software gebruikt worden voor het verwerken van de beelden.

Werken met de basis sensoren van LEGO

Het basispakket van sensoren is beschreven in het hoofdstuk over sensoren. Na het testen van de sensoren bleek dat het niet mogelijk is om met één enkele sensor de ballen en hun kleur te detecteren.

Er moeten systemen bedacht worden waarin de verschillende sensoren elk een functie vervullen, en de robot moet een aangepast parcours volgen, waardoor elke bal die op het veld ligt kan gedetecteerd worden.

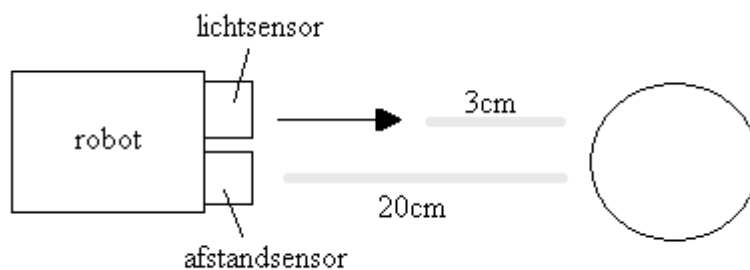
Keuze uit de sensoren en samenwerking tussen de sensoren

De afstand- en lichtsensor kunnen gebruikt worden voor de detectie van de ballen. Combinaties met de andere sensoren komen hiervoor niet zozeer in aanmerking, omdat de ballen geen geluid maken voor de geluidsensoren en ook niet voldoende tegendruk kunnen geven voor de druksensoren.

Herhaling: conclusie van de sensor testen:
afstandssensor: mogelijk te detecteren tot op 20cm afstand
lichtsensor: - voor de blauwe bal: tot op 8cm - voor de rode bal: tot op 3 cm

Men kan de afstandssensor gebruiken om de aanwezigheid van objecten in de omgeving op te sporen. Hierbij weet men nog niet met zekerheid welk object de sensor waarneemt. Het kan zijn dat er een muur gedetecteerd wordt.

Aan de hand van het navigatiesysteem kan de robot wel weten waar de muren zich bevinden. In dat geval moet hij beslissen dat dit geen doel is, en dus verder zoeken tot hij een object vindt dat wel een bal kan zijn.



Wanneer de robot in de nabijheid van het object gekomen is, kan nagegaan worden of er wel degelijk een bal ligt. Zo kan men zeker zijn dat het niet om een vergissing gaat, of dat de concurrerende robot in het gezichtsveld gepasseerd is.

Deze detectie kan gebeuren met de lichtsensor die dan ook ter bevestiging de kleur van het object aangeeft.

Het uitlezen van de lichtsensor kan gebeuren nadat de robot een grijppoging ondernomen heeft, zo kan men controleren of het grijpen succesvol geweest is. Door op dat moment een waarde uit te lezen, kan men weten of de gripper leeg is of niet.

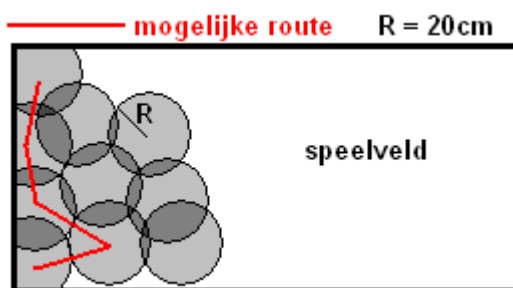
Plaatsing van de sensoren

Als men met de afstandsensor alle richtingen wil verkennen, moet deze kunnen ronddraaien. Ofwel draait de robot in zijn geheel rond over een bepaalde hoek, ofwel wordt de sensor op een ronddraaiend gedeelte gemonteerd. De afstandsensor moet zich op de juiste hoogte bevinden, zodat hij niet boven de ballen uitkomt. De lichtsensor kan zich in de grijper bevinden, of op een plaats waar hij zich vlakbij de bal bevindt als deze opgetild of vastgeklemd is.

Parcours

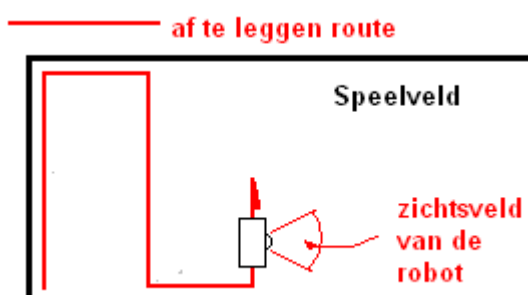
Daar het detectiesysteem afhangt van (o.a.) het parcours van de robot, is het ook afhankelijk van het type robot. Doordat het 'zicht' van de robot beperkt is tot 20cm zal het speelveld onderverdeeld moeten worden in verschillende zones die de robot kan gaan bekijken. Vanaf nu zal men het systematisch nagaan van zones op het veld 'scannen' noemen. Als men het veld scant is het aangeraden om te onthouden welke zones al nagekeken zijn en leeg zijn om geen tijd te verspillen. Ook mag de robot, als hij een bal gevonden heeft, enkel door een lege zone rijden om niet tegen andere ballen te botsten die dan kunnen weggrollen in zones waar de robot niet meer gaat scannen omdat ze leeg beschouwd worden. Hiervoor is een zeer goede navigatie nodig (hoe dit in zijn werk gaat wordt uitgelegd uit in het volgende onderdeel.)

De manier waarop het veld gescand wordt hangt af van de vorm van de robot en hoe de sensors erop gemonteerd zijn. Als de afstandsensor op een ronddraaiend gedeelte gemonteerd is, is het het voordeligst om naar een bepaalde plaats te rijden en daar een cirkel te scannen.



De route die de robot op dat moment volgt is vooral bepaald door de strategie, hij kan zo snel mogelijk naar het midden werken om ballen van het andere team te stelen, of hij kan het veilig spelen en eerst de ballen op de eigen helft verzamelen.

Als de robot niet goed ter plaatse kan roteren of traag roteert, is het voordeliger om lange rechte lijnen te scannen, de afstandssensor wordt dan op de rechterkant gemonteerd zodat de robot rechtdoor kan rijden tijdens het scannen.



Bij dit geval is het gemakkelijker om de zones die al gescand zijn te onthouden omdat enkel maar 1 coördinaat bijgehouden moet worden,

wat de navigatietechnieken aanzienlijk vergemakkelijkt.

Door de beperkte mogelijkheden van de sensors die beschikbaar zijn is de combinatie afstandssensor/lichtsensor de enige mogelijkheid die toelaat om ballen te detecteren.

De scantechnieken laten de robot toe om niet blind rond te rijden op het veld en in dit geval een bal tegen te komen op goed geluk. Aan de andere kant is het veld klein genoeg en liggen er voldoende ballen om blind rond te rijden. De belangrijkste keuze bij een robot zal er dus, op vlak van detectie, op neerkomen of er een scansysteem gekozen wordt of niet.

Navigatiesystemen

Met navigatie wordt de plaatsbepaling van de robot op het speelveld bedoeld. In het volgende onderdeel zullen methoden uitgewerkt worden om die plaatsbepaling accuraat uit te voeren.

Het nut van navigatie

Waarom moet de robot weten waar hij is op het speelveld?

- **Efficiënt zoeken van ballen**

Als de robot een scanning techniek gebruikt is het zeer handig om te weten waar geen ballen liggen zodat hij die zone niet meer moet scannen en het zoeken van ballen sneller gaat. Door goede navigatie en dus plaatsbepaling weet het programma waar de robot heeft gescand en welke dus de zones zijn waar er geen ballen meer liggen.

- **Bepalen van de beste routes om te rijden**

Als de robot een bal heeft moet hij terug naar de goot, liefst zonder ergens te blijven hangen of elke bal op het speelveld te verplaatsen door ertegen te botsen (de bal zou in een zone kunnen terechtkomen die al nagekeken is en dus zal die zone opnieuw gescand moeten worden)

Hij moet dus een degelijke route zoeken en daarvoor moet hij weten waar hij is en waar hij is geweest.

Oorzaken van fouten bij de robot

Fout op de bepaling van afstand door rotaties van de motor te tellen:

- Fout op de motor (relatieve fout is 2.5%)
- Doorslippen wielen
- Doorslippen tandwielen
- Botsingen

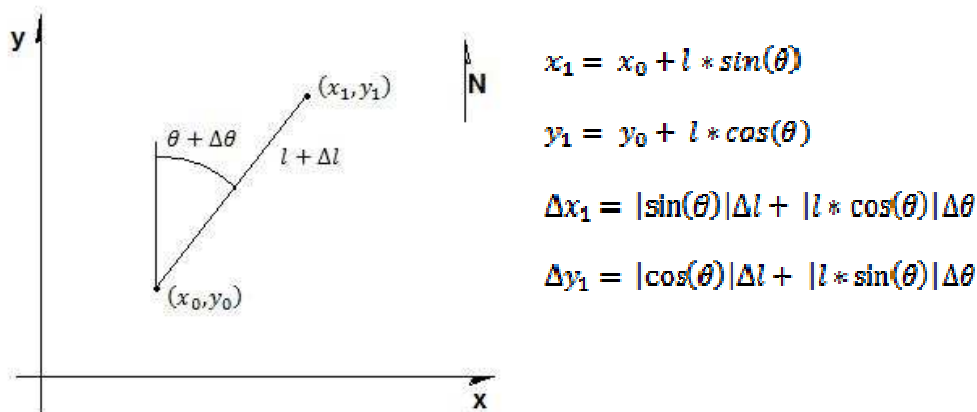
Fout op de bepaling van de richting: (te bepalen in een meting)

- Resolutie = 2°
- Fout van het toestel

Navigatie door enkel gebruik te maken van een gekend startpunt

Het spel begint in een punt dat gekend is, van daar moet de robot verder. Op het speelveld heeft de robot 3 vrijheidsgraden, een positie die uit 2 coördinaten bestaat en een richting.

Stel: het 'noorden' van de richtingssensor ligt volgens de y-as en de robot start in een punt (x_0, y_0) . Hij verplaatst zich over een lengte l in de richting θ . Op beide zit een fout:



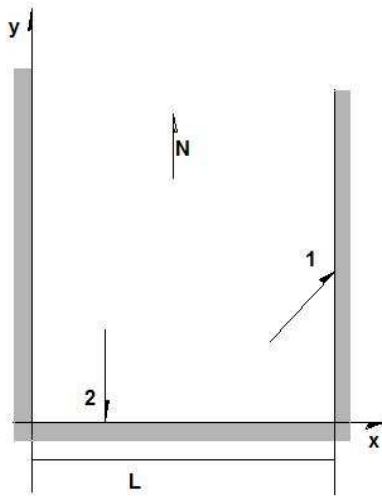
Deze fouten kunnen klein zijn, het probleem is dat als de robot vanuit (x_1, y_1) verder rijdt, dan wordt de fout:

$$\Delta x_n = \Delta x_{n-1} + |\sin(\theta)|\Delta l + |l * \cos(\theta)|\Delta \theta$$

$$\Delta y_n = \Delta y_{n-1} + |\cos(\theta)|\Delta l + |l * \sin(\theta)|\Delta \theta$$

De fout telt zich op bij elke beweging en kan zeer groot worden na een bepaalde tijd. Er moet dus een manier gevonden worden om de opgestapelde fout terug op 0 te zetten, hiervoor is er af en toe een referentiepunt nodig.

Botsing tegen een muur gebruiken als referentiepunt

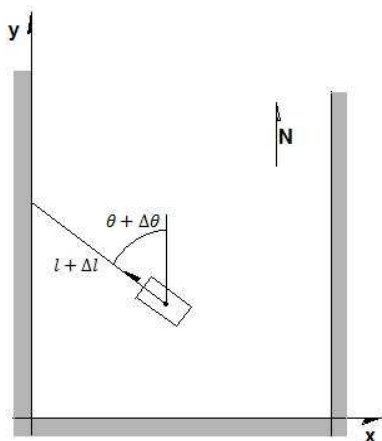


Als de robot een botsing detecteert kan hij in combinatie met de richtingssensor de fout op een van de coördinaten verkleinen.

1. De robot botst en rijdt richting noordoost
=> Hij kan de x-coördinaat instellen op L.
2. De robot botst en rijdt richting zuiden
=> Hij kan de y-coördinaat instellen op 0.

Op deze manier kan de fout bij elke botsing tegen een muur verkleind worden, een groot probleem ontstaat bij botsing tegen een andere robot.

Gebruik van de afstand tot een muur als referentiepunt



In combinatie met de richtingssensor kan een van de coördinaten bepaald worden als de afstand tot een muur gekend is. In de figuur:

$$x = l * \sin(\theta)$$

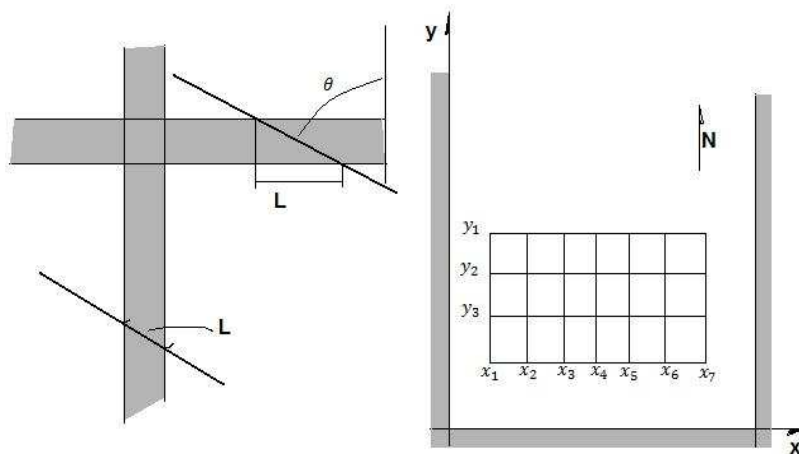
$$\Delta x = |\sin(\theta)| \Delta l + |l * \cos(\theta)| \Delta \theta$$

Door de onnauwkeurigheid van de afstandssensor is deze fout redelijk groot, maar als de fouten lang genoeg opgestapeld zijn zal deze techniek verbetering brengen.

In het programma zou een grove schatting van de fout kunnen gemaakt worden door te kijken hoe lang de robot al rond rijdt zonder te botsen. Als deze fout te groot wordt kan de robot deze correctie uitvoeren.

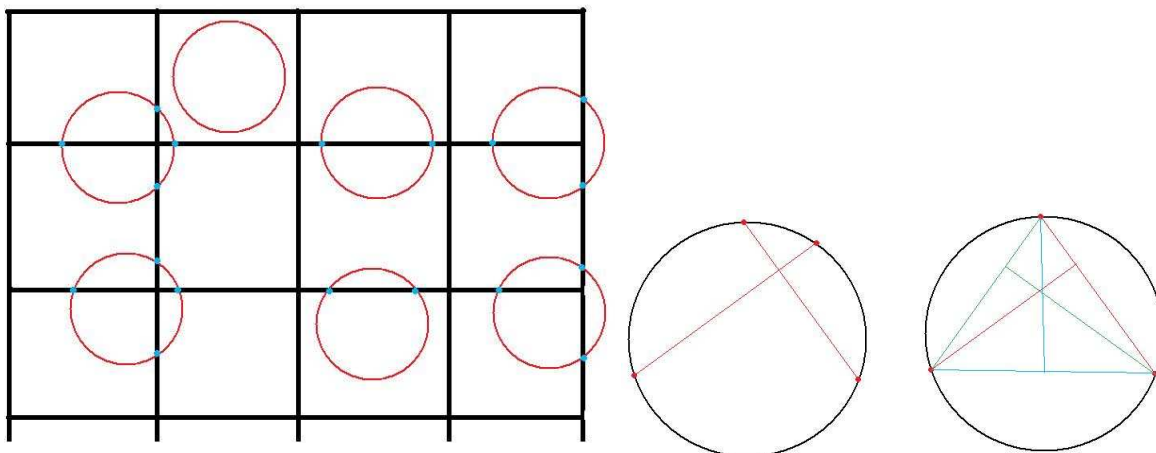
Er kunnen problemen ontstaan als er een bal in de weg ligt of de andere robot in de weg staat.

Gebruik van de lijnen op het speelveld als referentiepunt



Elke keer dat de robot een lijn tegenkomt kan hij - door de tijd te bepalen die hij nodig heeft om er over te rijden, de richtingssensor en de coördinaat waar hij is om te bepalen welke lijn het is - één van de coördinaten aanpassen.

Het is ook mogelijk de coördinaat te corrigeren door af en toe eens een cirkel te scannen met de lichtsensor. Als de snijpunten met de lijnen bepaal zijn kan de positie van lijnen ten opzichte van de robot bepaald worden om op die manier de coördinaat te corrigeren.



Als er 4 snijpunten zijn kunnen de lijnen eenduidig bepaald worden, als er 3 snijpunten zijn kunnen de lijnen op 3 verschillende manieren liggen, hier is de richtingssensor nodig om te bepalen welke correct zijn.

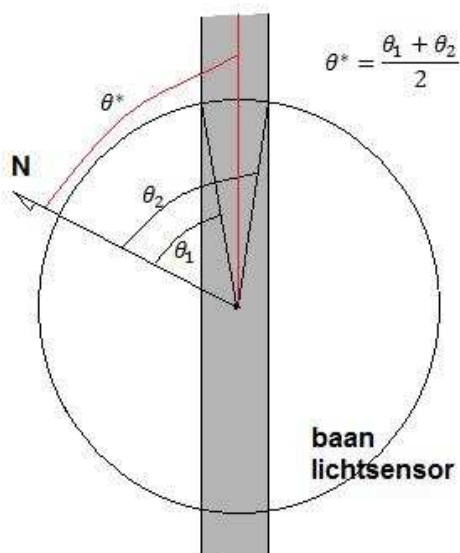
Voor deze methode moet de robot af en toe een rondje draaien en dit is tijdverlies, daarom is deze methode niet echt optimaal. Ze kan wel gebruikt worden om te weten of er ergens een kruispunt is als de robot de lijnen volgt.

Navigatie door de lijnen op het speelveld te volgen.

Omdat de balletjes meestal in de buurt van of op de kruising van twee lijnen zullen liggen kan de robot de lijnen volgen om alle kruisingen af te gaan. In dit geval wordt de lichtsensor onderaan de robot gehangen, gericht naar beneden.

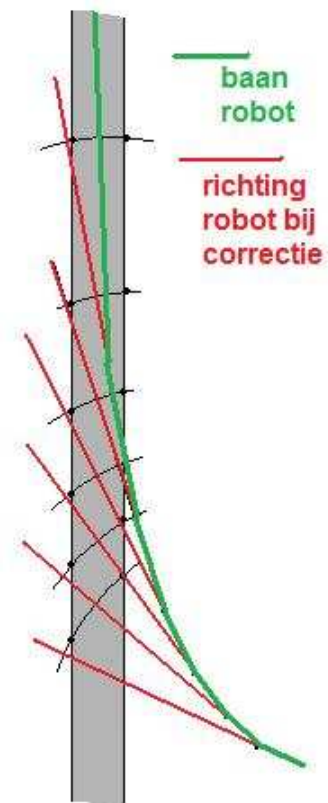
Het volgen van de lijnen zonder richtingssensor

Met behulp van de lichtsensor kunnen de lijnen gevolgd worden. Als de robot een lijn verlaat moet hij draaien om de lijn terug te vinden.



Het gemakkelijkste geval doet zich voor als het draaipunt van de robot op de lijn ligt. Wanneer de lichtsensor dan de lijn verlaat, moet de robot draaien tot hij op de lijn is. Dan doordraaien tot hij weer van de lijn is, de richting van de lijn ligt tussen de twee.

Als het draaipunt van de robot niet op de lijn start kan door een aantal keer de correctie toe te passen op de lijn gereden worden.



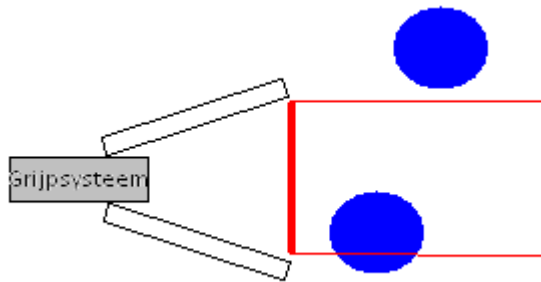
Met de richtingssensor is het veel gemakkelijker om een lijn te volgen, de richtingssensor geeft de richting van de lijn aan en de lichtsensor bevestigt of de robot al dan niet op de lijn zit, dit vermindert het aantal correcties dat moet uitgevoerd worden aanzienlijk en versnelt dus het rijden van de robot.

Besluit: Over het algemeen is het volgen van de lijnen trager dan de andere methode en omslachtiger. Als het speelveld veel groter zou zijn zou het de moeite lonen om de lijnen te volgen, maar op ons speelveld is het sneller om een willekeurig punt te nemen, er naartoe te rijden en daar eens rond te kijken op zoek naar balletjes. Daarnaast is het ook niet 100% zeker dat een balletje op het kruispunt ligt en moet er sowieso rond gekeken worden in elk kruispunt.

Grijpsystemen

In dit stuk worden enkele mogelijke grijpsystemen besproken die kunnen worden aangewend voor het grijpen van ballen. Tenzij anders vermeld, kunnen deze grijpsystemen op alle vormen van robots geplaatst worden.

Vorklift



Het idee van de vorklift volgt uit het idee dat de grijpkansen van de robot zo hoog mogelijk moeten gemaakt worden. Een grote vereiste in de ontwikkeling van de robot is te denken in functie van nauwkeurigheid en efficiëntie. Een van de voorstellen om dit te verwezenlijken, was het aanbrengen van een soort V-vormige vork, welke tegen het speelveld of alleszins ter hoogte van de ballen zou worden gemonteerd. Indien de robot een bal zou detecteren, zou het positioneren voor de

bal, alvorens te grijpen, minder nauwkeurig mogen gebeuren, omdat de vork bij het vooruit rijden de bal toch naar het midden van het grijpsysteem zou leiden.

In principe kan deze techniek bij verschillende grijpsystemen als hulpmiddel worden toegepast. De link met een vorklift is echter snel gelegd. Hier krijgt de vork naast het leiden van de ballen ook de functie van 'grijper'. Het is wel belangrijk te vermelden dat deze techniek zijn nut heeft omdat de ballen afgevlakt zijn aan de onderkant. Indien dit niet zo zou zijn, zou het positioneren van de robot nog steeds grote nauwkeurigheid vragen, omdat de ballen dan volledig binnen de rode balk moeten liggen. Ze zouden namelijk weggrollen bij het raken van de vork. Een ander voordeel is dat er slechts één beweging nodig is om de ballen op te rapen, namelijk het op en neer bewegen van de vork. Het nadeel van de vorklift is echter dat het geen ideale oplossing is om bolvormige objecten op te rapen, afgevlakt of niet, omdat ze er vrij makkelijk af vallen. Er zou dus moeten gedacht worden aan een manier om de ballen op de vorklift te klemmen bij het vervoeren.

Besluit:

- Grotere grijpkansen door V-vormige vork
- Positioneren hoeft minder nauwkeurig
- Slechts 1 beweging: op en neer gaan van de vork
- Niet ideaal voor bolvormige objecten
- Systeem nodig om ballen te klemmen bij het vervoeren

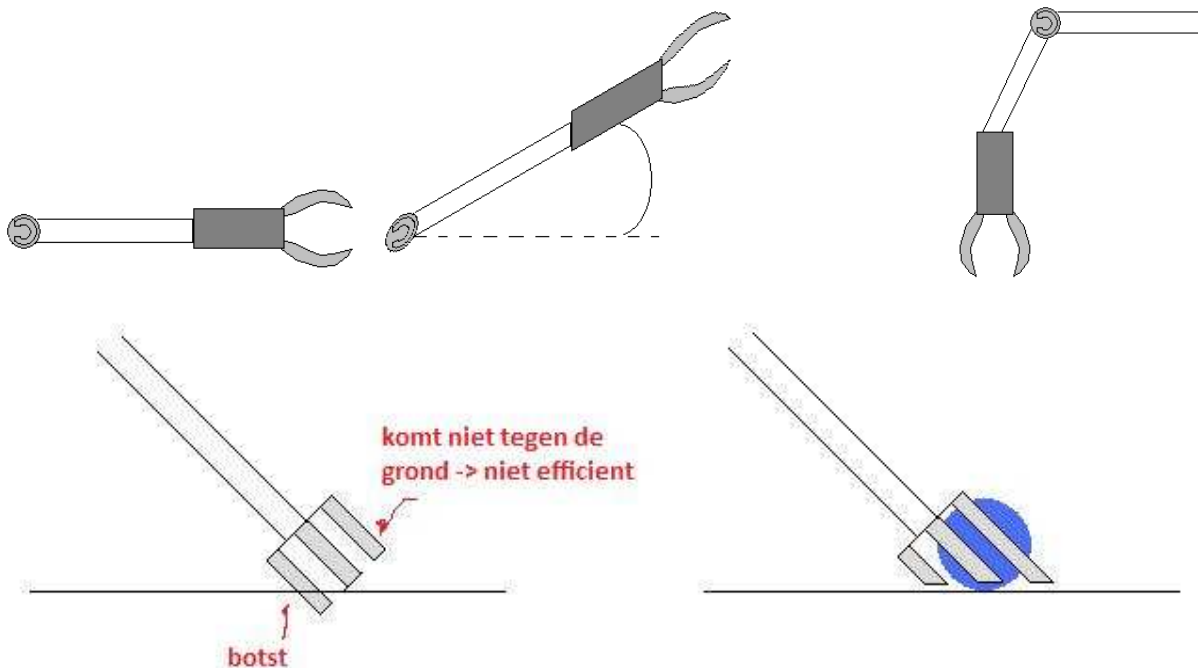
Grijper

Het klassieke voorbeeld van een grijpsysteem is de grijper. Deze bestaan in verschillende maten en soorten, zoals te zien op de onderstaande afbeeldingen.



Het grote voordeel van dit systeem is dat bolvormige objecten stevig geklemd kunnen worden en makkelijker gegrepen kunnen worden door de ronde vorm van de grijper en meerbepaald de scharen van de grijper. Het is ook mogelijk om meerdere scharen of scharenparen toe te voegen aan de grijper om bijvoorbeeld een grotere klemkracht te verkrijgen of om de ontsnappingskans van de ballen te verkleinen.

Nog een groot voordeel is dat men de grijper op verschillende manieren kan monteren en gebruiken. Men heeft allereerst al de keuze om de grijper verticaal, horizontaal of diagonaal te gebruiken. Verder kan men de grijper onder andere monteren op een bewegende arm of een niet bewegende arm. De bewegende arm zou verticaal, horizontaal of diagonaal kunnen verplaatst worden. In het geval van een diagonaal bewegende grijper zou men de grijper wel moeten aanpassen opdat hij de bal volledig binnen zijn scharen zou kunnen nemen zonder tegen de grond te botsen.



In het geval van een bewegende arm kan men de beweging van de arm en de knijpbeweging van de grijper door eenzelfde motor laten uitvoeren door middel van een differentieelsysteem, wat inhoudt dat één motor twee bewegingen aandrijft. Hierbij is het belangrijk dat de gewenste bewegingen goed worden uitgewerkt en begrepen, want bij het gebruik van dit systeem is het belangrijk dat men weet welke bewegingen op hetzelfde moment mogelijk zijn en welke op die momenten met de minste moeite kunnen worden uitgevoerd. Dit wordt verder uitvoerig uitgelegd.

In het geval van een niet-bewegende arm heeft men geen differentieelsysteem nodig. Men moet er dan simpelweg voor zorgen dat de robot met open grijper op de juiste plaats (plaats van de bal) terecht komt, de grijper gesloten wordt en de robot weer verder rijdt.

Besluit:

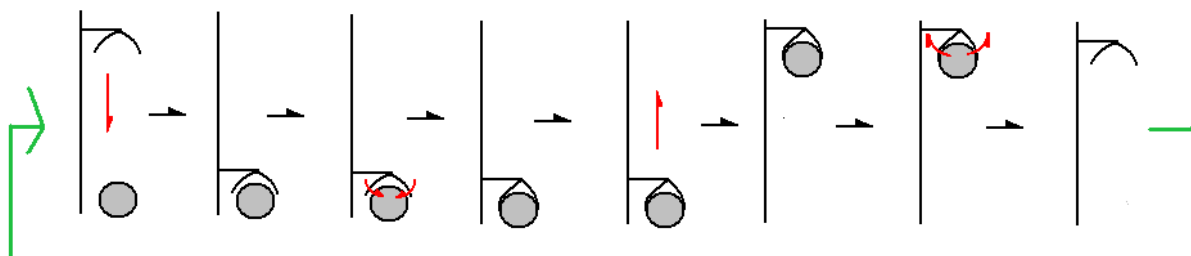
- Grote klemkracht
- Gebruik differentieelsysteem mogelijk
- Horizontaal, verticaal en diagonaal gebruik
- Verschillende montage mogelijkheden

Differentieel-grijper systemen

Een grondige bespreking van het gebruik van een differentieelsysteem voor een grijper is belangrijk alvorens men kan beslissen dit systeem toe te passen. In dit geval wordt het uitgewerkt voor een verticale grijper, de redenering voor een horizontale grijper verloopt analoog buiten het effect van de inspelende zwaartekracht. Bij toepassingen zou het horizontale equivalent hiervan dan kunnen worden nagebootst met behulp van bijvoorbeeld een veer.

In dit deel wordt er dieper ingegaan op twee specifieke methodes om enerzijds het openen en sluiten en anderzijds de verticale beweging van de grijper te combineren met behulp van de differentieel. Door de differentieel kunnen beide bewegingen dus aangestuurd worden door een en dezelfde motor.

De bedoeling van beide methodes is om volgende beweging vlot te laten verlopen:



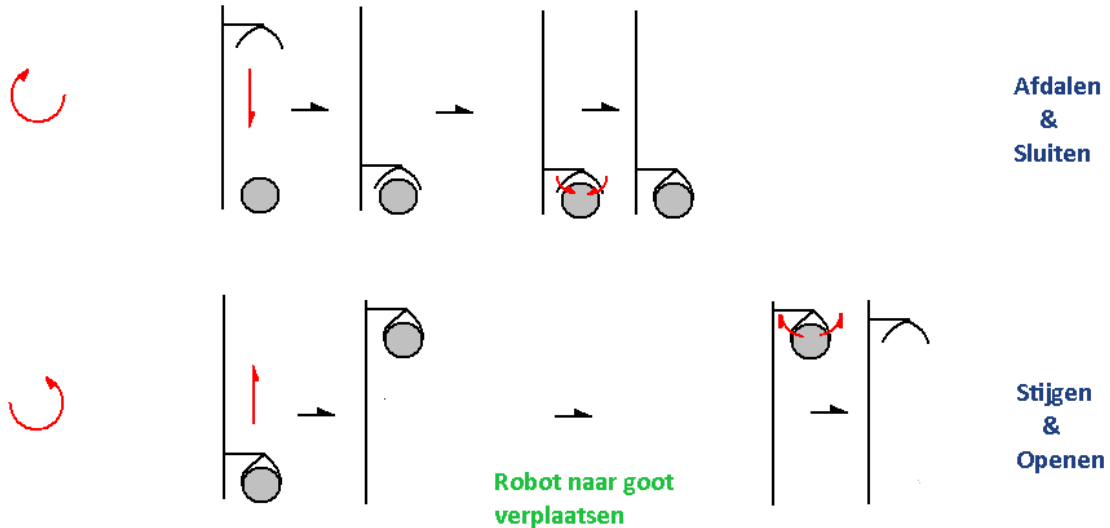
Het verschil bestaat erin wanneer de motor van draairichting moet veranderen en welke belastingscombinaties wanneer gecreëerd moeten worden.

Optie 1

De eerste methode ziet eruit als volgt:

MOTORRICHTING

GRIJPER



Omdat het de bedoeling is de gripper eerst te doen dalen alvorens hij sluit (1) en in de omgekeerde beweging (omdraaien van de motor) de gripper eerst te doen stijgen alvorens hij terug opent (2) moeten de volgende belastingsverhoudingen voor de gripper gerealiseerd worden:

Belasting:

1. sluiten > afdalen
2. openen > stijgen

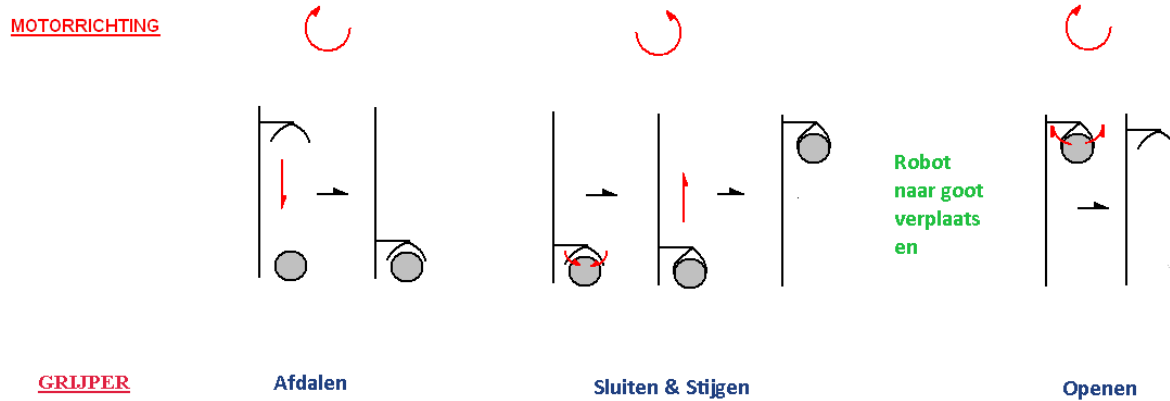
De eerste belastingsverhouding is makkelijk te realiseren: door de massa van de verticaal opgehangen gripper is de belasting op de neerwaartse beweging zeer klein, het sluiten zal dus al snel een grotere krachtvereiste hebben dan het dalen. De tweede belastingsverhouding daarentegen is moeilijker te verwezenlijken. Door de massa van de gripper is er al een grote kracht vereist om deze omhoog te trekken. Om de tweede belastingsverhouding te realiseren zou de vereiste belasting voor het openen van de gripper dus nog groter moeten zijn dan de al zeer grote belasting die gepaard gaat met het omhoog hijsen van de gripper.

Om hiervoor te zorgen zou men de gripper heel hard moeten dichtspannen en vanaf dan komt men in een kleine evenwichtszone. Het probleem is vrij eenvoudig: vanaf dat men de gripper zodanig hard gaat dichtspannen opdat de tweede belastingsverhouding verwezenlijkt wordt, dreigt het gevaar dat de belasting voor het dichtgaan zodanig klein wordt dat deze nog kleiner wordt dan deze voor het afdalen. Hierdoor zou belastingsverhouding 1 niet meer plaatsvinden waardoor het hele systeem niet meer werkt, de gripper zou namelijk sluiten voor hij afdaald. De moeilijkheid bevindt zich er dus in om in deze evenwichtszone te geraken, in het geval deze al zou bestaan.

Conclusie: Efficiënte methode maar probleem van kleine, misschien onbestaande belastingevenwichtszone.

Optie 2

De tweede methode ziet eruit als volgt:



In dit geval is het de bedoeling de gripper eerst te openen alvorens hem te laten dalen (1) en in de omgekeerde beweging (omdraaien van de motor) de gripper eerst te sluiten alvorens hem terug te laten stijgen(2). Na deze cyclus moet de robot eerst naar de goot verplaatst worden voordat de gripper opengaat. Na het openen wordt de cyclus weer stil gelegd en loopt dan weer verder bij detectie van een nieuwe bal. De volgende belastingsverhoudingen moeten gerealiseerd worden:

Belasting:

1. dalen > openen
2. stijgen > sluiten

Men ziet onmiddellijk dat de eerste belastingsverhouding niet zo vanzelfsprekend is, het dalen van de gripper vereist een zeer kleine kracht en het is dus zeer moeilijk om te zorgen dat het openen nog minder kracht vereist. Een manier om hiervoor te zorgen is het openspannen van de gripper met behulp van elastieken, maar dan wordt het weer moeilijk om ervoor te zorgen dat het sluiten niet meer kracht gaat vereisen dan het stijgen waardoor belastingsverhouding 2 niet meer zou plaatsvinden. Op deze manier komt men dus opnieuw in een vergelijkbare evenwichtszone als in optie 1.

In dit geval is er wel een alternatieve oplossing mogelijk. De gripper moet namelijk openen voordat hij aan de afdaling begint. Eens de gripper open is, is het net de bedoeling dat de afdaling minder kracht vereist, hetgeen ook het geval is, want de gripper is volledig open en een verdere opening zou dus een 'oneindige' kracht vereisen. Men moet er dus voor zorgen dat de gripper bovenaan 'vast' gehouden wordt zolang hij niet geopend is. Een veer biedt hiervoor geen oplossing omdat de kracht van een veer net evenredig is met de afstand terwijl er in dit geval wordt gezocht naar een systeem dat omgekeerd evenredig is met de afstand. Indien men hier dus een methode voor vindt, biedt optie 2 een oplossing zonder dat men zich moet bezighouden met de moeilijke evenwichtzone.

Conclusie: Zelfde probleem met belastingevenwicht als in optie 1 maar alternatief mogelijk.

Extra mogelijkheden

Er zijn buiten de twee algemeen besproken concepten nog andere mogelijkheden zoals zuignappen, een werparm of grijpers die meerdere ballen tegelijk vervoeren. Omdat de conclusie van deze besprekingen meestal zeer negatief is en de systemen daardoor onbruikbaar zijn heeft men besloten ze niet tot de inhoud van dit verslag te laten behoren. De lezer kan ze wel terugvinden in het deel 'extra grijpsystemen' in bijlage 2.

Opslagsystemen

In deze paragraaf wordt het vervoer van meerdere ballen tegelijk besproken. Omdat het grijpen van meerdere ballen tegelijk al eerder besproken is, gaat men hier uit van het specifieke geval dat de grijper 1 bal per keer opneemt, en dat men het aantal heen-en-weer-ritten van en naar de goot wil beperken.

Het beperken van het aantal heen-en-weer-ritten lijkt in elk geval de efficiëntie ten goede te komen. Toch kan het niet als een absoluut streefdoel beschouwd worden om alle ballen te blijven bijhouden en slechts 1 terugrit naar de goot te plannen. Het heen en weer rijden kan ook voordelen hebben: de robot kan zijn positie terugvinden (voor de navigatie).

Als het reservoir op een verkeerde plaats de ballen zou verliezen, is het beter dat het aantal ballen beperkt is gebleven. Daarom moet men een evenwicht vinden waarbij een verantwoord aantal bijgehouden ballen en een verantwoord aantal heen-en-weer-ritten worden gemaakt. Hiervoor moet men rekening houden met de snelheid van de robot en met zijn oriëntatievermogen.

Wanneer men een idee heeft van het aantal ballen dat zal vervoerd worden, resten er nog de volgende uitdagingen:

- Het reservoir moet licht zijn en toch een redelijke hoeveelheid ballen kunnen stapelen
- De ballen moeten vanuit de grijper in de opslagplaats kunnen gelegd worden.
- Het reservoir moet langs de onderkant gesloten zijn en de ballen moeten over de opstaande rand van het veld kunnen geloosd worden
- Het reservoir moet zich kunnen openen op het juiste moment
- Het reservoir moet zich kunnen openen zonder een motor te gebruiken

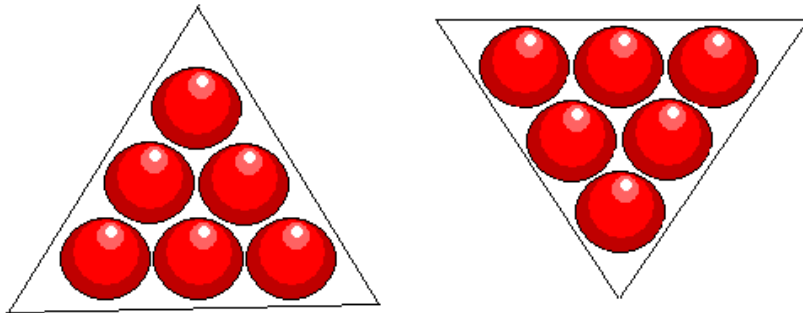
(de 3 beschikbare motoren worden in de meeste gevallen al gebruikt voor het rijden en het oprapen van de ballen)

De vormgeving en constructie van het reservoir

De vorm

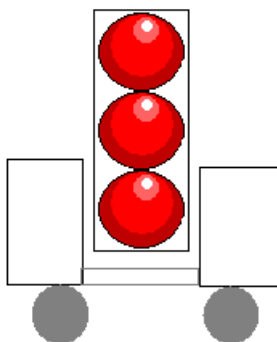
De ballen zijn rond en hebben een diameter van 7,5 cm. Het kan nuttig zijn om een stapelmanier te zoeken waarbij het ruimteverlies tussen de ballen beperkt blijft.

Voorbeelden:

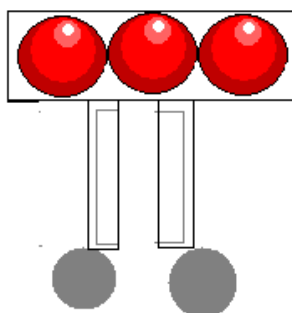


Andere mogelijkheden zijn:

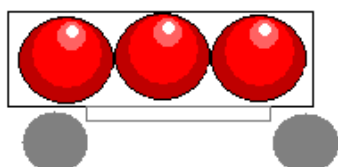
in de hoogte werken



- ✓ klein oppervlakte grondvlak -> voordelig voor het manoeuvreren
- Opletten voor de stabiliteit

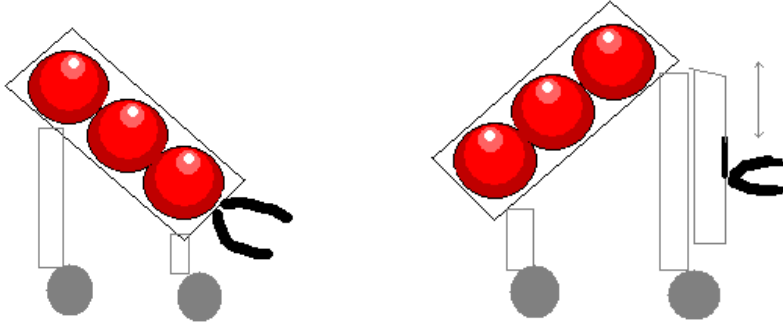
horizontaal

- ✓ Nog mogelijk gemakkelijk te manoeuvreren
- De ballen moeten hoog getild worden



- ✓ De ballen moeten niet hoog getild worden
- mogelijke problemen bij het manoeuvreren

Diagonaal: Voordeel: De diagonale zijde is de langste zijde -> meer stapelplaats



Constructie

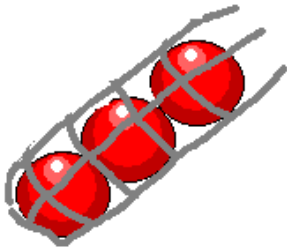
Mogelijkheden:

Volle wanden -> De ballen kunnen er niet uitvallen

De constructie moet wel licht blijven, bv. stof of gaas rond een stevig geraamte



Een geraamte waartussen de ballen gekneld zitten -> bv. uit lego of sterke ijzerdraad



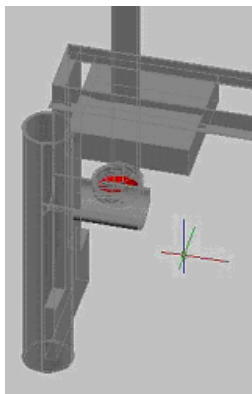
Kippendraad: dit is stevig, plooibaar, de gaten zijn klein genoeg zodat de ballen er niet uitvallen

De ballen in het opslagsysteem leggen

Men moet ervoor zorgen dat de ballen vanuit het grijpsysteem in het reservoir terechtkomen, en dit best zonder terug te botsen of ernaast te vallen.

Mogelijkheden om de overgang van grijpsysteem naar reservoir te overbruggen:

een gootje of een teut die naar het reservoir leidt



Voorbeeld van een gootje

Een kliksysteem dat opgeduwd kan worden door de motor van de gripper, als de bal erin zit, klemt het terug toe.

Het reservoir boven de goot openen zonder motor

Naargelang de vorm van het reservoir en de manier waarop de ballen naar binnen komen, zijn er meerdere mogelijkheden, o.a.:

het reservoir openen met een mechanische klep

het reservoir is al open: de ballen naar achter duwen volgens het principe van de tandpasta tube

Mechanische klep: moet in werking treden bij het botsen tegen de opstaande rand van het veld: hiervoor moet men een S-vormige klep maken die de onderkant van het reservoir doet openen bij het botsen tegen de rand.

Reservoir met opening: de gripper moet met genoeg kracht tegen de vorige ballen duwen, waardoor ze naar achter rollen en in de goot vallen. Dit moet op het juiste moment gebeuren, met de druksensor kan de rand van het veld gedetecteerd worden, en hierop kan het programma commando geven aan de gripper.

Mogelijke combinaties

Op basis van voor- en nadelen van de besproken voortbeweging-, detectie- en grijpsystemen kan men al dan niet systemen combineren om aldus een concept voor een robot te bekomen. Het voortbewegingsysteem beïnvloedt de doeltreffendheid van zowel het detectiesysteem als het grijpsysteem. Detectiesysteem beïnvloedt het grijpsysteem echter niet en omgekeerd. De volgende tabel toont ons welk wielsysteem met welke detectie- en grijpsystemen gecombineerd kan worden.

	Cirkelscansysteem	Rechtlijnsysteem	Verklift	Grijper	Zuignap	Werfarm
Rupsbanden	X	✓	✓	X	✓	X
2+steunpunt	✓	✓	✓	✓	✓	✓
2+stuurwiel	X	✓	✓	X	✓	X
4	X	✓	✓	X	✓	X
Kraanmodel	X	✓	X	✓	X	X
	Grijpsystemen					
	Detectie/navigatie					

- Voor elke combinatie kan een opslagsysteem uitgedacht worden
- ▲ De grijper wordt namelijk op een op en neer bewegende arm gemonteerd, het vorkstelsel dient tegen het speelveld te worden gemonteerd
 - De grote nauwkeurigheid van beweging verklaart de vele combinaties
 - △ Het cirkelscansysteem vereist nauwkeurige rotatiemogelijkheid

Met het kraanmodel wordt een concept op zich bedoeld, niet echt een wielsysteem, maar bij het maken van de combinaties kan het wel als dusdanig worden beschouwd. Het bestaat uit twee torens, welke op de randen van het speelveld rijden en overbrugd worden. Op de overbrugging rijdt een karretje met een toren op. Via deze toren kan een arm met grijper afdalen tot tegen het speelveld. Dit concept is gebaseerd op rolbruggen, welke men in veel magazijnen aantreft.

Dit idee heeft zeer veel zeer praktische voordelen in vergelijking met de klassieke los bewegende robot. Voor redenen die in het volgende hoofdstuk worden besproken, heeft men hier gekozen voor het ontwerpen van een dergelijk kraanmodel.

Argumentatie keuze kraanmodel

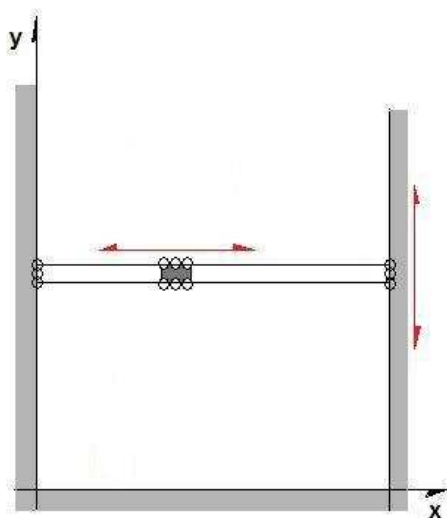
In dit deel wordt aangetoond waarom het team van ontwerpers heeft gekozen om een kraanmodel te ontwikkelen. Bij het maken van deze keuze heeft men zich gebaseerd op de voorgaande conceptbesprekingen. Elk onderdeel hiervan zal hier bestudeerd worden in het geval van een kraanmodel. Op deze manier zullen de voordelen van de kraan ten opzichte van een los bewegende robot worden aangetoond. Omdat deze kraan een meer uitgebreide constructie is dan de meeste los bewegende robots, ontstaan er grote bijkomende nadelen. Deze zullen vervolgens afgewogen worden ten opzichte van de voordelen en daarna zal, ondanks deze extra moeilijkheden, aangetoond worden waarom het team vond dat een kraansysteem de meest efficiënte keuze was.

Pro

Navigatie

Als men de ideeën uit het stuk over navigatie toepast op het kraanmodel vindt men enkele duidelijke voordelen terug:

In tegenstelling tot een los bewegende robot wordt de beweging gesplitst in twee bewegingen volgens de x en y richting, hierdoor heeft men de richtingssensor niet meer nodig. Dit verkleint de fout op de plaatsbepaling en bovendien is er extra ruimte om een andere sensor te gebruiken.



Het karretje met de grijper dat in de x-richting beweegt kan op elk moment de fout op de x-coördinaat corrigeren door botsing. De enige fout die dan nog overblijft, is die op de y-coördinaat. Als we geen opslagsysteem gebruiken kan deze gecorrigeerd worden door botsing als we een bal in de goot gaan leggen.

Op gebied van navigatie is de kraan dus een ideaal systeem.

Programmatie

Een kraan versimpelt het programmeren aanzienlijk in vergelijking met een vrij rondrijdende robot. Er is namelijk een vrijheidsgraad minder, de rotatie in het speelveld, waarmee men rekening moet houden. Hierop konden en zouden fouten optreden.

Een belangrijker voordeel aan het wegvallen van deze vrijheidsgraad is misschien niet de programmatie, maar wel de kalibratie van de waardes na programmatie.

Bovendien verhindert de constructie dat de verscheidene sensoren een muur kunnen detecteren, zodat geen algoritme geschreven dient te worden om deze te kunnen onderscheiden van een bal.

De opstelling van de sensoren, met name de combinatie van laser en lichtsensor, zorgt ervoor dat er geen zoekalgoritme, al is het maar willekeurig rondrijden, geschreven moet worden, of dat de locaties van eerder gevonden ballen moeten bijgehouden worden in dit algoritme.

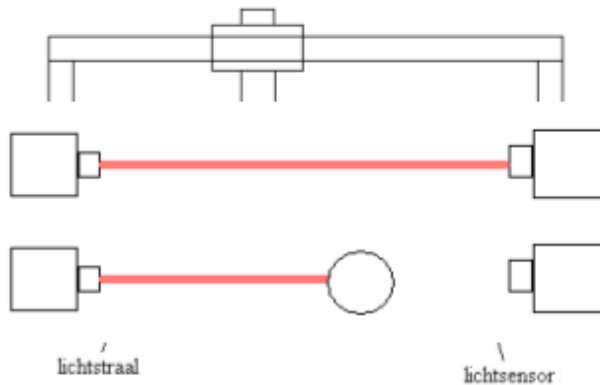
Ongetwijfeld is het grootste voordeel dat de eenvoud achter het idee van de kraan resulteert in eenvoud van de code. Niet zozeer de eenvoud om te programmeren is belangrijk, men had immers tijd genoeg om na te denken over de structuur van de code en deze ook uit te voeren in de zes weken van het project, maar door deze eenvoud is het resultaat een relatief korte code, een vijfhonderdtal lijnen, wat op zijn beurt resulteert in een sterk vereenvoudigde debugging. Het is regelmatig het geval dat het niet geweten is waar precies een fout optreedt in het programma en wat de oorzaak hiervan is. Het is triviaal dat in een kortere, compactere code dit probleem makkelijker insluitbaar en localiseerbaar is.

Immers, hetgeen het meeste tijd kost bij programmeren is fouten zoeken en deze oplossen.

Detectie

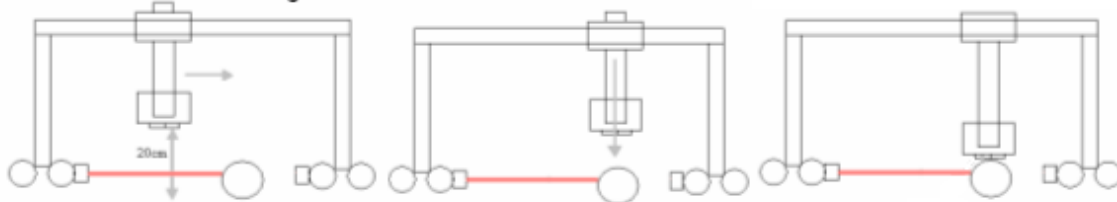
Een van de grootste voordelen van het kraanontwerp is dat, doordat er twee steunpunten zijn aan weerskanten van het speelveld, men een detectietechniek kan toepassen die gebaseerd is op een bewakingssysteem:

Als er tussen de twee delen van het onderstel een laser schijnt op een lichtsensor kan men alles detecteren wat zich daartussen bevindt. Als de kraan met het onderstel rijdt tot de lichtstraal onderbroken wordt weet het programma dat er ergens een balletje tussen de laser en de lichtsensor ligt. Eenmaal de lijn, waar het balletje ligt, gekend is, kan deze afgegaan worden met een afstands sensor die naar beneden gericht is om de exacte positie te vinden. Daarna is het slechts een kwestie van de grijper in werking te zetten en het balletje op te rapen.

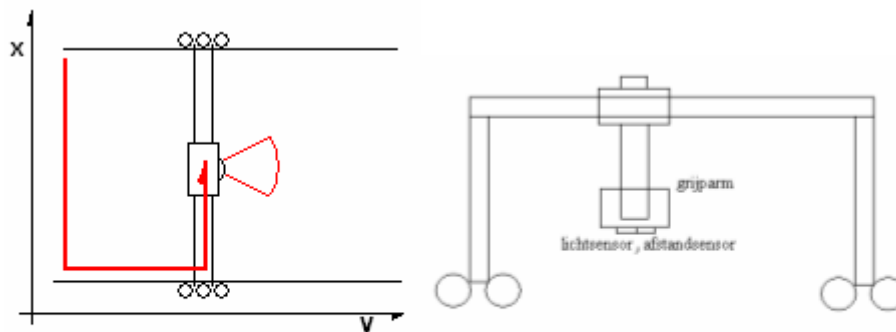


Als de lichtstraal sterk genoeg is geeft de lichtsensor een hoge waarde weer. Als er wel een voorwerp ligt, is de waarde een stuk lager, waardoor men met zekerheid een bepaalde drempel kan kiezen.

schematische weergave van de acties die de kraan uitvoert om de bal te detecteren



Als het lasersysteem niet zou werken kan men op het kraanontwerp gemakkelijk het scansysteem toepassen dat in detectietechnieken besproken werd. De afstandsensor wordt gemonteerd op de achterkant van de grijper, naar voor gericht:



Met deze configuratie is het eenvoudig om de rechte lijn scantechniek toe te passen. We rijden 20 centimeter volgens de y-richting en scannen dan de breedte af in de x-richting. Als we een bal detecteren kennen we de locatie van de bal en kunnen we hem pakken met de grijper.

Voortbeweging

Binnen de bespreking van de voortbewegingsystemen valt er een moeilijkheid op die gemeenschappelijk is voor alle besproken systemen namelijk rotaties. In alle besproken systemen was het de bedoeling de fout op de rotatie te verkleinen door onder andere slip te verminderen en het wiel aantal te veranderen. Wanneer men opteert voor een kraanmodel zijn er geen rotaties vereist dus vallen al deze problemen weg. Men moet zich geen zorgen meer maken over de berekening van de plaatsverandering en de fout hierop.

Contra

Het veld is 130cm breed, deze afstand moet dus overbrugd worden. Met Lego is dit zeer moeilijk: men zou een grote vakwerkconstructie ¹ kunnen maken, maar dan zou het geheel redelijk zwaar zijn. Er wordt voor aluminium L-profielen gekozen om die overspanning te maken. Ze buigen minder door en zijn lichter.

Een tweede moeilijkheid is dat de wielen aan weerskanten van het veld even snel moeten bewegen, anders trekt heel de constructie scheef.

Een derde probleem is het evenwicht van het geheel. Omdat de constructie redelijk hoog is komt het massamiddelpunt ook hoog te liggen. Als er dan te snel wordt opgetrokken zou het geheel gemakkelijk kunnen omvallen door het creëren van een groot moment.

Omdat de afstand van de toren tot de NXT module groot kan worden moeten er langere kabels gebruikt worden welke niet in het basispakket zitten. Die lange kabels kunnen in de weg komen te liggen als de toren dan wel dicht bij de NXT module staat.

Omdat de kraan een uitgebreide constructie is, zal het aanzienlijk langer duren om het geheel uit te werken en te bouwen.

Samenvatting

PRO

- **NAVIGATIE:** fout op de richting valt weg.
- **SENSOREN:** geen richtingssensor => poort vrij voor andere sensoren.
- **WIELEN:** de wielen moeten enkel rechtdoor kunnen rijden.
- **DETECTIE:** Lasersysteem versnelt het zoeken naar ballen aanzienlijk.
- **PROGRAMMATIE:** Doordat de beweging gesplitst wordt in componenten is het programma eenvoudiger.

CONTRA

- **CONSTRUCTIE:** Omdat de kraan zo groot is, is de constructie ervan moeilijker.
- **CONSTRUCTIE:** er moet een overspanning van 130cm overbrugd worden
- **STABILITEIT:** de robot kan gemakkelijker kantelen
- **KABELS:** Omdat de kraan zo groot is hebben we lange kabels nodig.

¹ Vakwerken: zie bijlage 1.

Ontwerp en Bouw

In dit voorlaatste hoofdstuk van het verslag wordt de ontwikkeling en het ontwerp van de uiteindelijke robot besproken. De tegengekomen problemen zullen aangehaald worden en, indien deze aanwezig waren, zullen ook de oplossingen uitvoerig besproken worden. Achtereenvolgens bespreekt men eerst de bouw van alle onderdelen, met daarin besprekingen over de stabiliteit, het scheefftrekken van de brug, de bouw van de steunkarretjes en de ontwikkeling en verandering van de grijper en het bijhorende differentieelsysteem. Vervolgens is er een korte bespreking van de moeilijkheden bij de ontwikkelingen waarbij het toegevoegd materiaal aan bod komt, namelijk laserdetectiemethode en het vinden van een efficiënt verschuivingsysteem van de kabels. Tot slot wordt het besturingsprogramma uitgelegd aan de hand van een flow-chart, dit is een schematische voorstelling van het programma, en enkele onderdelen van de code. De volledige code vindt de lezer terug in bijlage.

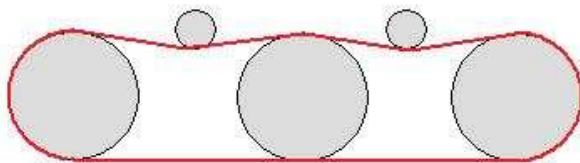
Bouw van verschillende onderdelen

Poten:

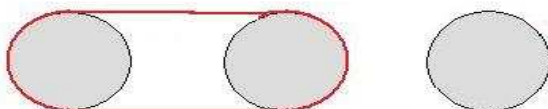
De poten van de robot rijden over de houten balken aan de zijkanten van het speelveld. Ze moeten stevig zijn omdat het volledige gewicht van de robot hierop rust. Ze zijn opgebouwd uit drie delen: de wielen, de motoren, en de steun voor de overbrugging. Hierna volgt een gedetailleerde bespreking van elk deel apart.

Wielen:

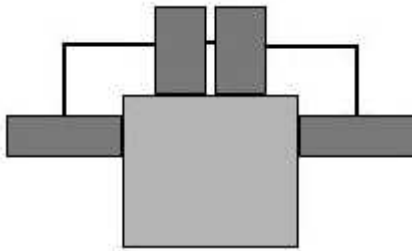
Er is gekozen voor 6 wielen op de houten balk te laten rijden. In de eerste versie werden alle wielen aangedreven met een ketting. De ketting paste juist niet, daarom moest ze aangespannen worden met 2 extra tandwielletjes. Dit gaf te veel wrijving en de wielen liepen stram.



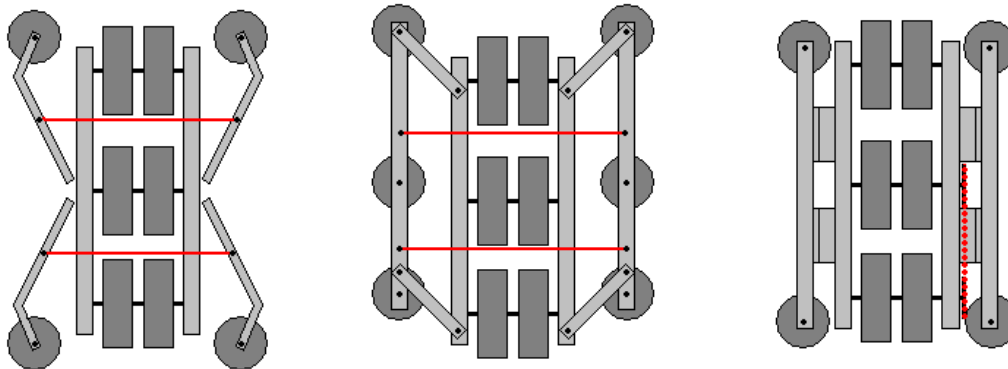
Om de wrijving te verminderen worden er slechts 2 wielen aangedreven met een ketting, dit is het minimum aantal wielen om zeker te zijn dat er altijd minstens 1 aangedreven wiel op de balk steunt



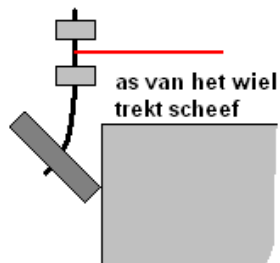
Aan weerszijden van de houten balk staan elk 2 wielen om de stabiliteit te verhogen. Er is eerst geopteerd voor indrukbare wielen omdat die stabiel zijn, maar ze verhogen de kans op scheefftrekken, dus zijn ze vervangen door volle banden die minder scheefftrekken.



Om de wielen te bevestigen heeft men verschillende opties:

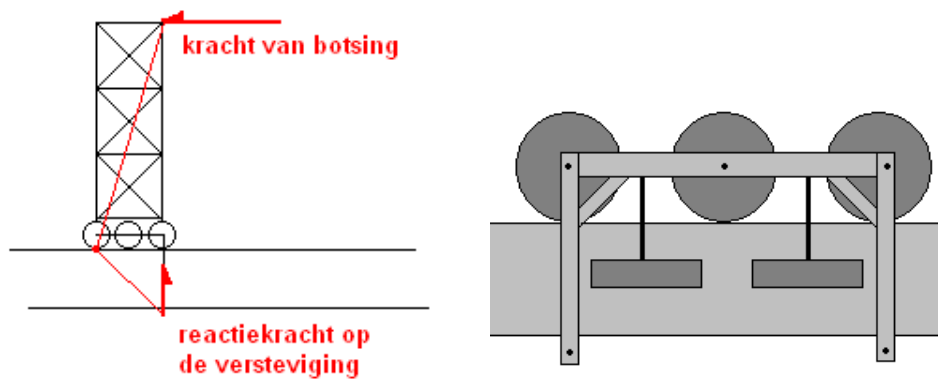


Het nadeel van de eerste 2, waarbij de wielen met elastieken op het hout getrokken worden, is dat de wielas scheeftrekt.



Hierdoor heeft men voor de derde optie gekozen. Deze heeft als nadeel dat de wielen niet zo stevig vastzitten als bij de eerste 2. Aan de andere kant is het wel veel makkelijker om dingen hierop te monteren.

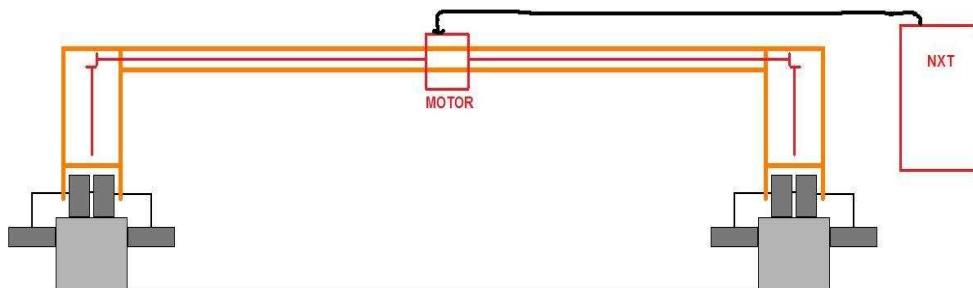
Er kan een probleem ontstaan als de andere robot tegen de bovenkant van de kraan botst, of bij het optrekken als het massamiddelpunt te hoog ligt. De robot kan namelijk kantelen. Dit kan echter tegengehouden worden door verstevigingen aan te brengen onderaan de wielen.



Motoren:

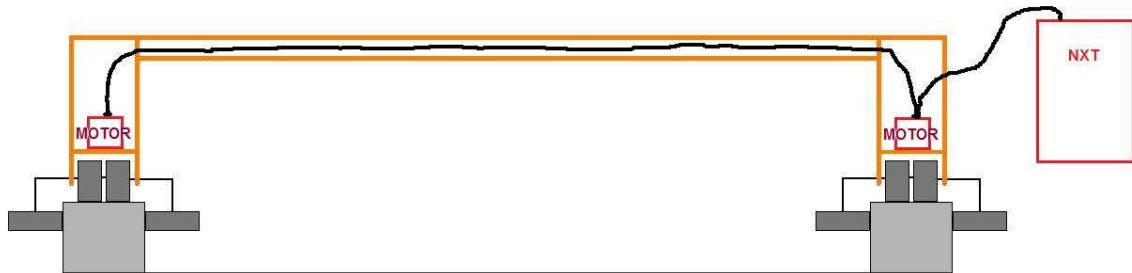
Het oorspronkelijke idee was om 1 motor in het midden te zetten en dan een as naar de 2 poten te laten lopen die dan de wielen aandrijven.

Voordelen	Nadelen
<ul style="list-style-type: none"> - wielen synchroon 	<ul style="list-style-type: none"> - moeilijk te bouwen - as tordeert: kan breken - grote wrijving - buiging as: vermijden door cardanas



Vervolgens werd een aanpassing aangebracht: een motor op elk wiel zetten en die door een lange kabel te verbinden.

Voordelen	Nadelen
<ul style="list-style-type: none"> - eenvoudiger bouwen - geen wrijving 	<ul style="list-style-type: none"> - motoren niet even snel: kraan trekt scheef



Mogelijke redenen waarom de motoren niet even snel zouden rijden zijn:

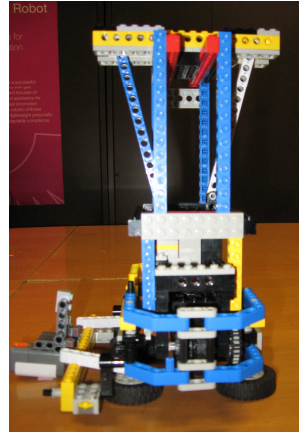
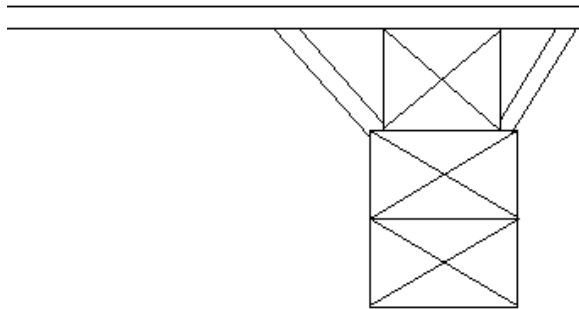
- Imperfecties bij de constructie
- Veroudering, de ene motor heeft meer afgezien dan de andere
- ongelijke belasting
- ongelijke voedingsspanning door kabels
- doorslippen van wielen
- doorklikken van tandwielen

Er zijn tests uitgevoerd en er is gebleken dat indien er evenveel gewicht op de poten rust, het scheeftrekken minimaal is. Dit wordt ook gecorrigeerd bij het terugbrengen van een bal naar de goot: doordat beide wielen tegen de kant botsen staan ze recht.

Steun voor overbrugging:

De aluminium L-profielen die voor de overbrugging zorgen, rusten op de twee poten. Hiervoor zijn in de profielen vijf gaten geboord die de juiste afmetingen hebben om verbindingsspinnen in te steken zodat men deze pinnen met een legoblokje kan verbinden.

Om te voorkomen dat de overbrugging in de richting haaks op de profielen zou wankelen, zijn de basissen van de wielen breed gemaakt. Voor dit in de evenwijdige richting te voorkomen, zijn er vakwerken aangebracht.



Wagentje:

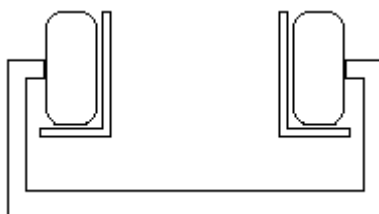
Het wagentje is het onderdeel dat over de overbrugging rijdt en waarop de toren met grijper bevestigd is. Het bestaat uit 2 delen: de wielen en de motor

Wielen:

Het wagentje heeft in totaal 6 wielen, 3 op elke staaf. In de eerste versie staan deze horizontaal op de profielen.



Dit was niet stevig en de wielen trokken scheef. Daarom zijn ze verticaal op de profielen gezet. Omdat de breedte van de staven geen lego-afstand was, is er een stuk afgezaagd zodat het veel simpeler is om de wielen erop te laten rijden. Om de grip te verhogen is de bovenkant van de staven ruwer gemaakt.

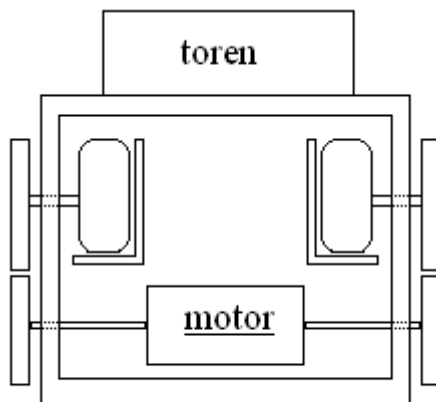


Eerst werd er geopteerd om slechts 2 van de 3 wielen aan te drijven, maar omdat er dan achterwielaandrijving was in een rijrichting begon de toren te schommelen bij het optrekken. Hierdoor raakte niet altijd voldoende wielen de profielen en schokkend draaiden ze door, wat

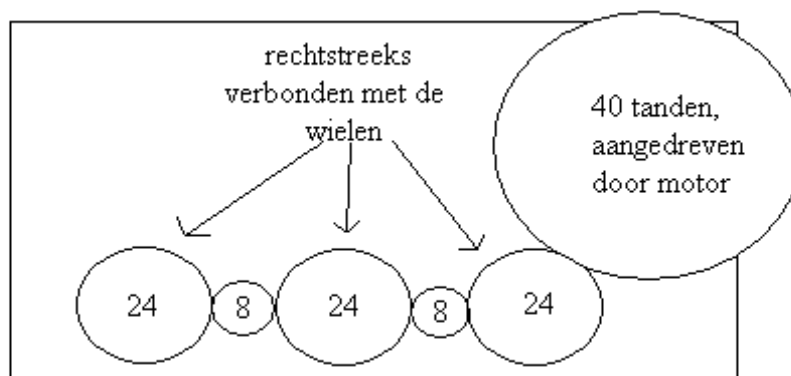
softwareproblemen gaf. De wielen draaiden wel, maar volgens het programma was het wagentje verder gereden dan werkelijk gebeurde. De oplossing is simpel: alle wielen aandrijven. Hierdoor schokt de toren veel minder.

Motor:

Het wagentje wordt aangedreven door een servomotor die onder de staven hangt.



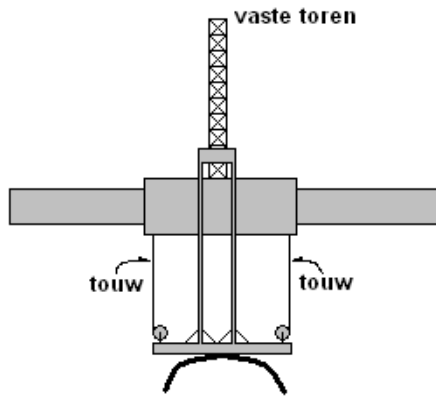
De tandwieloverbrenging van de motor naar de wielen is 3:5, d.w.z. dat als de motor 3 rotaties maakt, het wiel er 5 doet. Dit wordt als volgt gerealiseerd:



Toren

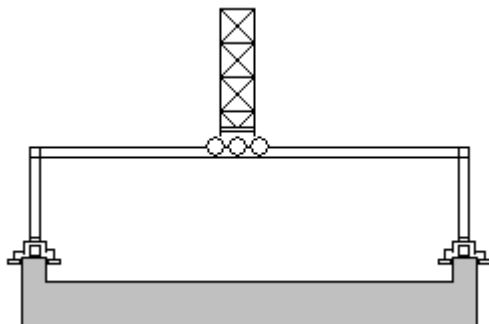
Om de grijper te begeleiden tijdens zijn verticale beweging heeft men voor een toren gekozen. Dit omdat touwen niet star genoeg zijn en de grijper dus zou schommelen. Het schommelen zou men kunnen compenseren door met een begeleidende staaf te werken. Die zou dan boven op de toren gezet

worden om niet in de weg te staan. Er zou dan een constructie op het verticaal bewegende deel over kunnen schuiven om de schommeling te blokkeren.



Men kon de grijper vast aan de toren hangen en heel de toren naar boven en beneden laten gaan, maar dan is het zeer moeilijk om nog met eenzelfde motor de tandwielen die vast staan op het rijdende gedeelte en de grijper aan te drijven. Er werd dus gekozen voor een vaste toren. Als de toren onderaan het rijdende gedeelte zou hangen, zou de andere robot ertegen kunnen botsten. Er werd dus gekozen om de toren bovenop het rijdende gedeelte te zetten.

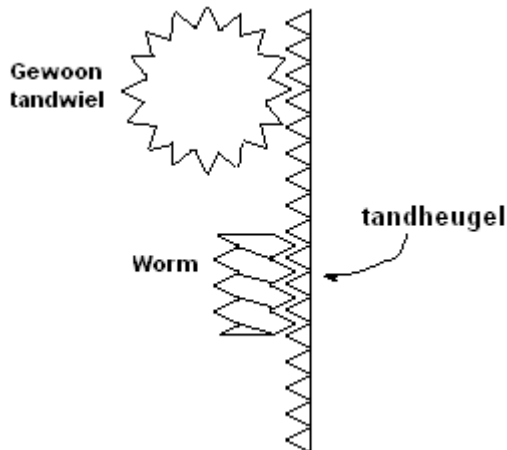
De toren is verstevigd met vakwerk om de structuur stevig genoeg te maken².



² Zie het onderdeel over vakwerken op pagina **XXXXXXXX**

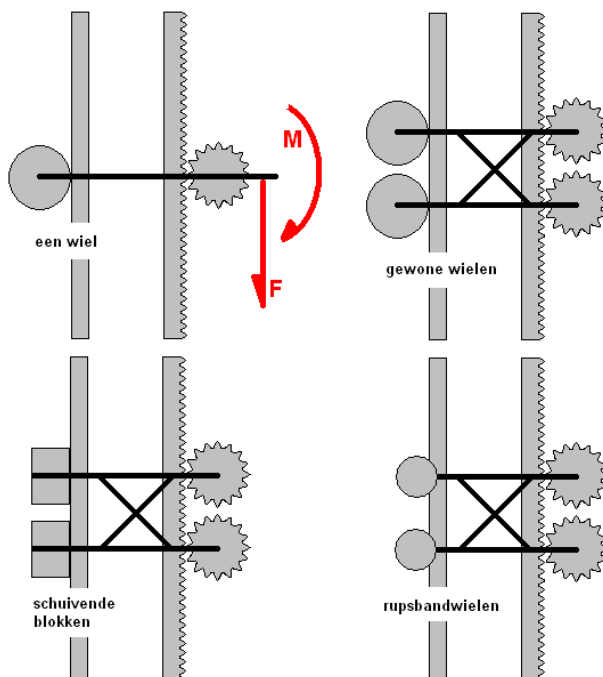
Verticale beweging op de toren

Om de grijper op de toren te bewegen wordt er gebruik gemaakt van een tandheugel op één kant van de toren. Om op de heugel te bewegen waren er 2 opties: Met een tandwiel of met een worm



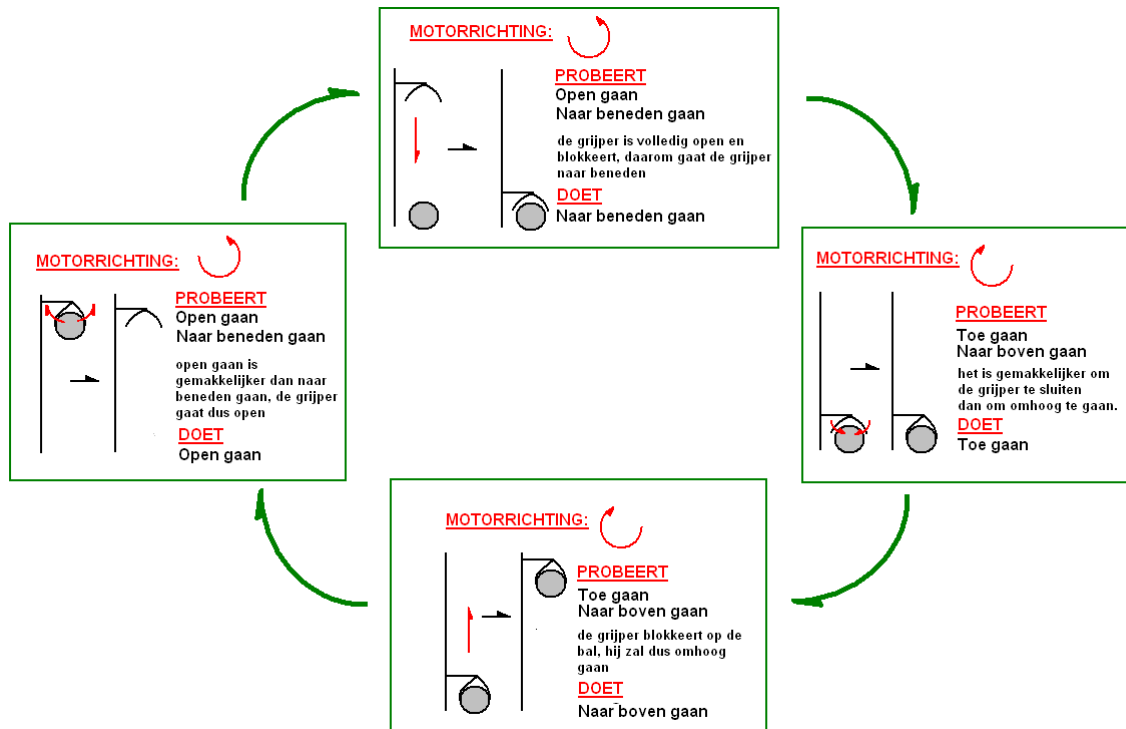
De constructie met de worm is niet uitvoerbaar in Lego omdat de wormen die Lego maakt te dicht op de tandheugel hangen om aangedreven te kunnen worden. Om deze reden is er gekozen voor het gewone tandwiel. Eerst maakte men het verticaal bewegende deel een maal vast aan de toren met een wiel dat erop reed. Deze constructie kon de trekkracht van de grijper niet voldoende opvangen en men hing er daarom een extra tandwiel aan.

Omdat de afstand niet exact uit te voeren is in Lego zaten de wielen aan de achterkant te los of te vast. Een mogelijk alternatief was om gewone gladde blokken te laten schuiven over de toren. Men vond een goed alternatief in wielen die gebruikt worden voor rupsbanden. Deze hebben als bijkomend voordeel dat ze over hun 'rails' schuiven en er niet kunnen afspringen.



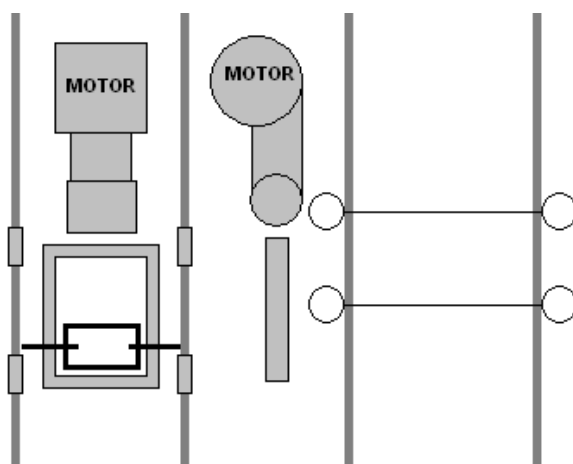
Op de hierboven vermelde constructie wordt dan de motor, de differentieel en de grijper gemonteerd. De functies dalen en stijgen en openen en sluiten gebeuren met een differentieel.

Als men ervoor zorgt dat het open gaan gemakkelijker is dan het dalen en het sluiten gemakkelijker dan het omhoog gaan vindt men volgend schema:



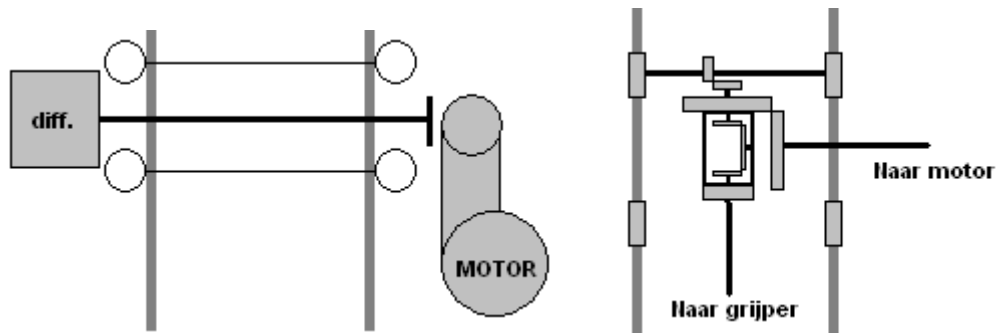
Montage van de differentieel op de toren

Bij de eerste constructie hingen de motor, grijper en differentieel aan dezelfde kant van de toren:



Bij deze uitvoering was de massaverdeling slecht omdat alles aan de voorkant hing. De eerste tests die ermee gebeurden toonden wel aan dat de motor voldoende sterk is om de constructie snel genoeg omhoog te krijgen.

Om de massaverdeling te verbeteren hing men de motor aan de andere kant en liet men de aandrijfjas door het midden van de toren lopen naar de voorkant waar de differentieel hangt.

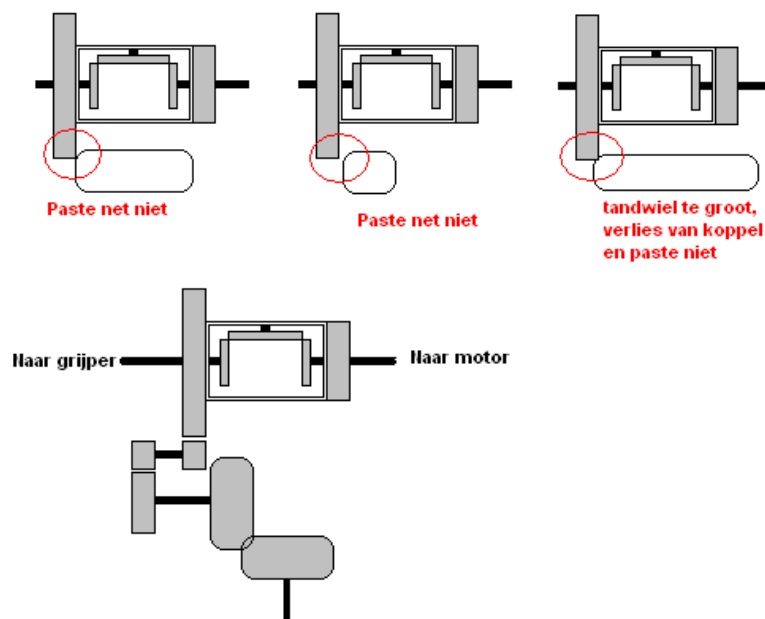


Men koos ervoor om de differentieel horizontaal te hangen om het aantal tandwielverbindingen dat 90° draait te beperken tot één. Als de differentieel verticaal wordt gehangen zou de grijper gemakkelijker aanspreekbaar zijn.

Men herbouwde de differentieel zelf ook om hem steviger te maken. De voorwaarden waaraan de nieuwe differentieel moest voldoen waren:

- Zo dicht mogelijk bij het aandrijvende tandwiel zitten.
- Grote kracht kunnen genereren en kunnen weerstaan
- Een aansluiting hebben in het midden om naar de motor te kunnen.

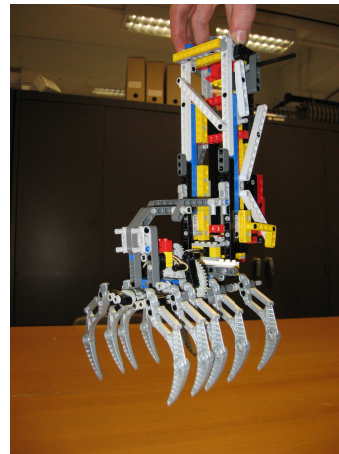
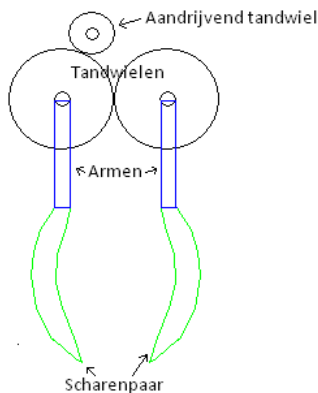
De eerste versie was te ingewikkeld en bevatte te veel tandwielen. Het vereenvoudigen was niet gemakkelijk omdat de tandwielen een hoek van 90 graden moesten maken. Hierdoor werd het aantal tandwielen beperkt en velen pasten niet of klikten te gemakkelijk door. Er moest ook koppel gewonnen worden naar de differentieel toe om voldoende kracht op te wekken om de grijper omhoog te krijgen.



Uiteindelijk vond men tandwielen die niet gemakkelijk doorklikten. Door combinatie van deze tandwielen met een klein tandwiel dat naar de differentieel gaat kon er ook koppel gewonnen.

De gripper

Het gebruikte grijpsysteem



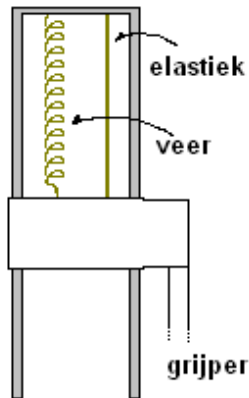
Het uiteindelijke grijpsysteem bevat 5 scharenparen evenwijdig en symmetrisch gemonteerd op twee armen, elk aangedreven door eenzelfde tandwiel.

Het gebruik van één enkele aandrijving voor de twee armen is nogal logisch, omdat als arm 1 een uitwijking van 30° maakt, arm 2 een uitwijking van -30° moet maken. Een aandrijving voor elke arm voorzien, vermoeilijkt het grijpen enkel en vereist een extra motor.

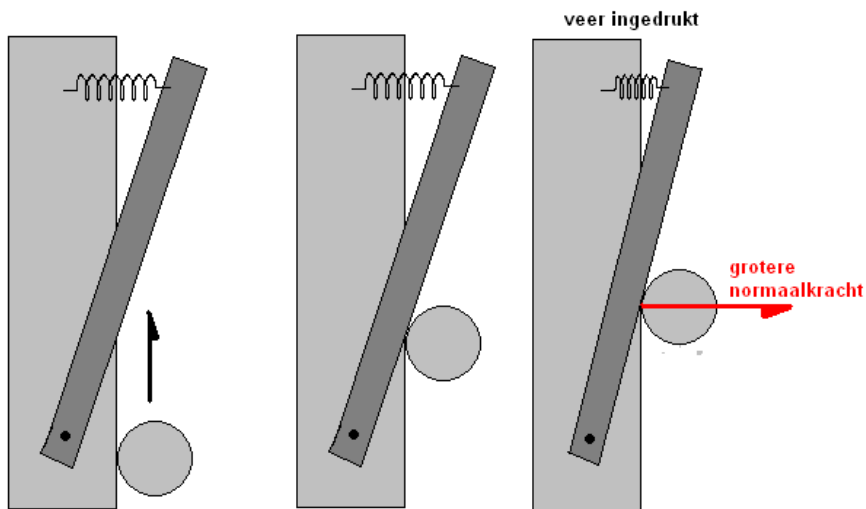
De grootte van de tandwielen waarop de armen gemonteerd zijn maakt niet zoveel uit, zolang ze even groot zijn. Hierdoor wordt er geen koppel verloren of gewonnen wordt ten opzichte van elkaar en zullen de beide armen zo een evengrote, tegengestelde beweging maken. Bij de keuze van de grootte van deze tandwielen hield men voornamelijk rekening met het praktische aspect. Ze moesten plaats laten voor de lichtsensor, die tussen de armen gemonteerd werd. Bovendien werd er gestreefd naar een zo stevig mogelijk grijpsysteem, dat ballen wel degelijk deftig kan klemmen, dus mochten zeker niet de kleinste tandwielen gebruikt worden.

Het aantal scharenparen werd gekozen op basis van de nauwkeurigheid waarmee de kraan boven de bal kon gaan hangen nadat hij gedetecteerd was. Bij een gripper met drie armen bleek de kraan niet nauwkeurig genoeg te zijn, er werd dus gekozen voor vijf armen om het missen te minimaliseren.

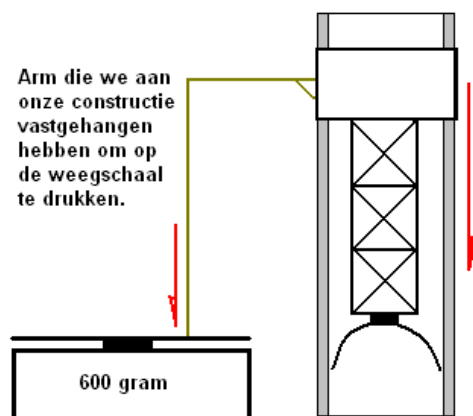
Omdat bij deze gripper het opengaan niet gemakkelijker kon gemaakt worden dan het naar beneden gaan (of als het lukte ging hij niet meer dicht voor hij omhoog ging) moest het naar beneden gaan moeilijker gemaakt worden. Eerst werd er met elastieken en veren geprobeerd, maar hun deze boden geen oplossing, bij veren en elastieken wordt de kracht namelijk groter met de afstand, terwijl de kracht juist maximaal moest zijn als de gripper bovenaan staat.



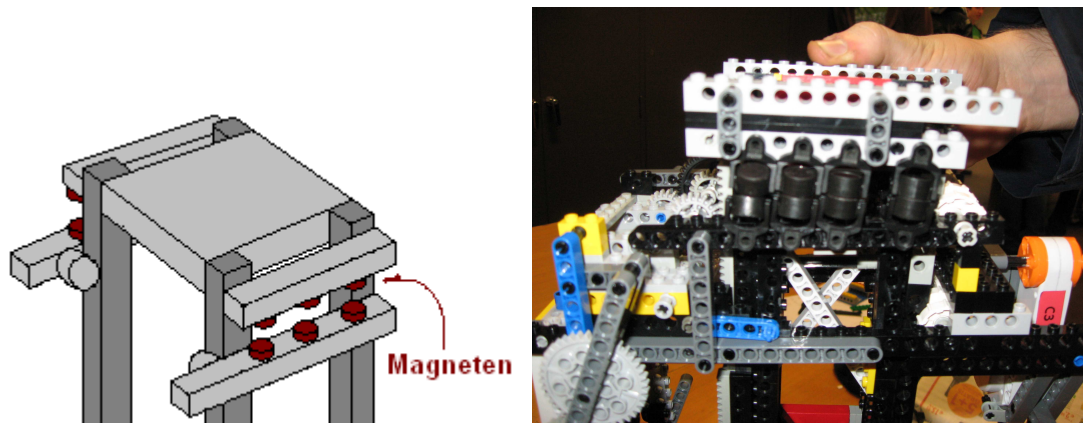
Er werd met een uitschuivend stuk geprobeerd om de wrijving te vergroten vanboven, maar de kracht was nog steeds te klein. De lego is redelijk glad, waardoor de wrijvingscoëfficiënt niet voldoende hoog was. Hierdoor kan men niet de nodige remming halen.



Door metingen werd bepaald dat de vereiste kracht om de grijper eerst te laten opengaan 600 gram was. De tekening hieronder toont hoe dit bepaald werd: aan het verticaal bewegende stuk werd een arm gehangen die op een weegschaal drukt. Het gewicht dat de weegschaal afmeet als de grijper opengaat, is het gewicht dat moet tegen gehouden worden aan de bovenkant.



Er werd gekozen voor magneten om bovenaan te blokkeren, elke magneet kon 100 gram trekken en daarom werden er 6 opgehangen aan de constructie. Om het evenwicht te bewaren heeft men 3 magneten aan elke kant van de toren gehangen.



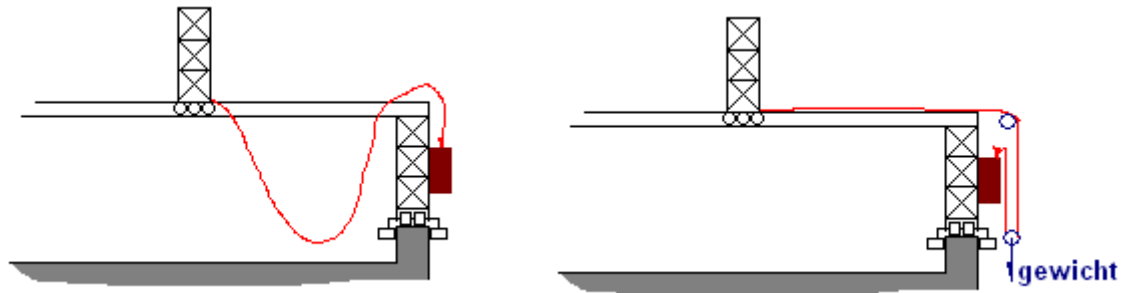
Toegevoegd materiaal

Zoals al eerder vermeld, waren de kabels geleverd bij het Lego Mindstorms NXT bouw pakket te kort voor de verbindingen die nodig waren bij het bouwen van de kraan. Daarom werden extra kabels aangekocht. Bij het monteren van deze kabels traden er problemen op met de ophanging en het vast blijven van de aansluiting in de motoren. Dit wordt kort toegelicht. Voor het gebruikte detectiesysteem diende er gebruik te worden gemaakt van een laser. Deze werd niet geleverd bij het bouw pakket en moest dus door het team zelf worden toegevoegd. De verschillende opties die hiervoor zijn afgewogen worden kort besproken.

Kabelbegeleiding

Na het verbinden van de kabels uit de toren met de NXT module stelde zich een probleem bij beweging van de toren. Als de toren dichterbij de NXT module kwam, begonnen de kabels in de weg te komen: ze botsten tegen de balletjes en lagen voor de sensoren. Er moest dus een oplossing gevonden worden voor dit probleem.

Een eerste mogelijkheid was om de kabels naar beneden te trekken met een gewicht. Hierdoor zouden de kabels constant gestrekt blijven en dus niet in de weg komen te liggen. (zie figuur)

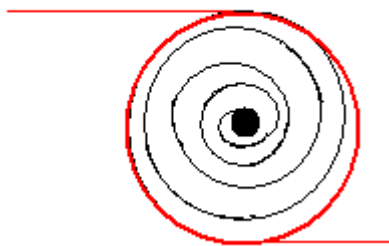


De kabels waren echter niet lang genoeg om deze methode toe te passen.

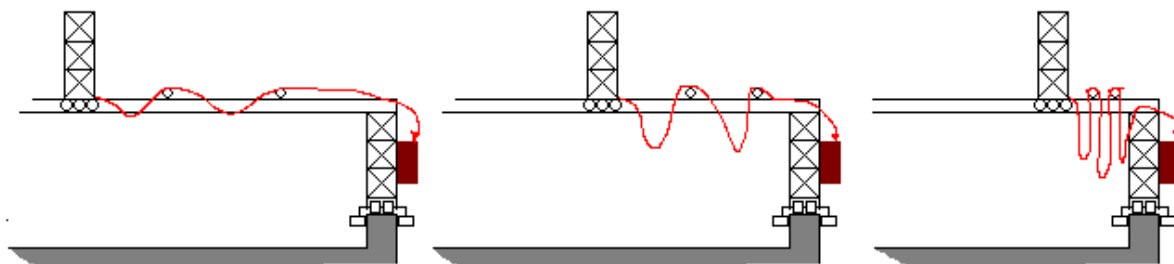
Een tweede mogelijkheid was om tussen de L profielen een platform te maken zodat de kabels hierop konden rusten en dus niet meer in de weg kwamen te liggen. Het probleem hierbij was dat de constructie de L profielen uit elkaar duwde, waardoor de toren niet meer vlot reed. (zie figuur)



Nog twee andere mogelijke oplossingen zijn een zelfoprollend systeem (cilinder met een veer in die zich opspant als de kabel afrolt) en een kabelrups. Deze systemen zijn te ingewikkeld om te verwezenlijken.



Onze uiteindelijke oplossing bestond eruit dat er 2 punten van de kabels op de aluminium profielen gehangen werden die op de profielen kunnen bewegen. (zie figuur).



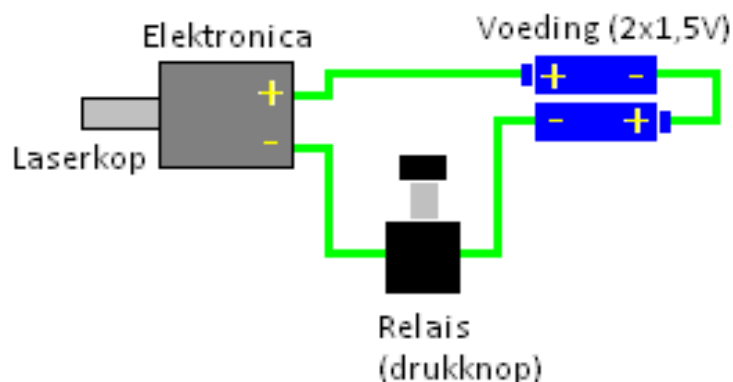
Laser

Zoals reeds uitgelegd bestaat de eerste stap van ons detectiesysteem uit een laser die constant op een lichtsensor is gericht. Bij onderbreking van de laserstraal zal de grijparm op de lijn van de laserstraal zoeken naar een bal. In deze paragraaf wordt dieper ingegaan op dit systeem.

De gebruikte lichtsensor is de standaard Lego sensor die ter beschikking werd gesteld. Als laser werd eerst een helium-neon laser met een vermogen van 10mW gebruikt. Deze laser werd uitgeleend bij de vakgroep TONA. Het probleem met deze laser was dat de lichtsensor constant de maximale waarde weergaf bij het beschijnen, waardoor er gevreesd werd dat de intensiteit van de laser nog groter was dan wat de lichtsensor kon verdragen en dat deze de lichtsensor zou beschadigen bij langdurige bestraling. Het was aangewezen op zoek te gaan naar een minder sterke laser.

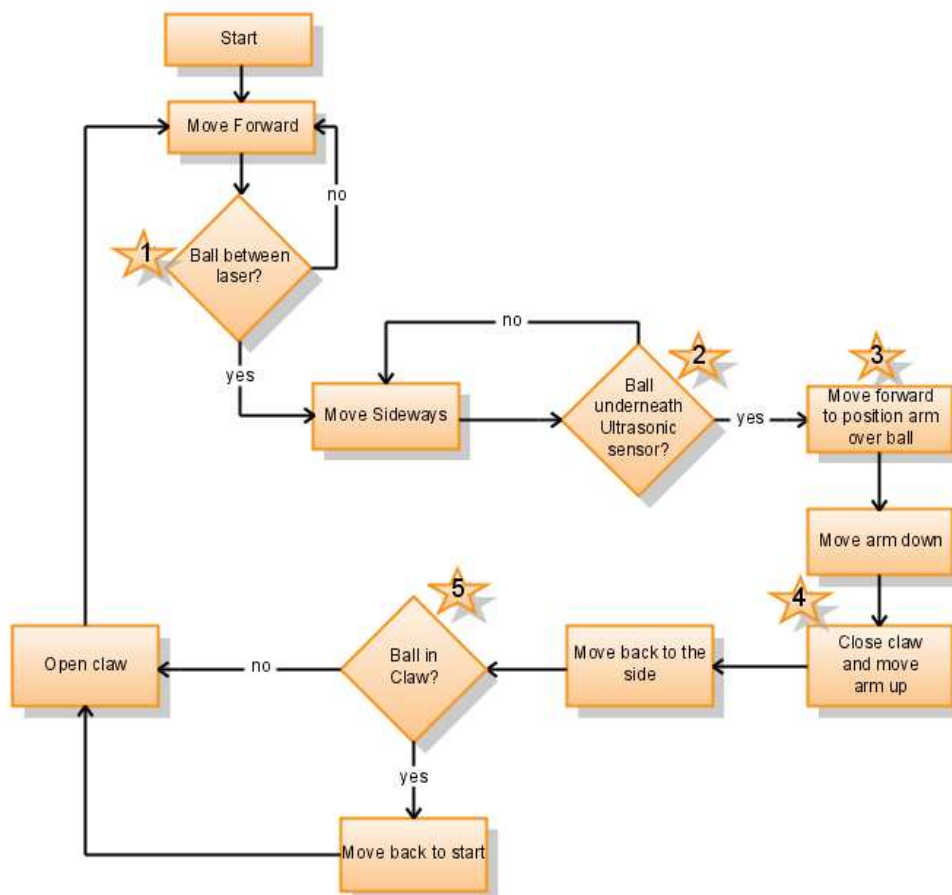
De tweede keuze was een helium-neon laserpointer. Het was het soort pointer dat doorgaans te winnen valt bij kermisattracties. De stip die men waarneemt bij zulke lasers is echter veel te klein. Deze zou namelijk constant gericht moeten zijn op het zeer kleine oog van de sensor, welke zich 1m30 verder bevindt. Dit terwijl de robot rijdt en dus schokt. Om dit probleem op te lossen werd een lens uitgeleend bij de vakgroep TONA. De piepkleine stip wordt hierdoor omgezet naar een voldoende grote vlek. Hiermee deed zich echter een nieuw probleem voor. Door de lens verliest de laserbundel immers aan intensiteit, waardoor de vlek voor een te kleine fluctuatie in uitgelezen waarden van de lichtsensor zorgt. Er moest op zoek gegaan worden naar een iets sterkere laser.

De derde en uiteindelijke keuze viel op een laser specifiek ontworpen als pointer voor presentaties. Door zijn specifieke doel viel deze laser nogal groot uit. Hij werd gedemonteerd en de elektronische componenten en laserkop werden eruit gehaald. Zo kon de laser compacter in een Legoconstructie geplaatst worden. De voeding, twee AAA batterijen van elk 1,5V, moest natuurlijk met elektrische draadjes aan de laserkop worden gesoldeerd, bovendien werd een handige drukknop toegevoegd, zodanig dat de laser na het monteren makelijker aan en uit kon worden gezet. Hier een schema van de uiteindelijke schakeling:



Het Programma

Hoewel C# een object georiënteerde taal is, en men dus meestal met verschillende klassen zal werken, volstaat het in dit geval de aansturing van de robot in een enkele klasse te laten gebeuren en functioneel te programmeren. Dit is mede mogelijk door de 'lineaire' structuur van het uiteindelijke programma, waarvan hieronder een flowchart te zien is.



Gemaakt via: www.gliffy.com

Nadere kijk op de delen aangeduid met een genummerde ster:

1. De robot is vooruit aan het rijden, terwijl continu, zo snel als de verbinding toelaat, de lichtsensor, uitgebreid met laser, uitgelezen wordt. De waarde die deze teruggeeft wordt vergeleken met een referentiewaarde, uitgelezen na de start. Wanneer het verschil tussen de waarde en de referentiewaarde te groot is, dan bevindt er zich een bal tussen laser en lichtsensor.

De code die dit realiseert ziet er als volgt uit:

```

--
lightsensor1.Polled += new Bram.NxtSharp.SensorEvent(LineDetected);1
--
lightsensor1.Poll();
--

void LineDetected(Bram.NxtSharp.NxtSensor sensor)
{
    if (Math.Abs(lightdefault - lightsensor1.Value) > lightchange)2
    {
        nxtBrick1.MotorA.Brake();
        ScanLine();3
    }
    else
    {
        lightsensor1.Poll();4
    }
}
}

```

Wanneer de lichtsensor (lightsensor1) uitgelezen wordt (Poll), wordt de methode 'LineDetected' uitgevoerd.

- Het verschil tussen de uitgelezen waarde en de referentiewaarde wordt vergeleken.
 - Het verschil is groot genoeg (bal gevonden), dus er moet zijwaarts gescanned worden.
 - Het verschil is niet groot genoeg, dus er wordt er wordt een nieuwe waarde opgevraagd van de sensor.
2. De lichtsensor heeft aangegeven dat er zich een bal op de lijn bevindt en de arm wordt zijwaarts bewogen over deze lijn. De waarde van de ultrasone sensor geeft aan of er al dan niet een bal ligt.

Hieronder is er een test te zien, met de uitgelezen waardes van de ultrasone sensor, waarbij de sensor over een bal bewogen werd. De positie van de bal is hierin duidelijk zichtbaar aan de lagere waardes van de sensor die hier voor de duidelijkheid omkaderd zijn.

...13 13 14 14 13 13 13 12 13 11 11 8 6 7 7 7 6 6 7 8 8 7 10 9 10 12 11 11 11 11 11...

De code hiervoor is analoog aan bovenstaand voorbeeld.

3. Door de constructie van de robot liggen lichtsensor en ultrasone sensor niet op eenzelfde lijn met de grijparm. Wanneer de lichtsensor een lijn aanduidt waarop een bal ligt, en de ultrasone sensor op zijn beurt de bal gelocaliseerd heeft, moet de gehele robot nog een kleine afstand vooruit rijden om zo de grijparm boven de bal te positioneren. Aangezien deze beweging zich

voltrekt met behulp van 9V motoren, in plaats van servomotoren, kan hier geen gewenste rotatie ingesteld worden. De correcte verplaatsing wordt dus bekomen door de motoren aan te zetten voor een bepaalde duur.

4. Hier komt de werking van de differentieel aan te pas. Deze combineert het naar beneden gaan van de arm met het open gaan van de grijper, en het naar boven gaan met het sluiten.

Noemt men de richting waarin de motor moet roteren om het eerste te realiseren 'richting1', de richting voor de tweede beweging 'richting2', de hoek waarover de motor moet roteren om de arm volledig naar beneden te laten gaan 'hoek1' en de hoek om de grijper te openen 'hoek2', indien dan gegeven dat de grijper zich initieel bovenaan en in open positie bevindt, dan moet de motor, om de bal te grijpen, volgend proces doorgaan: 'richting1' over 'hoek1' (grijper bevindt zich nu beneden), gevolgd door 'richting2' over 'hoek2'+ 'hoek1' zodat de grijper dichtgaat en zich weer bovenaan bevindt.

Wanneer men, op het einde om de bal naar de goot te brengen, de motor in 'richting1' over 'hoek2' laat roteren, bevindt de robot zich weer in de startpositie.

5. Voordat een gegrepen bal teruggebracht wordt, wordt de lichtsensor in de grijper uitgelezen. Deze waarde is een indicator om te zien of er zich al dan niet een bal in de grijper bevindt; een hoge waarde, dus een sterkere lichtintensiteit, wijst op de afwezigheid van een bal onder de sensor.

Dit vermijdt een nodeloze terugkeer naar de startpositie wanneer de grijper de bal gemist heeft.

Ingebouwde correcties om scheef rijden tegen te gaan

Een eerste correctie gebeurt wanneer de robot een gegrepen bal naar de achterkant van het veld brengt. De robot detecteert deze achterkant met een druksensor, die ingedrukt wordt wanneer de achterkant bereikt is. De motoren blijven hier even draaien om op deze manier de robot met beide poten tegen de kant te trekken.

Wanneer de robot scheefgetrokken is, is de spot van de laser niet meer op de lichtsensor gericht. Dit zorgt voor hetzelfde effect als een bal tussen laser en sensor. De robot zal deze niet bestaande bal gaan zoeken, maar niet vinden. Wanneer de grijparm de volledige overspanning afgelopen heeft zonder bal te detecteren, keert hij terug. Gebeurt dit tweemaal op rij, is dit een indicatie dat de robot scheefgetrokken is. Hij zal dan terugkeren naar de achterkant om zichzelf recht te zetten.

Kleinigheden

Zoals vermeld maakt de laser-lichtsensor gebruik van een richtwaarde gemeten in het begin. Om verkeerde uitlezingen van de sensor tegen te gaan, wordt deze waarde enkele keren gemeten om daarna een gemiddelde te nemen.

Wanneer de grijper een bal gegrepen heeft, wordt de arm van de robot terug naar de zijkant gebracht voor een betere massaverdeling. Hoewel de positie van deze arm in principe gekend is met behulp van de eerder vermelde tachocount, is het niet aangewezen om de motor in tegengestelde richting te laten draaien over deze hoek om terug aan de zijkant te belanden. Dit resulteert in onnauwkeurigheden, waarbij de arm te ver zal willen rijden, maar geblokkeerd wordt door de poot van de robot. Het programma zal wachten totdat de rotatie is uitgevoerd, maar door de blokkering zal dit niet gebeuren. Hiervoor is een veiligheid ingebouwd, zodat wanneer de motor niet meer roteert het programma de opdracht om het aantal specifieke graden te roteren schrapt en verder gaat met zijn volgende taak, de bal naar de goot brengen.

Besluit

Alvorens een conclusie te formuleren over het afgeleverde product, wordt er eerst een synthese gemaakt van de manier waarop dit product en dit verslag tot stand zijn gekomen.

In alle eerlijkheid moet het team toegeven dat de methode van de ontwikkeling van de robot vooral tijdens de eerste werkdagen niet altijd getuigde van een genuanceerd ingenieursdenken. Tijdens het ontwerpproces werd vaak veel te snel een beslissing genomen zonder andere mogelijkheden af te wegen en zonder de werkelijke uitwerking van deze beslissingen grondig te bestuderen. Hoewel er tijdens de eerste drie werkdagen regelmatig door de begeleiders werd gewezen op het belang van conceptueel denken, heeft het team, en dus ook de coördinator, pas later het nut hiervan ingezien. Belangrijk is dat naar het einde van het project toe, alle leden van het team zich wel realiseerden hoe belangrijk het is om te denken alvorens te doen.

Bij het maken van dit verslag is volgens het team wel de juiste aanpak gehanteerd. Het verslag is opgebouwd vanuit het probleem, men heeft dan aan de hand van het materiaal concepten beschreven en op basis hiervan een keuze gemaakt en deze grondig bestudeerd en uitgewerkt. Bepaalde delen zijn daarom chronologisch gezien niet helemaal waarheidsgetrouw. Sommige concepten zijn bijvoorbeeld niet getest en zijn pas bedacht nadat de keuze voor het kraanmodel al gemaakt was. Deze zijn dan verder uitgewerkt en tijdens het maken van het verslag toegevoegd. Met dit verslag wil het team aantonen dat ieder lid veel heeft bijgeleerd en dat men nu ook beter is staat is om een project als dit op een constructieve manier aan te pakken.

Het aantal tekens van het technische verslag overschrijdt weliswaar de afgesproken limiet maar het team staat hier volledig achter. Stukken wegnippen uit dit verslag zou afdoen aan de inhoud en zou voor onvolledigheid zorgen. Indien men het aantal tekens van de hoofdstukken 'Conceptvoorstellen' en 'Ontwerp en Bouw' optelt komt men ongeveer aan de beoogde 50 000 tekens maar het team vond het ook zeer belangrijk om een grondige materiaal- en softwarestudie en een duidelijke argumentatie toe te voegen.

Op het uiteindelijke resultaat is het team zeer trots. De robot presteerde, op een paar typische schoonheidsfoutjes na, zeer goed tijdens de wedstrijd en kreeg ook van veel medestudenten goede kritiek. Op vlak van technische realisatie zitten er enkele indrukwekkende en keurig afgewerkte elementen in. Ook de verschillende oplossingen voor technische problemen, zoals het toevoegen van o.a. de magneten en het systeem voor de kabelbegeleiding, zijn meestal eenvoudig maar efficiënt. De voornaamste problemen bij het demonstreren, ontstonden door fouten van het lasersysteem en door het scheeftrekken van de hele robot. Het eerste kwam omdat deze laserstraal door een lens werd gestuurd die op zijn plaats werd gehouden met plakband, wat geen betrouwbare methode is. Het tweede grote werkingsprobleem kwam door de niet-synchrone werking van de motoren bij een ongelijke massaverdeling.

Hierdoor presteerde onze robot vrij gemiddeld in vergelijking met andere robots. De 2 genoemde oorzaken van falen van de robot waren de avond voor de wedstrijd getest en op dat ogenblik werden hierbij geen problemen ondervonden.

Vele andere robots gebruikten ook de methode van het blind op zoek gaan naar ballen wat door de kleine omvang van het veld geen problemen gaf. De prestatie van de robot van team A, onze robot, was totaal onafhankelijk van de grootte van het veld en zou ook op een veel groter veld, waar deze andere robots veel aan prestatie zouden verliezen, goed blijven presteren.

Als conclusie stelt Team A dus dat het goed in zijn opzet geslaagd is

Team A zou graag de volgende mensen bedanken:

De begeleiders Ronald en Jelle, Jean-Paul, de mensen van het atelier, de andere begeleiders, Professor Stijns (TONA) en de cola-automaat in het atelier

Boekhouding

Werkuren per dag:

dag 1:

- Zjef & Avdyl: programmeren: 5,5u
- Maarten & Goedele: sensoren 5,5u
- Adam, Hannes & Pieter: concepten bedenken: 5,5u

dag 2:

- Avdyl: afwezig
- Iedereen: rondleiding: 1,5u
- Zjef: programmeren: 6u
- Goedele, Adam, Hannes & Maarten: concepten: 4u
- Pieter: wagentje: 6u
- Maarten: grijper: 2u
- Hannes: toren: 2u
- Adam: poten: 2u
- Goedele: opslagsysteem: 2u

dag 3:

- Avdyl: afwezig
- Zjef: programmeren: 8u
- Maarten & Hannes: differentieel: 7u
- Pieter & Adam: poten: 7u
- Goedele: programmeren: 4u
 - : opslagsysteem: 3u
- Hannes: grijper: 1u
- Pieter: laserdetectie: 1u

- Maarten & Adam: differentieel: 1u

dag 4

- Avdyl: programmeren: 2u
- overbrugging: 4u
- Zjef: programmeren: 8u
- Goedele: verslag: 6,5u
- Pieter: laserdetectie: 8u
- Adam & Maarten: differentieel: 7u
- Hannes: grijper: 7u
 - Planning en voorbereiding: 1u
- Maarten & Adam: grijper: 1u

dag 5:

- Zjef & Avdyl: programmeren: 8u
- Goedele: sensoren: 2u
- Goedele: verslag: 4u
- Adam: grijper: 8u
- Pieter: wagentje: 2u
- laserdetectie: 6u
- Hannes: verslag: 8u
- Maarten: poten: 2u
- grijper: 4u
- laserdetectie: 2u

dag 6:

- iedereen: testen: 8u

extra dag: dinsdag 17/03

- Hannes: verslag: 1u
- Pieter: toren: 2u
- Adam: differentieel: 2u

extra dag: donderdag 19/03

- Adam: toren: 4u
- Hannes: verslag: 2u

extra dag: vrijdag 20/03

- Avdyl & Goedele: testen & kalibreren: 7u
- Adam, Zjef, Maarten, Pieter, Hannes: testen & kalibreren: 12u

extra dag: maandag 23/03

- Pieter & Maarten: kalibreren: 5,5u
- Adam & Zjef: kalibreren: 9u

Werkuren per persoon:

- Adam Cooman: 70,5u
- Hannes Maes: 58,5u
- Pieter Ruelens: 63u
- Avdyl Shala: 34,5u
- Zjef Van de Poel: 64,5u
- Goedele Van kerkhoven: 46u
- Maarten Weckx: 59u

totaal: 395u**Werkuren per onderdeel:**

- concepten: 32,5u
- programmeren: 55u
- sensoren: 13u

- wagentje: 8u
- poten: 16u
- toren:8u
- grijper:24u
- differentieel: 32u
- opslagsysteem: 5u
- laserdetectie: 17u
- testen: 93u
- kalibreren: 66u
- verslag: 21,5u
- overbrugging: 4u

totaal: 395

Reflectie

Goedele Van kerkhoven



De eerste sessie, het brainstormen, was heel hoopgevend. We hadden veel ideeën, zelfs veel originele ideeën, en alle groepsleden schenen erg aangesproken door de opdracht. Tijdens het brainstormen en plannen met de hele groep heb ik notities genomen. Met die nota's heb ik dan een eerste overzicht gemaakt van de conceptvoorstellen, waar we ons ook later op gebaseerd hebben voor het verslag.

Het idee om een kraan te maken sloeg vanaf het begin bij iedereen erg aan. We waren er allemaal enthousiast over, en dachten dat die eensgezindheid een groot voordeel was om snel van start te kunnen gaan.

Aan het eind van de eerste werkdag wezen de begeleiders er ons op dat we onze beslissing beter nog wat zouden uitstellen.

Het was niet gemakkelijk om het enthousiasme nog even van ons af te zetten en terug naar de vorige fase te gaan, waar we terug alles in vraag moesten stellen.

We hebben hieraan nog een deel van de tweede werkdag besteed, maar niemand scheen van gedacht te veranderen, dus besloten we definitief voor de kraan te kiezen.

Omdat de hoofdkeuze vastgelegd was, konden we na wat takenverdeling elk aan onze deelopdracht beginnen.

Ik heb toen gewerkt aan een flowchart voor het programma. Daarbij werden heel wat problemen duidelijk waar het concept nog aan moest aangepast worden. (bijvoorbeeld: we hadden er nog niet aan gedacht dat het moeilijk zou zijn voor het programma om op elk moment te weten wat de robot juist aan het doen was)

Ondertussen waren de andere groepsleden volop aan het bouwen geslagen, ze schenen ook allemaal veel ervaring te hebben in het bouwen met lego. Op dat vlak was ik duidelijk minder gevorderd dan zij, dus kon ik maar weinig aan hun werk toevoegen.

Ik had ook zelf niet verwacht dat er zo snel ging gebouwd worden, omdat ik er vanuit ging dat we eerst uitgebreide schetsen zouden maken, en daarna pas beginnen. Dit was ook omdat ik niet gewoon was om me iets in te beelden dat uit lego gemaakt is.

De teamgenoten verzekerden mij dat het in lego gemakkelijker is om te bouwen zonder eerst afstanden uit te meten, omdat de kunst erin bestaat om te denken in de lego lengte-eenheid, dus niet in cm of mm.

Ik geloofde mijn teamgenoten, en was blij dat zij snel vooruit gingen met de constructie. Dit (blinde) geloof in mijn team werd aan het wankelen gebracht doordat de begeleiders de commentaar gaven dat we misschien te snel waren begonnen met de bouw, en beter nog meer onze concepten hadden uitgewerkt.

Dit gaf een dubbel gevoel, omdat ik enerzijds blij was dat de constructie opschoot, anderzijds doken er heel wat onverwachte problemen op, dus moest er dikwijls herbegonnen worden.

Er waren ook veel problemen bij die we waarschijnlijk, zelfs na het maken van uitgebreide schetsen, toch niet hadden kunnen voorzien.

Even rees er de vraag of we een opslagsysteem konden toevoegen aan onze constructie. Omdat deze uit een ander materiaal dan lego kon gemaakt worden voelde ik me geroepen om daaraan te beginnen. Helaas bleek dit achteraf nutteloos, omdat er met de groep beslist werd om geen opslagsysteem te gebruiken. (we liepen wat achter op het schema, en daarom besloten we eerst een eenvoudige versie af te werken, zonder opslagsysteem. Omdat de tijd inderdaad erg beperkt was is het uiteindelijk bij die versie gebleven)

Daarna ben ik al begonnen aan het opstellen van het verslag. Ik heb vrij veel aandacht besteed aan het opstellen van enkele stukken tekst, omdat hieruit onze methodiek moest blijken.

Gezien we in het begin wat snel voor 1 concept hadden gekozen, vond ik het op dat moment een manier om dit recht te zetten, en het algemene probleem zoveel mogelijk uit te werken in alle facetten.

Ik vond dit leerrijk, omdat er later nog van ons zal verwacht worden om wetenschappelijke teksten te schrijven, dus was het goed om wat ervaring op te doen.

Daarbuiten waren er ook technische concepten die voor mij nieuw waren, zoals het differentieelsysteem. Ook het duidelijke besef dat er in de realiteit veel meer problemen opduiken dan bij de eerste planning was een leerrijke ervaring. De stabiliteit van onze constructie dreigde naar het einde toe een groot probleem te worden. Gelukkig konden we dit gevaar mits enkele aanpassingen inperken. Maar ik denk dat we daardoor wel allemaal hebben ingezien dat stabiliteit een heel belangrijk woord is, en dat we er in ons verdere leven misschien toch net iets meer aandacht aan moeten besteden.

Het groepswerk had positieve en negatieve kanten. Voor mij persoonlijk was het niet echt nieuw om in groep te werken, doordat ik al eerder een diploma behaald heb aan het conservatorium. Toch was het werken in deze groep en aan dit project een heel nieuwe ervaring. Ook het feit van met 7 mensen aan 1 verslag te werken, en toch proberen de coherentie te bewaren, bleek helemaal niet evident.

Het was duidelijk dat dit project ons al een goede duw in de richting van het ingenieur worden heeft gegeven, maar ook een aanwijzing dat we nog niet aan de eindstreep zijn.

Avdyl Shala



Als ik terugkijk naar de voorbije weken dan zie ik dat ik eigenlijk veel nieuws heb meegemaakt en geleerd. Qua techniek, teamwork en mening.

Direct al in het begin zag ik eigenlijk dat er veel nagedacht moest worden vóór het bouwen zelf. Ik weet zeker dat als we de legoblokjes al kregen dat er sowieso gebouwd zou worden diezelfde dag. Maar als ik nu terugdenk, dan is het dus beter dat we deze niet gekregen hadden aangezien er eigenlijk heel veel nagedacht moest worden over het idee zelf. Het klonk als een relatief simpel opdracht, pak het balletje op en doe het in het mandje, maar toen ik er verder over nadacht, zag ik dat het eigenlijk hélemaal niet zo simpel is. Er zit veel meer achter, dus het was echt wel een goed ding dat we lang gediscussieerd hebben voor het bouwen zelf. Dit is zeker iets waar ik rekening mee zal houden. Dag 2 en 3 waren dagen waarop ik afwezig was vanwege een ziekte. Dit heeft eigenlijk de grootste impact gehad op mij de voorbije weken. Iedereen begon al te bouwen aan een zeker stuk en iedereen was direct druk bezig bij mijn terugkomen, dus ik moest heel veel inhalen. Ik probeerde alles in te halen, maar ik heb een grote fout gemaakt: ik moeide mij niet genoeg in de zaak. Ik rekende teveel op de coördinator om mij taken te geven en te zeggen wat ik moest doen en wie ik moest helpen, terwijl dit eigenlijk helemaal niet de bedoeling was. Hierdoor heb ik mijn achterstand eigenlijk vergroot. Uiteindelijk had ik het wel door en probeerde ik eerst rond te vragen wat iedereen aan het doen was en wat al gedaan was, ik kreeg hierbij steun van al mijn teamgenoten. Ik heb dag 4 doorgebracht met het inhalen van mijn achterstand en het helpen met kleine taakjes. De voornaamste les die ik hieruit geleerd heb is dat ik assertiever moet zijn en sneller initiatief nemen, ongeacht welke toestand. Ik heb ook veel nieuwe technieken geleerd. Het was misschien uit lego, maar de hoeveelheid combinaties die mogelijk zijn met tandwielen zijn verbazingwekkend. Vooral de differentiaal heeft mij verbaasd. Hoe het mogelijk was met 1 motor 2 verschillende bewegingen te doen, in een zekere volgorde dan wel heeft mijn ogen wel opengemaakt voor technieken. Ik heb nu ook gezien dat starre lichamen in de realiteit zeker niet bestaan en hoe sommige constructies zoals de driehoeksconstructie zeer handig zijn om gewicht op te leggen. Het meest belangrijke dat ik geleerd heb is dat een planning

ongelooflijk moeilijk is om na te komen. Er moet altijd een foutmarge zitten in de tijd die wordt gegeven voor een zekere taak. Het is bijna onmogelijk om aan te nemen dat een opdracht een specifieke hoeveelheid tijd nodig heeft om te slagen, 99% van de gevallen klopt dit gewoon niet. Vanaf nu af aan zal ik daar sowieso constant rekening mee houden. Ik weet zeker dat dit een zeer handige les is voor de toekomst. Nog iets waar ik rekening mee moet houden is dat bij zo'n project ik afhankelijk ben voor 6 andere personen. Ik was vergeten dat wat ik doe ook voor hun geldt. Dit is mij duidelijk gemaakt door mijn teamlid Maarten Weckx. Ik moest in het weekend werken aan een bepaald stuk van het verslag, maar heb mijn tijd onverwacht moeten opgeven aan een andere, in mijn ogen, belangrijkere taak voor mijn familie. Waar ik geen rekening mee gehouden had, was dat er 6 andere personen waren die wel afhankelijk waren van dat deel. Ik zie in dat wat ik gedaan had wel egoïstisch is en dat ik in het vervolg zeker rekening moet houden met zo'n gevallen. Ik moet beter prioriteiten kunnen stellen. Qua teamwork moet ik in de toekomst meer aandacht schenken aan zoiets. Sommige dingen moeten gewoon wanneer ze moeten. Zeker als er mensen op mij rekenen.

Ik kan eerlijk zeggen, ik heb veel geleerd de voorbije weken en zal zeker mijn best doen dit in de toekomst zo goed mogelijk te gebruiken. Ik dank mijn teamleden voor al hun hulp, mijn coördinator voor zijn geduld, en de assistenten voor hun tijd en mening.

Pieter Ruelens



Na zes weken (en een paar nachten) werken is het project eindelijk af. Het was een tof project op zich hoewel het volgens mij geen goed beeld geeft van de richting Werktuigkunde-Elektrotechniek.

Dit wil niet zeggen dat ik geen nieuwe dingen geleerd heb tijdens het project, meer zelfs, ik heb vooral zaken geleerd die ik overal kan gebruiken, niet alleen in de richting WE.

Zo heb ik gemerkt dat een planning opstellen niet alleen noodzakelijk is, maar ook veel moeilijker als initieel gedacht. In de planning moet rekening gehouden worden met het feit dat er sowieso wel onderdelen van het project vertraging oplopen waardoor heel de planning in de war gestuurd is en de kans reëel wordt dat de deadline niet gehaald wordt.

Concepten uitdenken is ook veel belangrijker gebleken dan gedacht, zo kwam het toch iets te vaak voor dat we een onderdeel compleet moesten herbouwen omdat er met iets geen rekening gehouden is. Als we het concept verder hadden uitgedacht zou dit niet nodig zijn geweest.

Het was ook de eerste keer dat we in zo'n grote groep moesten werken, vroeger waren groepswerken met maximum 3 man, nu met 7. Dit bracht meer moeilijkheden met zich mee dan verwacht, ook in verband met taakverdeling, soms stonden mensen niets te doen, gewoon omdat ze het werk alleen maar zouden hinderen, men kan immers niet met 5 man aan 1 pc werken.

Voor mij was dit project dus een zeer leerrijke en plezierige ervaring, ook al heb ik niet echt het gevoel gehad dat ik met een project werktuigkunde-elektrotechniek bezig was.

Zjef Van De Poel



Na zes weken werken aan dit project, waarvan de laatste zeer intensief, is het zeer aangenaam om eindelijk een werkend resultaat te kunnen aanschouwen. Dit is een excellente beloning voor het harde werk geïnvesteerd in de robot.

Echter, een spijtige zaak is dat er weinig concrete, theoretische kennis of praktisch, bruikbare vaardigheden zijn opgedaan tijdens dit technologieproject, dit in tegenstelling tot bijvoorbeeld het project rond EIT, waarbij men de mogelijkheid heeft om over een analoog systeem te leren en dit te bouwen voor de regeling van een watertoren.

Ook lijkt soms de link tussen het bouwen van een robot, met behulp van lego, en de richting Werktuigkunde-Elektrotechniek ver te zoeken.

Dit wil natuurlijk niet zeggen dat er helemaal niets bijgeleerd is, en alles weggegooide tijd is geweest. Werken aan een project in teamverband kan op eerste zicht makkelijker lijken dan het in werkelijkheid is. Dit project heeft duidelijk geleerd dat om alles in goede banen te leiden, er ten eerste een goede planning nodig is, maar het meest belangrijke toch een doordachte conceptuitwerking nodig is, en op voorhand, voor zover mogelijk, problemen te voorzien.

Verder is men tijdens de programmatie van de besturing voor de robot in contact gekomen met de programmeertaal C#, een tool die op verscheidene plaatsen van nut kan zijn.

Al bij al een interessant project, dat misschien iets beter bij de richting zou mogen aansluiten.

Maarten Weckx



Na zes weken van frustraties, gelach, gevloek en euforie is het project werktuigkunde-elektrotechniek op zijn einde en zijn we er in geslaagd een werkende robot te bouwen die in staat is ballen op te rapen en te vervoeren. Het was een leuk project, dat volgens mij niet echt een totaal beeld gaf van de desbetreffende afstudeerrichting. Ik heb echter veel bijgeleerd.

We zijn volledig verkeerd gestart aan het project. We wouden te snel overgaan op het werkelijk bouwen van de robot en hebben daardoor in het begin tijd verloren door te prutsen zonder voldoende na te denken. Naarmate de weken vorderden begonnen we dit te beseffen en schakelden we stilaan over op een betere werkwijze. Moest ik nu aan een dergelijk project beginnen, zou ik gestructureerder te werk gaan. Ik zou beginnen met alle mogelijkheden en probleemstellingen op een rijtje te zetten en één voor één grondig te bespreken. Om vervolgens hierop te baseren bij het begin van de bouw. Dit zou het schrijven van het eindverslag tevens bevorderen. Verder heb ik geleerd dat bij vele problemen meer komt kijken dan op het eerste zicht lijkt. Bij onze constructie moesten er namelijk bijna constant verbeteringen worden aangebracht om stabiliteit te verbeteren, systemen te verbeteren en het doorschieten van tandwielen te verhinderen. Elk onderdeel is ook minstens twee maal herbouwd geweest.

Ik had het geluk in een groep te zitten die bestond uit zeer leuke mensen die elk op hun eigen manier een steentje, tot vaak een grote steen, bijdroegen tot de uitwerking van het project en de fantastische groepssfeer. Ondanks dat we met een mankracht minder waren en gedurende twee weken zelfs met twee, vind ik dat het eindresultaat, waar we als groep voor hebben gewerkt, er dubbel en dik mag zijn. Uiteraard hingen er, zeker naar het einde toe, wat spanningen in de lucht. Ze waren vaak gegrond, maar daar kan ook uit worden geleerd.

Als besluit wil ik de assistenten en proffen bedanken voor hun begeleiding en advies. De mensen van het atelier, omdat ze altijd klaar stonden, en in het bijzonder Jean-Paul.

Hannes Maes



Na zes weken zwoegen is het nu eindelijk zover dat ik aan deze reflectie kan beginnen.

Om te beginnen wil ik zeggen dat ik, vijf minuten voordat de vraag gesteld werd, er nog niet aan gedacht had om coördinator te worden. Op het moment dat er een coördinator gekozen moest worden was ik wel nieuwsgierig wat dat inhield en besloot ik mij dus maar kandidaat te stellen, al doende leert men nu eenmaal. De bedoeling, zoals deze in het begeleidend document vermeld wordt, was eigenlijk niet dat de coördinator de delegerder zou zijn maar al snel werd duidelijk dat dit in onze groep toch wel meer die kant op ging. Vanaf het eerste moment werd er op mij gerekend voor het verdelen van de taken en voor het bewaren van het overzicht. Het nadeel hiervan was dat sommige leden van mijn team hierdoor sneller de neiging hadden om zich minder aan te trekken van het totaaloverzicht wat het voor mij allemaal niet vereenvoudigde.

Dit behouden van het overzicht is niet zo eenvoudig als het klinkt heb ik geleerd. In het begin bestond dat uit op een kladpapier kribbelen wat er nog moest gebeuren, hierna evolueerde dit al naar het maken van meer gestructureerde To Do –lijstjes maar door deze elk uur opnieuw te vernieuwen werd ook dit al snel een chaos. Naar het einde toe merkte ik dat de beste aanpak voor mij het optimaal behouden van je concentratie en rust is. Uit deze manier komen in mijn geval de vlotste resultaten voort.

Opmerkelijk aan dit project was ook wel de variërende groep, hierin zaten personen die praktisch constant in het oog moesten gehouden worden maar er zaten eveneens personen tussen die zonder mijn niet veel slechtere resultaten zouden hebben afgeleverd. Dit maakte het bewaren van het overzicht moeilijker maar op zijn eigen manier ook interessanter. Naar het einde toe relateerde ik met elk specifiek probleem een persoon waardoor de afloop toch redelijk vlot verlopen is.

Van mijzelf vind ik dat ik het redelijk goed heb gedaan als coördinator, misschien dat dit zonder de gewoontelijke vrijdagse vermoeidheid beter zou zijn geweest maar dat zien we de volgende keer wel. De communicatie met de coördinatoren verliep volgens mij vlot, ik deinsde er niet voor terug om hun advies te vragen maar ook niet om hun advies in vraag te stellen indien nodig.

Naar mijn team toe denk dat ik meestal zo diplomatisch mogelijk ben gebleven, ook in tijden van tijdsnood of een mislopende planning. Ik ken ze ondertussen allemaal een stuk beter en heb eigenlijk op een geval na op het einde nooit spanningen gevoeld. Ik ben ook nooit de confrontatie uit de weg gegaan wanneer iemand het niet akkoord met mij was. Volgens mij heb ik telkens wanneer het

ontwikkelingsproces fout dreigde te lopen op tijd ingegrepen behalve in het begin maar ook dit is te nuanceren. Na de eerste werkdag had ik overlegd met de begeleiders en ben ik tot de conclusie gekomen dat het bedenken van concepten op dat moment het belangrijkste was. Ik heb dit toen voorgelegd aan mijn team en heb een vergadering met 4 van de op dat moment 6 aanwezige leden georganiseerd, de andere twee vonden dit op dat moment niet relevant en ik heb hen dus niet gedwongen om aan deze bespreking deel te nemen. Al snel werd duidelijk dat de andere 3 medewerkers het nut van deze bespreking ook niet zo hoog inschatten dus heb ik na enkele uurtjes gezegd dat het verstandiger zou zijn om door te werken. Een bespreking waarin drie van de vier mensen zitten te zuchten en zeggen dat het zo snel mogelijk klaar moet zijn is namelijk geen bespreking.

Op democratische wijze werd er al na anderhalve dag beslist om definitief voor het kraanmodel te gaan. Tot op het moment dat ik dit schrijf weet ik niet in hoeverre ik toen had kunnen of moeten ingrijpen. Omdat ik op dat moment, en nu nog steeds, ook geloofde in de kraan ben ik hierbij meegegaan. Moest dit een model zijn geweest wat ik volledig onverstandig had gevonden dan had ik hier wel kunnen ingrijpen maar in dat geval zou ik ook betere argumenten hebben kunnen bedenken. Nu kwam er het tegenargument: 'Het is toch duidelijk dat de kraan efficiënter is, enkel al op basis van de navigatie, waarom zouden we dan nu onze gezamenlijke werktijd verspillen aan het bespreken van iets wat we ook achteraf kunnen doen'.

Kort samengevat: het belangrijkste dat ik geleerd heb aan de laatste zes weken en waar ik het meest in gegroeid ben is het samenwerken met en afstemmen van een groep toch wel uiteenlopende mensen. Ook op het vlak van analytisch denken en snel beslissingen maken heb ik veel bijgeleerd.

Analytische geest	4		4
Probleemoplosser	3	Op dit vlak ben ik veel sneller geworden	4
Creatieve geest	3		3
Beslisser	4	Ook hierin ben ik sneller en kordater geworden	4.5
Doener	3.5		3.5
Organisator	3.5	Op dit vlak heb ik het meest bijgeleerd	4.5
Praktische kennis, kunde of ervaring	3	Ervaring op vlak van werken met een groep	4

Adam Cooman



Het is nu 5 uur 's morgens en het verslag is eindelijk af. Deze week is een van de meest stresserende van de laatste 3 jaar geweest, maar het was de moeite. We hebben keihard gewerkt aan onze robot en het verslag en het resultaat mag er wel zijn. Het was voor mij het eerste project waar we op deze manier in groep moesten samenwerken en daar heb ik ook het meest uit geleerd. Ik heb kunnen zien hoe het werken in groep het beste uit mensen naar boven kan halen, ik heb geleerd hoe de sterke en zwakke punten van de groepsleden gecombineerd werden tot een geoliede machine en ik heb ook geleerd hoe de machine niet meer goed werkt als er iemand niet zijn beste kant laat zien.

Ik heb meer inzicht gekregen in hoe het ingenieurswerk er in het echt aan toe gaat, ik heb geleerd dat het inderdaad goed is om eens na te denken over wat je aan het doen bent. Op vlak van technische kennis viel dit project wel mee, ik heb wel een paar mooie nieuwe tandwielschakelingen leren kennen maar hetgeen waar ik op technisch vlak het meest van geleerd heb is de rondleiding die we kregen en het contact dat we hadden met de mensen die in gebouw Z werken.

Als ik terugkijk naar mijn eigenanalyse van een paar weken geleden blijkt dat ik mij toen redelijk goed ingeschat had ik zal nu de verschillende klassen nog eens afgaan:

Analytische geest	3		3,5
Probleemoplosser	3	Hier ben ik beter in geworden dan ik er voor was	4
Creatieve geest	4	Dit was ook correct	4
Beslisser	3		3
Doener	5	Dit bleek zeer correct te zijn	5
Organisator	2	Hier ben ik ook een klein beetje verbeterd, er is nog ruimte voor verbetering	3
Praktische kennis, kunde of ervaring	2	Dit is nu iets verbeterd	3

Al bij al vond ik het een zeer geslaagd project en ik kijk al uit naar de volgende projecten.

Bijlagen

Bijlage 1: Vakwerken

Een vakwerk (ruimtelijk of vlak) wordt gekenmerkt door zijn bijzondere geometrische eigenschappen alsook door de specifieke manier waarop de belasting aangrijpt. Een vakwerk wordt opgebouwd uit staven die zodanig verbonden zijn dat ze een onvervormbaar of star geheel vormen. Een knooppunt is het punt waar de staven elkaar ontmoeten. In de berekening neemt men aan dat de staven in de knooppunten wrijvingsloos en scharnierend met elkaar verbonden zijn. Vakwerken zijn opgebouwd uit driehoeken om de eenvoudige reden dat een rechthoek of vierkant, gemaakt uit scharnierende staven, niet vormvast is; het is een mechanisme. Een driehoek is dit wel, vandaar ook dat men soms spreekt van driehoeksvakwerken. Dat de eindpunten van elke staaf scharnierend uitgevoerd worden impliceert onmiddellijk dat de staven enkel op trek of druk kunnen belast worden en niet op buiging.



De meeste materialen zijn het efficiëntst in trek en druk. Hierdoor kan men een zeer lichte constructie bouwen omdat er niet veel materiaal nodig is om zeer stevige constructies te bouwen.

Bijlage 2: Onrealistische grijpsystemen

Zuignap

Een zuignap is in staat ballen stevig te klemmen en vast te houden tijdens het vervoeren, ondanks schokken en trillingen. Het is wel noodzakelijk om de robot perfect te positioneren, zodanig dat de zuignap zich recht boven een bal bevindt, kleine afwijkingen kunnen er immers voor zorgen dat de zuignap niet volledig afgesloten wordt en niet vacuüm kan trekken. De zuignap kan als oplossing wel met een bepaalde kracht op de ballen worden gedrukt om volledig afsluiten te verzekeren, maar de ballen worden nogal makkelijk ingedrukt.

Het is echter moeilijk om een systeem te bouwen dat een zuignap vacuüm kan trekken, bovendien moet dan op zoek gaan naar een zuignap die mooi op de ballen past.

Besluit:

- In staat stevig te klemmen
- Geen speling op positioneren, dit moet praktisch perfect gebeuren
- Moeilijk te construeren

Werparm

Met een werparm bedoelt men een arm die in staat is een bal op te scheppen of op te pakken en weg te werpen. Hiermee kunnen we tijd winnen, want als de robot in staat is ballen in de goot te werpen, moeten de robot niet telkens terug naar de goot rijden om de gevangen ballen te lozen. Het is echter zeer moeilijk om het werpen precies genoeg te maken.

Besluit:

- Zeer grote precisie vereist

Het vervoeren en optillen van meerdere ballen tegelijk

Het spreekt voor zich dat het de efficiëntie ten goede komt een systeem te bedenken dat meerdere ballen tegelijk naar het doel kan brengen. Een nadeel is echter dat men hierbij in veel gevallen grof tewerk gaat en dat de andere robot of de ballen zouden kunnen beschadigd worden.

Toch zullen we de mogelijkheden bespreken waarbij zoveel mogelijk ballen tegelijk naar het doel worden gebracht. De ontwerpen die weinig verfijnde handelingen uitvoeren, zullen, door de onnauwkeurigheid, tenzij in uitzonderlijke gevallen (indien bijvoorbeeld het spel zou doorgaan zonder tegenstander) niet als uiteindelijk concept verkozen worden, maar ze zullen hier toch besproken worden, om de volledigheid van de studie te verzekeren.

In het geval het niet mogelijk is om meerdere ballen te grijpen, kan men nog proberen om ze toch samen te vervoeren. Daarover meer in de paragraaf 'Opslagsystemen'.

Des te meer ballen men tegelijk wil overbrengen, des te groter de gripper en/of de ganse machine moeten zijn. In dat geval heeft het minder nut om de sensoren te gebruiken voor het detecteren van ballen, en kan men beter op een systematische manier het veld afgaan. In sommige gevallen zal het mogelijk zijn om in slechts enkele 'happen' alle ballen mee te nemen.

Het bouwen van een grote gripper of een grote machine impliceert ook dat er meer materiaal moet gebruikt worden. Hiervoor zal men dan andere materialen aanwenden, omdat het moeilijker zal zijn om alles met Lego te realiseren.

Een grote machine vereist ook een stevig onderstel om stabiel te kunnen blijven. Hierbij kan het een optie zijn om een kraanachtige robot te bouwen, waarvan het onderstel zich eigenlijk buiten het speelveld bevindt, zodat dit niet in de weg staat bij het grijpen van de ballen.

We kunnen nog het volgend onderscheid maken bij de bespreking van de methoden:

- I. de ballen worden meteen opgetild en in de hoogte vervoerd
- II. de ballen worden voortgesleept/-geduwd en op het einde over de rand getild.

Uitwerking:

I. Optillen en in de hoogte vervoeren

Voor het grijpen van ballen kan men een vorklift of een gripper gebruiken. Voor meerdere ballen kunnen hierop varianten bedacht worden:

- een vorklift kan meerdere tanden hebben, verspreid over een grotere afstand
- een gripper kan veel breder gemaakt worden en de ballen die op eenzelfde lijn liggen tegelijk insluiten.

II. Slepen/duwen, op het einde optillen

Een mogelijkheid zou zijn om met een kraan een eind naar voor te rijden, daar een plaat te laten zakken, en dan terug te rijden, zodat alle voorwerpen bij elkaar geveegd worden die zich achter de plaat bevinden. Deze plaat kan met behulp van een hoekvorm (zoals een boekensteun) onder de ballen

schuiven. Als de robot terug aan zijn eigen doel gekomen is, wordt de plaat terug naar boven gehaald en eventueel licht gekanteld, zodat de ballen in de goot terecht komen.

Hierbij heeft men geen enkele zekerheid dat er enkel ballen vervoerd worden. De andere robot kan zich voor de plaat bevinden en ook meegesleurd worden. Dit kan tot beschadiging leiden en is dus tegen de spelregels. Omwille van deze reden zijn deze methodes niet aan te raden voor de uiteindelijke keuze van het ontwerp.

Bijlage 3: Source code van het programma

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using Bram.NxtSharp;

//motor port 1: forward/backward
//motor port 2: sideways
//motor port 3: differential

//sensor port 1: lightsensor&laser
//sensor port 2: pressure sensors
//sensor port 3: light sensor in claw
//sensor port 4: ultrasonic sensor

//differential: close&up // open&down //first:down then up.

//motor power: positive if counter clockwise rotation (if rotating part of
the motor on the right)

namespace Robot
{
    public partial class Form1 : Form
    {
        private SensorEvent pressureevent;

        private int tries = 0;

        private int forwardpower=50; //power to move the motor forward
        private int sidepower = -80; //power needed to move the motor from
the original side to the other side
        private int downpower = 100; //power needed to move the arm down
        private int openpower = 100; //power needed to open claw to drop
ball

        static private int updegrees = 7800; //degrees needed to close claw
and move up
        static private int opendegrees = 1350; //degrees needed to open claw
        private int downdegrees = updegrees-opendegrees; //degrees needed
to move arm down (excluding opening)

        private int lightdefault; //light intensity on the sensor in open
field
        private int lightchange=11; //change in reading from the light
sensor when ball is on the line

```

```

        private int distanceball = 12; //a value lower than this means
there is a ball below the distance sensor

        private int correcttime = 1400; //time is ms to drive forward when
found ball.

        private int lightmin = 521; //value of the lightsensor when a ball
is in the claw

        private static NxtLightSensor lightsensor1,lightsensor2;
        private static NxtPressureSensor pressuresensor;
        private static NxtSonar distancesensor;

        private NxtCommunicator tacho;
        private static NxtGetOutputState state;

        private int sideposition;
        private int width = 2900; //width of the field in degrees

    public Form1()
    {
        InitializeComponent();

        lightsensor1 = (NxtLightSensor)nxtBrick1.Sensor1;
        distancesensor = (NxtSonar)nxtBrick1.Sensor4;
        lightsensor2 = (NxtLightSensor)nxtBrick1.Sensor3;
        pressuresensor = (NxtPressureSensor)nxtBrick1.Sensor2;
    }

    private void Accelarate(NxtMotor motor,int maxspeed)
    {
        for (int i = 3; i > 0; i--)
        {
            motor.Turn((int)(maxspeed / i), 0);
            Thread.Sleep(50);
        }
    }

    private void CorrectScanLine()
    {
nxtBrick1.MotorA.Turn((int)(forwardpower/Math.Abs(forwardpower)*25), 0);
        Thread.Sleep(correcttime);
        nxtBrick1.MotorA.Brake();
    }

    private int GetTacho(NxtMotor motor)
    {
        state = tacho.GetOutputState(motor.Port);
        return state.TachoCount;
    }

    private void Init()
    {
        // lightsensor environent intensity
        for (int i = 0; i < 4; i++)
        {
            lightsensor1.Poll();

```

```

        lightdefault += lightsensor1.Value;
    }
    lightdefault /= 4;

    //sideposition
    sideposition=GetTacho(nxtBrick1.MotorB);
}

private void button1_Click(object sender, EventArgs e) //connect
button
{
    try
    {
        tacho = nxtBrick1.Connect();
        Invoke((MethodInvoker)delegate()
        {
            labell1.Text = "Connected";
        });
    }
    catch
    {
        Invoke((MethodInvoker)delegate()
        {
            labell1.Text = "Connection failed";
        });
    }
}

private void button2_Click(object sender, EventArgs e) //Start
button
{
    Init();
    lightsensor1.Polled += new
Bram.NxtSharp.SensorEvent(LineDetected);
    distancesensor.Polled += new
Bram.NxtSharp.SensorEvent(FoundBall);
    pressureevent+= new SensorEvent(ArrivedatEnd);
    Forward();
}

private void Forward() // first step in finding the ball
{
    int i = 0;
    while (true)
    {
        nxtBrick1.MotorA.Turn(forwardpower,0);
        lightsensor1.Poll();
        i++;
    }
}

void LineDetected(Bram.NxtSharp.NxtSensor sensor)
{
    if (Math.Abs(lightdefault - lightsensor1.Value) >= lightchange)
//ball is on the line
    {
        tries++;
    }
}

```

```

        if (tries < 3)
        {
            nxtBrick1.MotorA.Brake();
            ScanLine();
        }
        else
        {
            BacktoStart();
        }
    }
    else //no ball on this line
    {
        lightsensor1.Poll();
    }
}

private void ScanLine() // Scan a line were a ball was found
{
    nxtBrick1.MotorB.Turn(sidepower,0);
    distancesensor.Poll();
}

void FoundBall(Bram.NxtSharp.NxtSensor sensor)
{
    if (distancesensor.RawValue <= distanceball) //there is a ball
below the distance sensor
    {
        nxtBrick1.MotorB.Brake();
        CorrectScanLine();
        CatchBall();
    }
    else
    {
        pressuresensor.Poll();
        if (pressuresensor.IsPressed)//didn't find ball on the
entire line
        {
            nxtBrick1.MotorB.Brake();
            Rotate(nxtBrick1.MotorB,
Math.Abs(GetTacho(nxtBrick1.MotorB) - sideposition - 200), -sidepower,
false);
        }
        else //no ball under the distance sensor
        {
            distancesensor.Poll();
        }
    }
}

private void Rotate(NxtMotor motor,int degrees,int power,bool
enforce) //the normal .turn method didn't always work
{
    int startdist = GetTacho(motor);
    int i = 0;
    int prevtacho=0;

    motor.Turn(power,0);

```

```

while (Math.Abs(GetTacho(motor) - startdist)<=degrees)
{
    if (!enforce)
    {
        if (prevtacho == GetTacho(motor))
        {
            i++;
        }
        else
        {
            i = 0;
        }
        prevtacho = GetTacho(motor);

        if (i == 7)
        {
            break;
        }
    }
}

private bool BallInClaw() //test with lightsensor if there is a
ball in the claw
{
    int lightvalue = 0;

    for (int i=0;i<3;i++)
    {
        lightsensor2.Poll();
        lightvalue += lightsensor2.RawValue;
    }
    lightvalue /= 3;

    return (lightvalue > lightmin);
}

private void CatchBall()
{
    Rotate(nxtBrick1.MotorC,downdegrees,downpower,true);
    nxtBrick1.MotorC.Turn(-downpower, updegrees);
    WaitWhileRunning(nxtBrick1.MotorC);
    Rotate(nxtBrick1.MotorB,
Math.Abs(GetTacho(nxtBrick1.MotorB) - sideposition-200), -sidepower,false);

    if (BallInClaw())
    {
        BacktoStart();
    }
    else
    {
        DropBall();
    }
}

private void BacktoStart() //drive back when caught ball

```



```

    {
        pressuresensor.Polled += pressureevent;
        nxtBrick1.MotorA.Turn(-forwardpower, 0);
        pressuresensor.Poll();
    }

void ArrivedatEnd(NxtSensor sensor)
{
    if (pressuresensor.IsPressed) //arrived at end
    {
        pressuresensor.Polled -= pressureevent;
        Thread.Sleep(300); //make sure both ends of the robot are
touching the end of the table
        nxtBrick1.MotorA.Brake();

        if (tries < 3)
        {
            DropBall();
        }
        tries = 0;
    }
    else
    {
        pressuresensor.Poll();
    }
}

private void DropBall()
{
    Rotate(nxtBrick1.MotorC, opendegrees, openpower, true);
}

private void WaitWhileRunning(NxtMotor motor)
{
    bool running = true;
    int tacho,prevtacho=0;

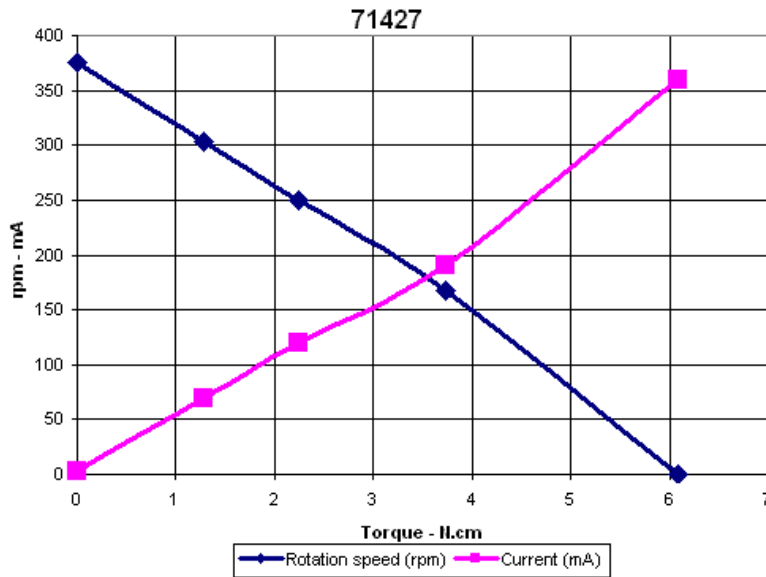
    while (running)
    {
        tacho = GetTacho(motor);
        Thread.Sleep(300);
        if (tacho == prevtacho)
        {
            running = false;
        }
        prevtacho = tacho;
    }
}

private void Form1_FormClosing(object sender, FormClosingEventArgs
e)
{
    nxtBrick1.MotorA.Brake();
    nxtBrick1.MotorB.Brake();
    nxtBrick1.MotorC.Brake();
    nxtBrick1.Disconnect();
}

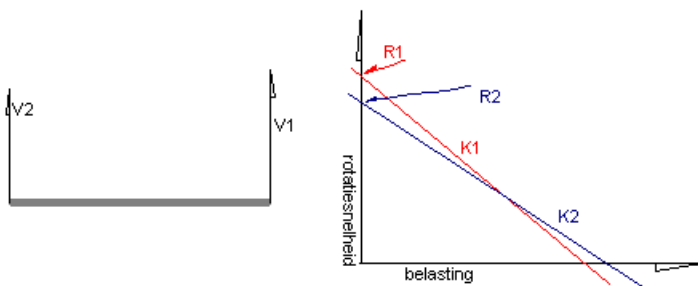
```

Bijlage 4: Berekening scheeftrekken

De motoren die we gebruiken hebben de volgende snelheid/belasting curve:



De kraan trekt scheef als de snelheid van de verschillende wielen verschillend is. We veronderstellen dat de snelheid waarmee de motor draait lineair is met de belasting.



$$V_1 = -(K_1 * T) + R_1$$

$$V_2 = -(K_2 * T) + R_2$$

$$V_1 = V_2$$

$$\Rightarrow -(K_1 * T_1) + R_1 = -(K_2 * T) + R_2$$

$$\Rightarrow T_1 = \frac{-(K_2 * T) + R_2 - R_1}{-K_1}$$

Als we lineaire wrijving beschouwen is de belasting lineair met de normaalkracht op de wielen.

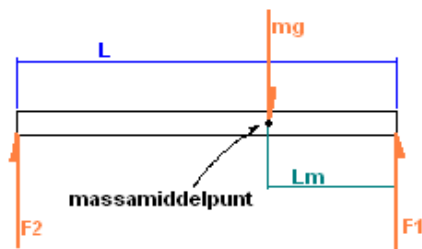
$$T = f(F_N)$$

f is de functie die bepaald wordt door de tandwieloverbrenging tussen de motor en de wielen.

We veronderstellen voor de gemakkelijheid dat f gewoon gelijk is aan 1. het verband dat dan tussen de normaalkrachten moet optreden zodat de snelheden gelijk zijn is:

$$F_1 = \frac{-(K_2 * F_2) + R_2 - R_1}{-K_1}$$

Als het massamiddelpunt op een bepaalde plaats op de kraan ligt kunnen we normaalkrachten bepalen:



$$\begin{cases} F_2 + F_1 - mg = 0 \\ L_m * mg - L * F_2 = 0 \end{cases} \Rightarrow \begin{cases} F_2 = (L_m/L) * mg \\ F_1 = (1 - L_m/L) * mg \end{cases}$$

Als we deze betrekkingen invullen in het verband tussen de krachten vinden we volgende uitdrukking voor de locatie van het massamiddelpunt:

$$\frac{L_m}{L} = \frac{K_1 * mg + R_2 - R_1}{mg * (K_2 + K_1)}$$

Als we hierin veronderstellen dat beide motoren hetzelfde zijn ($K_1=K_2$ en $R_1=R_2$) dan vinden we dat het massamiddelpunt in het midden moet liggen.

Bijlage 5: Tandwielen

We weten dat de transmissieverhouding van tandwiel 1 met een hoeksnelheid W_1 op tandwiel 2 met hoeksnelheid W_2 gelijk is aan:

$$i = Z_1/Z_2 = W_2/W_1$$

Uit het vermogen:

$$P = T \cdot W = F \cdot C_{te} \cdot W = C \cdot W$$

waarbij T de torsie, F de kracht en C een koppel. Bij de kennis van C_{motor} is men in staat om $C_{tandwiel}$ te vinden dat dus de koppel is van het tandwiel dat aan het motor hangt als men de verhouding van hun vermogens neemt en warmteontwikkeling verwaarloost:

$$P_{motor}/P_{tandwiel} = 1 = (C_{motor}/C_{tandwiel}) * (Z_{tandwiel}/Z_{motor})$$

waarbij de C_{motor} , Z_{in} , Z_{uit} gekend zijn en we dus hieruit $C_{tandwiel}$ vinden.

Hieronder wordt uitgelegd hoe wij de tandwielen gebruikt hebben in onze kraan; ingedeeld in 3 delen: de rijdende kar op het metaal, de kar aan de rand en de toren met de grijparm. Voor verdere uitleg over hoe tandwielen in mekaar zitten en werken zie het onderdeel over materiaal.

Rijdende kar op metaal:

De tandwielen die we gebruikt hebben op de rijdende kar zijn de volgende in volgorde: De motor is via een staafje verbonden met een tandwiel met 40 tanden, hierop zit een tandwiel met 24 tanden die zelf verbonden is met een tandwiel met 8 tanden en tegelijk ook met een staafje die een wiel aandrijft. Het tandwiel met 8 tanden is terug verbonden met een van 24 tanden die via een staafje terug verbonden is met een wiel. Indien we de transmissieverhoudingen afgaan:

$$i = (40/24) * (24/8) * (8/24) = 5/3$$

zien we dat per 3 omwentelingen van de motor, de wielen zelf 5 keer draaien wat dus inhoudt dat we winst hebben qua snelheid maar met een verlies qua koppel aangezien $Z_{tandwiel}/Z_{motor}$ kleiner is dan 1 en groter dan 0 dus als we

$$P_{motor}/P_{tandwiel} = 1$$

nemen dan is:

$$C_{\text{tandwiel}} = C_{\text{motor}} * (Z_{\text{tandwiel}}/Z_{\text{motor}})$$

en het gestelde volgt.

Kar aan de rand:

De motor is via een staafje verbonden aan een tandwiel met 16 tanden, die rechtstreeks met een verbonden is met 40 tandwielen en deze vervolgens met een tandwiel met 24 tanden. Deze laatste is via een staafje verbonden met een wiel en drijft deze dus aan, en vervolgens hangt hier een ketting op die met een ander tandwiel verbonden is van ook 24 tanden die eenzelfde wiel aandrijft op de wagen. Indien de motor 1 omwenteling aflegt dan leggen de 2 wielen $2/3$ van hun omwenteling af:

$$i = (16/40) * (40/24) = 2/3$$

Dan is:

$$C_{\text{tandwiel}} = C_{\text{motor}} * (Z_{\text{tandwiel}}/Z_{\text{motor}})$$

Met $Z_{\text{tandwiel}}/Z_{\text{motor}}$ groter dan 1 vindt men dus een winst in koppel, indien we aannemen dat de verhouding van de afgeleverd vermogen ten opzichte van de uitwendig verkregen vermogen gelijk is aan 1. Dus geen warmte-ontwikkeling.

Toren met grijp-arm:

Met de motor op een staaf verbonden zit een tandwiel met 12 tanden, die nog met zo een tandwiel is verbonden. Vervolgens bevindt er zich een tandwiel met 8 tanden op dezelfde staaf als de laatste tandwiel met 12 tanden. Deze 8 tanden drijven de buitenste 24 tanden aan van de differentiaal. Vanaf hier wordt het systeem opgesplitst in 2 verschillende wegen, een om de grijp-arm te besturen en een om naar boven en beneden te gaan:

De berekeningen dusver zijn de volgende:

$$(12/12) * (8/24) = 1/3$$

dus per 1 omwenteling van de motor doet de laatste tandwiel $1/3$ van zijn rotatie.

Naar boven en beneden:

De binnenste tandwiel van de differentiaal met 12 tanden is via een staaf verbonden met nog een tandwiel van 12 tanden die vervolgens een tandwiel aandrijft van 20 tanden en die dan verbonden is met een tandwiel van 36 tanden. Deze tandwiel bevindt zich op een staaf met 2 andere tandwielen met 20 tanden die zich optrekken en laten gaan op een legoblok met tanden zelf.

$$i = (1/3) * (12/20) * (20/36) = 1/9$$

Dus het is duidelijk dat per 1 omwenteling van de motor, de uiteindelijke tandwielen 1/9 van de omwenteling zullen afgelegd hebben, met een winst in koppel. (Uitleg is analoog aan “Kar aan de rand” hierboven.)

De grijparmen:

De tweede binnenste tandwiel van de differentiaal met 12 tanden is ook verbonden via een staaf aan een tandwiel met 24 tanden. Deze is dan rechtstreeks verbonden met een tandwiel met 40 tanden die op zich weer met een van 24 tanden is verbonden. Op eenzelfde staaf is met deze 24 tanden een tandwiel met 12 tanden verbonden. Hierop staat loodrecht nog een tandwiel met 12 tanden die een lange staaf aandrijft over de volledige lengte van de arm. Aan de uiteinde is nog een tandwiel met 12 tanden die wederom loodrecht een tandwiel van 12 tanden aandrijft. Deze is tenslotte verbonden aan een tandwiel met 24 tanden maar dan wel via een staaf. Deze laatste is dan weer verbonden met een tandwiel van 40 tanden die het eerste deel van de arm bestuurt (dus open of dicht) en aan deze 40 tanden is nog een tandwiel met 40 tanden die het ander deel van de arm bestuurt. Zo bekomt men dus de opstelling die wij gebruikt hebben.

$$i = (1/3) * (3/5) * (5/3) * (12/12)^2 * (12/24) * (24/40) = 1/10$$

Dus per omwenteling van de motor doet de laatste tandwiel 1/10 van zijn omwenteling. We hebben wederom een winst in koppel. (Uitleg is analoog aan “Kar aan de rand” hierboven.)