



Vrije Universiteit Brussel
Faculteit Toegepaste Wetenschappen
Vakgroep Werktuigkunde

Quasi-statische controle van een stappende
robot aangedreven door actuatoren met
regelbare stijfheid

Bram Vanderborght

Proefschrift ingediend tot het behalen van de academische graad van
Burgerlijk Werktuigkundig-Electrotechnisch ingenieur

Academiejaar 2002-2003

Promotor: Prof. Dr. ir. Dirk Lefebvre

Dankwoord

Om vooruitgang te boeken op wetenschappelijk vlak volstaat het niet meer om alleen te strijden, maar moet men met een team nieuwe horizonten afspeuren.

Ik zie het dan ook als één van de grote pluspunten van mijn thesis dat ik een stukje van het doctoraat heb mogen afwerken van Björn Verrelst en Ronald Van Ham.

Hen ben ik dan ook de grootste dank verschuldigd.

Een bijzonder woord van dank gaat uit naar Prof. Dirk Lefeber voor zijn goede opvolging en ondersteuning van het project.

Prof. Frank Daerden, Michaël Vandamme, Jimmy Vermeulen en Joeri Barbarein zou ik willen bedanken voor hun suggesties en tips die mij telkens wat dichter bij het doel brachten.

Jean-Paul Schepens, Thierry Lenoir en André Plasschaert voor hun technische hulp.

Mijn ouders en vrienden voor de morele steun.

Samenvattingen

2.1 Quasi-statische controle van een stappende robot aangedreven door actuatoren met regelbare stijfheid

Deze thesis behandelt de ontwikkeling van een tweepotige stappende robot geactueerd met geplooid pneumatische artificiële spieren. De connecties van de zeven geledingen, zijnde voeten, onderbenen, bovenbenen en bovenlichaam, worden verzekerd door zes gewrichten, dit zijn per been de enkel, knie en heup. De bewegingsruimte van de robot is beperkt tot het sagitaal vlak.

De controle van de robot gebeurt op twee niveaus. Een controle op globaal niveau -waargenomen door de computer die de jointtrajecten voor de geledingen genereert- en een controle op lokaal niveau voor elk gewricht -waargenomen door de microcontroller. Een snelle communicatie tussen beide is dan ook noodzakelijk.

De trajectgeneratie gebeurt aan de hand van objectieve parameters, de lokale controller is voorlopig een PID regelaar waarvan de acties variëren naargelang het traject.

De robot is momenteel in staat om op een quasi-statische manier stappen te zetten, hetgeen het doel was van de thesis. De volgende stap in het onderzoek is controlestrategieën te ontwikkelen die, gebruik makend van de adaptieve stijfheid, de robot in staat stelt om op een dynamische wijze te stappen.

2.2 Quasi-static control of a walking robot powered by actuators with adaptable stiffness

This thesis reports on the development of a bipedal walking robot powered by Pleated Pneumatic Artificial Muscles. The connections of the seven segments (feet, lower legs, upper legs and upper body), are ensured by six joints, these are for each leg the ankle, knee and hip. The movement of the robot is restricted to the sagittal plane.

The control of the robot is applied at two levels. A control on high level -observed through the computer which produces the trajectories of the different joints- and a control on local level for each joint -observed through the micro controller. A fast communication between both is thus also necessarily.

The generation of the trajectories is based on objective parameters, the local controller is temporary a PID controller of which the actions vary according to the trajectory.

The robot is currently able to move in a quasi-static way, what the goal was of the thesis. The next step in the study is to develop control-strategies enabling the robot to walk in a dynamically stable way while using the adaptable stiffness.

2.3 Le contrôle quasi statique d'un robot marchant propulsé par des actuateurs avec une souplesse adaptative

Cette thèse fait rapport du développement d'un robot bipède, actionné par des muscles artificiels pneumatiques pliés. Les connexions des sept segments (les pieds, les jambes, les jambes supérieures et le corps supérieur), sont assurées par six jointures, ceux-ci sont pour chaque jambe la cheville, le genou et la hanche. Le mouvement du robot est limité à la surface sagittale.

Le contrôle du robot s'applique à deux niveaux. Un contrôle au niveau supérieur - observé par l'ordinateur qui produit les trajectoires des jointures différentes- et un contrôle au niveau local pour chaque jointure -observé par le microcontrôleur. Une communication rapide entre les deux est nécessaire.

La génération des trajectoires est basée sur des paramètres objectifs, le contrôleur local est temporairement un régulateur PID dont les actions varient selon la trajectoire.

Le robot est actuellement capable de se déplacer d'une façon quasi statique, ce que était le but de la thèse. L'étape prochaine d'étude sera de développer des stratégies de contrôle pour le faire marcher d'une façon dynamiquement stable en utilisant la souplesse adaptative.

Inhoudstabel

1. Inleiding	1
1.1 <i>Waarom een studie over stappende robots</i>	2
1.2 <i>Statisch of dynamisch stabiel stappen</i>	2
1.3 <i>Verschil actief-passief stappen</i>	4
1.4 <i>Series elastic actuator</i>	5
1.5 <i>Pneumatische artificiële spieren</i>	6
1.6 <i>Werkwijze</i>	8
1.7 <i>Opbouw van de thesis</i>	8
2. Design	9
2.1 <i>Inleiding</i>	9
2.2 <i>Mechanisch Design</i>	9
2.2.1 <i>Pneumatische artificiële spieren</i>	9
2.2.2 <i>Antagonistische opstelling</i>	11
2.2.3 <i>Stijfheid</i>	12
2.2.4 <i>Algemene beschrijving</i>	13
2.2.5 <i>Keuze spieren en aanhechtingspunten</i>	14
2.2.6 <i>Perslucht</i>	17
2.2.7 <i>Kleppenblok</i>	19
2.2.8 <i>Drukvat</i>	20
2.2.9 <i>Uitlaatdemper</i>	20
2.3 <i>Electronisch design</i>	21
2.3.1 <i>Microcontroller</i>	21
2.3.2 <i>Speed-up circuit</i>	23
2.3.3 <i>Hoekmeting</i>	24

2.3.4 Drukmeting	25
2.3.5 Krachtmeting voet	25
2.3.6 Safety Board	26
3. Controle	28
3.1 <i>Inleiding</i>	28
3.2 <i>Communicatie</i>	28
3.2.1 ISABus-kaart	29
3.2.2 Databus collector	33
3.2.3 Driver	33
3.2.4 Priority-level	34
3.2.5 Fouten in de communicatie	36
3.2.6 USB	38
3.3 <i>Global controle</i>	40
3.3.1 Trajectgeneratie	40
3.3.2 Feedforward	42
3.3.3 Programmatie trajectgeneratie	46
3.3.3.1 Trajectprogrammatie a.d.h.v. discrete posities	46
3.3.3.2 Trajectprogrammatie a.d.h.v. objectieve parameters	48
3.4 <i>Communicatieprogramma</i>	50
3.4.1 Identificeren van de microcontrollers	51
3.4.2 Calibreren van de druksensoren	52
3.4.3 Het instellen van de R,P,I,D-waardes	53
3.4.4 Plaats in het programma	53
3.5 <i>Lokale controle</i>	54
3.5.1 PID regelaar	55
3.5.2 Bang-bang regelaar	57
3.5.3 Implementatie in de microcontrollers	58
3.5.4 Code voor de master	58
3.5.5 Code voor de joints	59

4. Resultaten	60
4.1 Registratie van de meetgegevens	61
4.2 Verhogen van de samplingfrequentie	61
4.3 Afwezigheid grondcontactsensoren	61
4.4 Variëren parameters PID regelaars	62
4.4.1 Invloed I-actie op heup	62
4.4.2 Invloed P-actie op enkel	63
4.5 Communicatiefouten	64
4.6 Drie stappen	67
4.6.1 Traject via objectieve parameters	67
4.6.2 Meetresultaten	69
4.6.3 Bespreking resultaten	70
4.7 Invloed van koppelkarakteristiek	71
4.8 Aanbrengen van extra belasting	73
5. Conclusie en toekomstperspectieven	74
5.1 Conclusie	74
5.2 toekomstperspectieven	76
6. Symbolenlijst	77
7. Literatuurlijst	78
8. Inhoudstabel bijlage	82

Inhoudstabel figuren

1. Inleiding

1-1 Asimo van Honda en SDR-4X van Sony	1
1-2 Statisch gebalanceerd stappen	2
1-3 Dynamisch gebalanceerd stappen	3
1-4 Passieve stapper van TUDelft (Museon en Mike)	4
1-5 Flamingo Spring van MIT	5
1-6 The Shadow Biped van The Shadow Robot Company	7
1-7 Lucy van de Vrije Universiteit Brussel	7

2. Design

2-1 Pneumatische artificiële spier	9
2-2 Roterende antagonistische opstelling van een paar pneumatische artificiële spieren	11
2-3 de gerealiseerde robot Lucy en een designtekening van de mechanische structuur	13
2-4 Definitie aanhechtingspunten hefboommechanisme	14
2-5 Definitie gebruikte hoeken	15
2-6 Rode plaatjes voor instelling aanhechtingspunten met APL	16
2-7 Schema persluchtschakeling	17
2-8 Ventielen persluchtschakeling	17
2-9 Centrale collector met drukmet er om 6 drukvaten te voeden	17
2-10 CAD tekening kleppenblok	19
2-11 Drukvat	20
2-12 Uitlaatdemper	20
2-13 Schema van elektronisch design	21
2-14 Microcontroller moederbord	22
2-15 Speed-up circuit met erachter kleppenblok	23
2-16 Schema oud design en nieuw design speed-up circuit	23
2-17 Hoekencoder zonder beschermkapje	24
2-18 Druksensor met AD-converter	25
2-19 Safety board	27

3. Controle

3-1 ISAbus-kaart	29
3-2 Schema van de controlebits	30
3-3 Schema ISAbus-kaart	32
3-4 Fysieke verbinding microcontrollers met ISAbus-kaart via databus collector	33
3-5 Bovenaanzicht robot met microcontrollers	39
3-6 Staafmodel robot	41
3-7 Screenshot hoofdvenster trajectprogramma a.d.h.v. discrete posities	47
3-8 Screenshot venster verandering robotgegevens	47
3-9 Screenshot hoofdvenster trajectprogramma a.d.h.v. objectieve parameters	48
3-10 Screenshot hoofdvenster communicatieprogramma	50
3-11 Venster voor export naar Excel file	51
3-12 Programma laat toe om drukken in te geven in verschillende eenheden	52
3-13 Ingeven van de waardes voor regelaar	53
3-14 Schema van de opstelling	54
3-15 Actie van de regelaar in functie van het foutsignaal P_f	57

4. Resultaten

4-1 Definitie gebruikte hoeken	61
4-2 Opeenvolgende posities 3 stappen beweging	68
4-3 Verandering koppelkarakteristiek door aanpassen lengte draadstang	71

Inhoudstabel grafieken-tabellen

2. Design

2-1 Dimensieloze krachtsfunctie f	10
2-2 Spierkracht (N) i.f.v. contractie	10
2-3 Opgemeten krachtsgrafiek	14
2-4 Koppelkarakteristiek van de spieren bij een druk van 3 bar (rel)	15
2-5 Weerstand (\bar{W}) daalt van FSR i.f.v. kracht	26
2-1 Gegevens aanhechtingspunten	15

3. Controle

3-1 Testtabel om communicatie te controleren	37
3-2 Deel van de ontvangen waarden	37
3-3 Berekend volume (ml) i.f.v. voor beide kniespieren	44
3-1 Instelling drukken bang-bang regelaar	57

4. Resultaten

4-1 Heuphoek bij stijgende I-actie (0,25-0,5-1-2-5)	62
4-2 Enkelhoek van een op de grond staande voet met grote P-actie (0-20 s) en kleine P-actie (20-40 s)	63
4-3 Enkelhoek van een niet op de grond steunende voet met kleine P-actie en grote P-actie	64
4-4 Communicatiefout leidt tot oscillaties	65
4-5 Verschil tussen 2 opeenvolgende stelwaarden voor een stapbeweging	66
4-6 Meetresultaten voor 3 stappen-beweging	69
4-7 Meetresultaten knie bij correct en foutief ingestelde draadstang	72
4-8 Meetresultaten knie bij toenemende belasting	73
4-1 Opeenvolgende posities 3 stappen-beweging	67

Hoofdstuk 1

Inleiding

Vroeger waren de tweepotige robots enkel terug te vinden in de academische en de militaire wereld. Momenteel is er een evolutie dat ook domotica-toepassingen en ontspanningsrobots worden ontwikkeld. De bekendste tweepotige robots zijn *ASIMO* (Advanced Step in Innovative MObility) van Honda [1] en *SDR-4X* van Sony [2].



Figuur 1-1: Asimo van Honda en SDR-4X van Sony

Dergelijke robots staan aan de frontlinie van de technologie omdat ze een breed spectrum aan geïntegreerde technologieën nodig hebben en er heel wat specifieke problemen dienen opgelost te worden; bijvoorbeeld stabilisatie-problemen op ruw terrein, herkenning van de omgeving, interactie met omgeving en uitvoeren van specifieke taken.

De onderzoeksgroep Multibody Mechanica van de vakgroep Werktuigkunde van de VUB wil zich concentreren op de stap- en loopbeweging, zowel via simulaties als via experimenten. Bij stappen blijft steeds minstens één voet in contact met de grond terwijl bij lopen fasen aanwezig zijn waarbij beide voeten van de grond zijn.

Tweepotige robots kan men globaal opdelen volgens twee criteria:

- Statisch of dynamisch stabiel stappen
- Actief of passief stappen

Hetgeen voor een groot deel de actuator bepaalt.

1.1 Waarom een studie over stappende robots?

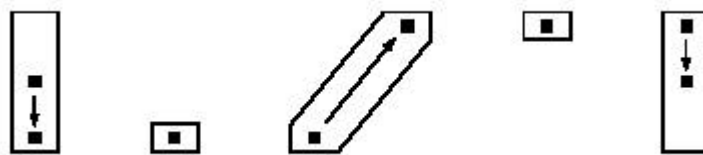
Alhoewel stappende robots de mensen steeds hebben gefascineerd door hun schijnbare complexiteit, zijn ze mobieler dan voertuigen met wielen, vooral op ongestructureerd terrein en in omgevingen gecreëerd door de mens zoals bijvoorbeeld huizen [3].

De eerste reden waarom robots op poten mobieler zijn dan robots op wielen, is omdat ze geïsoleerde contactpunten met de omgeving gebruiken en dus geen ononderbroken pad nodig hebben. De benen fungeren bij stappen als een actief ophangingsysteem voor het bovenlichaam

Het traject van lichaam en voeten kunnen verschillend zijn waardoor een last weinig versnellingen zal ondervinden ondanks het feit dat de stappende robot zich over een ruw terrein verplaatst.

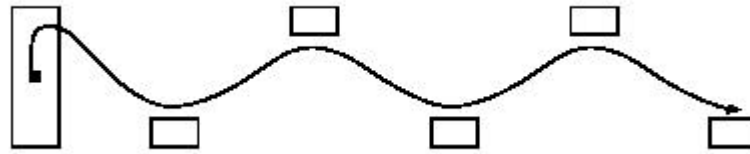
1.2 Statisch of dynamisch stabiel stappen

Bij statisch gebalanceerd stappen ligt de projectie van het zwaartepunt steeds binnen de voet van het ondersteunende been. Dit houdt in dat op elk moment de beweging stilgelegd kan worden zonder dat de robot omvalt. De wijze van voortbewegen van de robot is dan als volgt: het massamiddelpunt van de robot bevindt zich boven één voet, de robot zet de andere voet vooruit en verschuift vervolgens het zwaartepunt naar de tweede voet. Onderstaande figuur geeft een bovenzicht van statisch stappen waarbij het zwaartepunt (voorgesteld door \blacksquare) steeds binnen de polygoon van de ondersteuning valt.



Figuur 1-2: statisch gebalanceerd stappen [3]

Bij dynamisch stappen valt het lichaam naar voren, maar wordt op tijd de voet geplaatst om het vallen te verhinderen, het zwaartepunt is dus niet steeds binnen de voet van het ondersteunende been zoals figuur 1-3 weergeeft. Dit houdt eveneens in dat niet op elk moment de beweging stilgezet kan worden. Dynamisch stappen wordt gekenmerkt door hogere impacteffecten dan bij statisch stappen.



Figuur 1-3: dynamisch gebalanceerd stappen

Alhoewel dynamisch stappen de meest natuurlijke is en de minste energie vergt, bewegen de meeste elektrisch aangedreven robots op een statische manier. Een elektrisch aangedreven robot die wel stapt op een dynamische manier is de robot *Johnnie* [4] van de universiteit van Munchen.

Bij dynamisch stappen is het echter wenselijk om gebruik te maken van een actuator met een natuurlijke soepelheid om de impacten op te vangen. De meeste elektrische motoren worden gekarakteriseerd door een lage koppeldichtheid en kunnen dus enkel een grote vermogendichtheid halen bij hoge draaisnelheden. Dit betekent dat reductiekasten onvermijdelijk worden. Het gebruik van een N:1-overbrenging induceert echter een verhoging van de gereflecteerde traagheid met N^2 dewelke onder andere de tanden van de tandwielen beschadigt bij impacten [4].

Tandwielkasten hebben eveneens last van backlash omdat ze nooit precies in elkaar passen. Dit betekent dat het ene tandwiel een beetje kan draaien zonder dat het andere beweegt. Bij omkeren van de belasting leidt dit tot fouten op de positie.

Het weglaten van de reductiekast is reeds mogelijk bij robotarmen [5], maar de koppeldichtheid van elektrische motoren is nog te klein om dit eveneens toe te passen voor de benen.

Dynamisch gestabiliseerde robots zijn over het algemeen sneller dan statisch gebalanceerde robots omdat men bij statisch stappen de inertie-effecten klein moet houden en dus de snelheid gelimiteerd is.

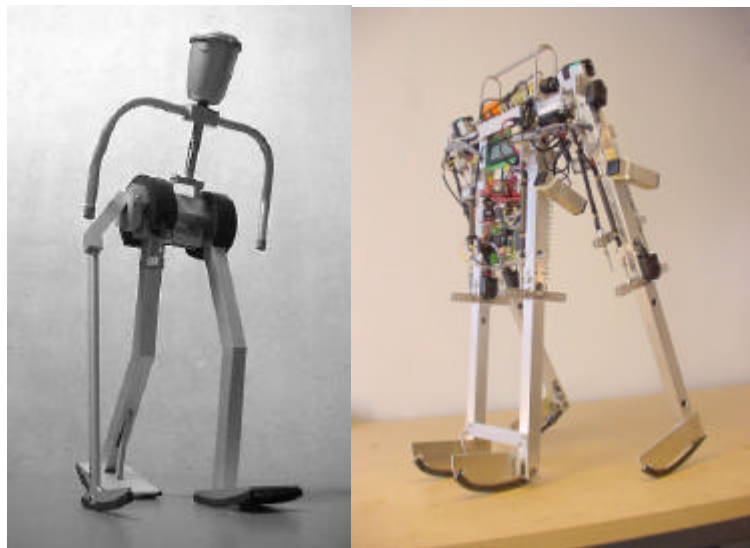
Bij het overgaan van passief naar dynamisch stabiel stappen is het dus wenselijk een actuator te nemen die een natuurlijke soepelheid bezit.

1.3 Verschil actief – passief stappen

Wil men een autonome robot maken dan zal men ook het energieverbruik zo laag mogelijk moeten houden. Vanuit dit oogpunt is de optimale beweging diegene die gebruikt maakt van de eigenfrequentie van het systeem. De *Museon* [6] van de TUDelft gebruikt enkel de gravitatiekracht om te stappen op een dalende helling en bevat geen actieve controle. Dit noemt men passief stappen. Het mechanisme wordt immers zo getuned dat het stappen volledig valt binnen de natuurlijke dynamica van het systeem.

Willen dergelijke robots stappen op een horizontaal vlak dan zal er een minimum aan actuatie nodig zijn om de wrijvingsverliezen te overwinnen. TUDelft heeft eveneens een dergelijke robot gebouwd, namelijk *Mike* [20].

Het nadeel van dergelijke mechanische constructie is dat de trajecten hiermee vastliggen en dus niet meer kunnen veranderd worden tijdens het stappen. De eigenfrequenties zijn namelijk niet meer te veranderen. Dit kan men oplossen door gebruik te maken van actuatoren met passieve elementen die wel een variabele stijfheid hebben en waarbij dus de eigenfrequentie wel adaptief is.



Figuur 1-4: Passieve stapper van TUDelft (Museon en Mike)

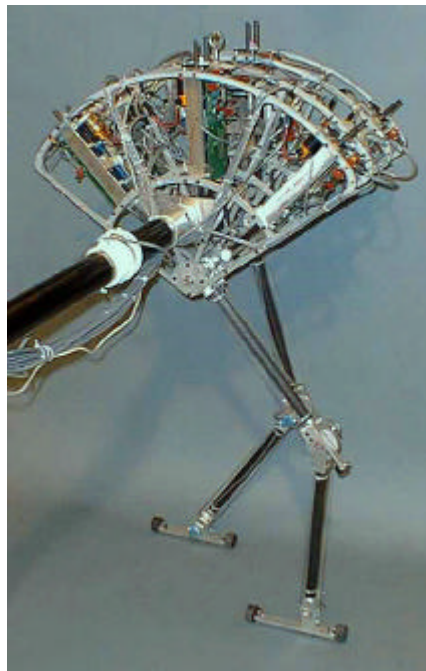
Uit onderzoek blijkt dat ook mensen gebruik maken van passieve dynamica [7]. Electromyografische data opgenomen door Mc Mahon heeft aangetoond dat er weinig elektrische activiteit is gedurende de zwaafase bij normale stapnelheid. Dit suggereert dat het been vrij zwaait gedurende deze periode. In het begin van de zwaafase ontvangt het been slechts een kleine impuls van energie en een tweede om het been te stoppen vlak voor het neerkomen.

1.4 Series elastic actuator

Een belangrijke topic binnen de robotica is dus de controle krijgen over een actuator die een natuurlijke soepelheid bezit.

Prof. Marc Raibert (Leg Lab van MIT) heeft natuurlijke soepelheid in zijn actuators gebracht door de 'series elastic actuator' [8] te ontwikkelen. Deze actuator bestaat uit een elektrische motor in serie geplaatst met een verend element.

De specifieke voordelen van een dergelijke actuator zijn het verhogen van de schoktolerantie, het verlagen van de equivalente traagheid, meer nauwkeurige en stabiele krachtscontrole, energiestockage en minder gevaar voor schade bij ongewild contact. In het Leg Lab van MIT werden voor dit type actuator controlestrategieën ontwikkeld en experimenteel geverifieerd op de *Flamingo Spring* [9]. Een beperking van de series elastic actuator is dat de stijfheid van het elastisch element niet continu aanpasbaar is.



Figuur 1-5: Flamingo Spring van MIT

In deze context ontwikkelde de groep rond Takanishi de *WL-14* waar een complex niet-lineair veermechanisme verandering in stijfheid mogelijk maakt [10]. Een elegantere oplossing is gebruik te maken van pneumatisch artificiële spieren.

1.5 Pneumatisch Artificiële Spieren (PAS)

Pneumatische Artificiële Spieren (PAS) zijn opgebouwd uit een flexibel, vezelverstevigd membraan dat aan beide uiteinden ingeklemd is in een aluminium eindstuk.

Door via een persluchtleiding de spier op te blazen zal door de opbolling de spier samentrekken en een axiale kracht ontwikkelen. De ontwikkelde kracht is evenredig met de interne druk en afhankelijk van de contractie van de spier. Deze kracht gaat van zeer grote waarden bij maximale lengte (0% contractie) tot nul wanneer de spier helemaal samengetrokken is (maximale contractie).

De PAS is gekenmerkt door zijn zeer hoge kracht tot eigengewicht verhouding en de mogelijkheid om rechtstreeks, d.w.z. zonder snelheidsreductie, aan de robotstructuur gekoppeld te worden. Dankzij deze eigenschappen kan het gewicht van de structuur sterk gereduceerd worden, de nuttige last verhoogd en de bouw vereenvoudigd worden, in vergelijking met elektrische aandrijvingen [11].

De spieren ontwikkelen enkel een trekkracht en geen drukkracht. Er zijn dan ook twee antagonistisch opgestelde spieren nodig om een bidirectioneel roterend scharnier te vormen.

Het verschil van de drukken in beide spieren bepaalt het moment en daaruit volgend de hoek, de som van beide drukken bepaalt de stijfheid. Stijfheid en positie kunnen bijgevolg onafhankelijk van elkaar ingesteld worden.

De meest gekende artificële spieren zijn de Mc Kibben muscles. Het bestaat uit een rubberen buis dewelke uitzet indien het opgeblazen wordt, het omhullende netje draagt de spanning over. Door de wrijving tussen het buisje en het netje is er hysteresis die de controle van dergelijke spieren moeilijker maakt.

Ook hebben ze een bepaalde drempeldruk nodig voordat de spieren reageren.

Een robot uitgerust met dergelijke spieren is *The Shadow Biped* van The Shadow Robot Company [12]. Deze robot heeft echter nooit gestapt.



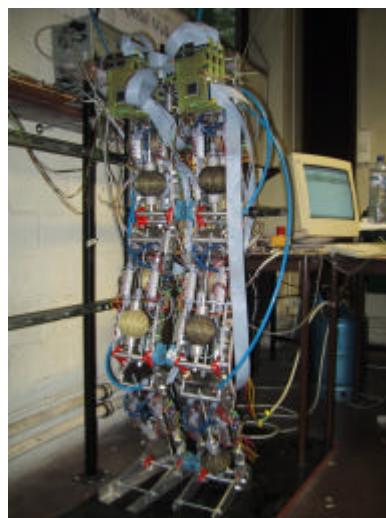
Figuur 1-6: The Shadow Biped van The Shadow Robot Company

Andere tweepotige robots uitgerust met McKibben spieren zijn de *Salford Lady* en de *Gorilla* [13] van de universiteit van Salford.

De Geplooide Pneumatische Artificiële Spier (GPAS) bezit deze negatieve eigenschappen van stickslip en hysteresis niet vermits het materiaal niet uitrekt maar door het ontvouwen van de plooiën een volumeverandering ondergaat.

Om al deze ideeën te kunnen testen, niet alleen via computersimulaties, werd de robot *Lucy* op de vakgroep Werktuigkunde van de Vrije Universiteit Brussel ontworpen.

Het doel is om een lichtgewicht tweepotige robot te maken die in staat is om te stappen op een dynamische manier, waarbij het passief karakter van de pneumatisch artificieële spieren ten volle wordt benut [14], [15], [21], [22].



Figuur 1-7: Lucy van de Vrije Universiteit Brussel

1.6 Werkwijze

Wegens de complexiteit van de robot was het noodzakelijk de robot stap voor stap op te bouwen.

De werkwijze bestond uit volgende fasen:

- Bouw, testen en debuggen van de verschillende elektronische componenten en aanmaken van een bibliotheek van programma's om de elektronica uit te lezen en te sturen via de microcontrollers.
- Ontwikkeling driver voor de ISABus-kaart en testen van de communicatie tussen de PC en microcontrollers.
- Programmatie PID regelaar voor trajectsturing en schrijven van communicatieprogramma. Dit programma stuurt het traject naar de robot en registreert de meetgegevens.
- Opbouw, gewricht per gewricht, van de robot
- Schrijven programma voor trajectgeneratie via objectieve parameters
- Uitvoeren van stapbewegingen

1.7 Opbouw van de thesis

Een beschrijving van de hardware componenten vindt men terug in *hoofdstuk 2 Design*. Dit bevat enerzijds het mechanisch gedeelte waarin de bespreking van de spieren het belangrijkste is en anderzijds het elektronisch gedeelte nodig om de drukken in de spieren te regelen. De drukken bepalen de posities en stijfheden van de gewrichten.

In *hoofdstuk 3 Controle* wordt de controlearchitectuur van de robot besproken die op twee niveaus gebeurt. Een controle op globaal niveau door de PC en een controle op lokaal niveau uitgevoerd door een microcontroller in elk gewricht. De PC produceert de trajecten aan de hand van objectieve parameters en de lokale PID regelaar zal dit traject trachten te volgen door regeling van de druk in de spieren. Daar een snelle communicatie tussen globale en lokale controller onontbeerlijk is start het hoofdstuk controle met dit onderwerp, vervolgens wordt de globale en lokale controle besproken.

De belangrijkste bevindingen aangaande de quasi-statische controle staan in *hoofdstuk 4 Resultaten*. De robot is in staat om op een trage manier bewegingen uit te voeren waaronder stappen te zetten.

Hoofdstuk 2

Design

2.1 Inleiding

Het doel van de robot Lucy is om de principes van de GPASsen te onderzoeken in een experimentele opstelling. Gans de opstelling is dan ook hiernaar gebouwd.

Dit hoofdstuk start met het mechanisch design, dit omvat een bespreking van de spieren en hun inbouw in de robot.

De elektronische componenten nodig om de druk in spieren te regelen staan in het deel elektronisch design.

2.2 Mechanisch Design

2.2.1 Pneumatische artificiële spieren.

Pneumatische artificiële spieren zijn opgebouwd uit een flexibel, vezelverstevigd membraan dat aan beide uiteinden ingelijmd is met epoxy in een aluminium eindstuk.

Het membraan van de spieren, gemaakt van polyester en luchtdicht gemaakt met polypropyleen, is plooigewijs opgebouwd zodat opbolling van het geheel kan gebeuren zonder dat het materiaal daarvoor hoeft uit te rekken. Dit zou immers gepaard gaan met een energieverlies en hysteresis. In de plooien bevinden zich Kevlar-vezels die de krachten overdragen op de aluminium eindstukken.

De aluminium eindstukken bevatten eveneens de aansluitingen van de persluchtleiding.



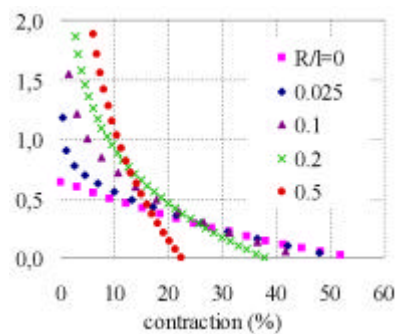
Figuur 2-1: pneumatische artificiële spier

Een axiale kracht wordt ontwikkeld bij het op druk brengen van de spier. Indien de vezels niet elastisch worden ondersteld dan is de ontwikkelde kracht evenredig met de interne druk p en afhankelijk van de contractie van de spier α , de initiële lengte L en de slankheid R/L [11].

$$F = p \cdot L^2 \cdot f\left(\alpha, \frac{R}{L}\right)$$

Deze kracht gaat van zeer grote waarden bij maximale lengte (0% contractie) tot nul wanneer de spier helemaal samengetrokken is (maximale contractie).

Onderstaande grafiek (2-1) toont de krachtsfunctie f voor verschillende slankheden in functie van de contractie.

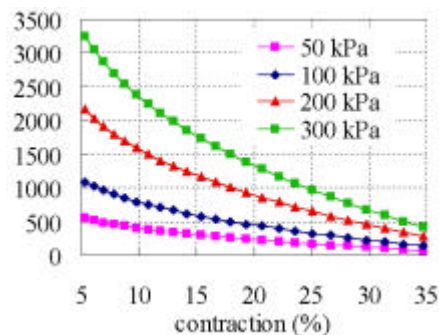


Grafiek 2-1: dimensieloze krachtsfunctie f [11]

In praktijk worden de limieten van de contracties beperkt van 5% tot 35%. Bij contracties lager dan 5% zijn immers de spanningsconcentraties te groot, bij contracties hoger dan 35% zijn de krachten onvoldoende.

Het is in deze range dat het bovenstaande mathematisch model de experimentele resultaten benadert met een fout kleiner dan enkele percenten.

Op onderstaande grafiek (2-2) zien we de ontwikkelde kracht van een spier met een initiële lengte van 10 cm en een straal van 1,25 cm, in functie van de contractie, voor verschillende waarden van de toegepaste druk. De karakteristiek is gelijkaardig met die van een skeletspier.

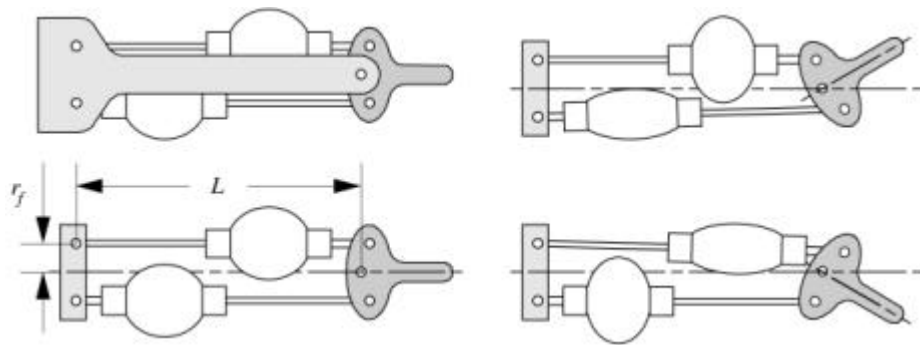


Grafiek 2-2: spierkracht (N) i.f.v. contractie [11]

2.2.2 Antagonistische opstelling

De spieren ontwikkelen enkel een trekkracht en geen drukkracht. Er zijn dan ook twee antagonistische opgestelde spieren nodig om een bidirectioneel roterende scharnier te vormen. In elk gewricht zijn beide spieren verbonden aan een hefboommechanisme. De punten van aanhechting zijn samen met de spierkarakteristiek bepalend voor de koppelkarakteristieken in functie van de hoek.

Het hefboommechanisme heeft een asymmetrische werking rond zijn centrale positie dewelke de grote niet-lineariteit van de spieren kan compenseren. Zoals immers op onderstaande figuur (2-2) te zien valt wordt de hefboomsarm kleiner wanneer de spier weinig gecontracteerd is en dus een grote kracht bezit, terwijl bij grote contracties en dus een kleine kracht, de hefboomsarm vergroot wordt.



Figuur 2-2: roterende antagonistische opstelling van een paar pneumatische artificiële spieren [11]

Uitgaande van $F = p \cdot L^2 \cdot f\left(\mathbf{e}, \frac{R}{L}\right)$ kan men het krachtmoment berekenen [11].

$$M_1 = p_1 \cdot L_1^2 \cdot r_1 \cdot f_1 = p_1 \cdot m_1(\mathbf{a}) \quad \text{voor strekspier}$$

$$M_2 = p_2 \cdot L_2^2 \cdot r_2 \cdot f_2 = p_2 \cdot m_2(\mathbf{a}) \quad \text{voor buigspier}$$

$$M = M_2 - M_1 = p_2 \cdot m_2(\mathbf{a}) - p_1 \cdot m_1(\mathbf{a}) \quad \text{totaal koppel van het gewricht}$$

Waarin r_1 en r_2 de hefboomsarmen zijn, dewelke afhankelijk zijn van de hoek \mathbf{a} . m_1 en m_2 zijn de koppelkarakteristieken.

De keuze van de aanhechtingpunten moet dus nauwgezet gebeuren omdat het samen met de spierkarakteristiek de functies m_1 en m_2 vastlegt. Men moet uitkijken naar volgende criteria:

- is er in gans het werkingsdomein voldoende koppel aanwezig om de last te dragen
- valt de contractie binnen de limieten (5%-35%)
- voldoende afstand tussen de spieren bij de opbolling, zodat ze elkaar niet hinderen

Om reeds een deel van het probleem van aanraking van de spieren op te lossen werden de spieren geschrinkt geplaatst.

2.2.3 Stijfheid

De stijfheid (=1/soepelheid) van een gewricht wordt bepaald door de koppelkarakteristiek $M = M_2 - M_1 = p_2 \cdot m_2(\mathbf{a}) - p_1 \cdot m_1(\mathbf{a})$ af te leiden naar de hoek [11]:

$$K(=C^{-1}) = \frac{dM}{d\mathbf{a}} = \frac{d(M_2 - M_1)}{d\mathbf{a}} = \frac{dp_2}{d\mathbf{a}} m_2 + p_2 \frac{dm_2}{d\mathbf{a}} - \frac{dp_1}{d\mathbf{a}} m_1 - p_1 \frac{dm_1}{d\mathbf{a}}$$

De soepelheid van de GPAS vindt dus zijn oorsprong in enerzijds de samendrukbaarheid van lucht ($\frac{dp_i}{d\mathbf{a}}$) en anderzijds de dalende kracht-contractie curve ($\frac{dm_i}{d\mathbf{a}}$).

Dit maakt ook energierecuperatie mogelijk waarvoor er wordt verwezen naar [18].

Uit [11] haalt men dat het verschil van de drukken in beide spieren het moment bepaalt en daaruit volgend de hoek, de som van beide drukken bepaalt de stijfheid.

Hoe groter deze som, hoe stijver de structuur. Stijfheid en positie kunnen bijgevolg onafhankelijk van elkaar ingesteld worden.

Hier komt het grote verschil met elektrische aandrijvingen naar boven. Deze kunnen enkel de positie instellen maar bezitten geen soepelheid indien een tandwielkast aanwezig is. De tandwielkast is nodig omdat de vermogensdichtheid te klein is bij lage toerentallen.

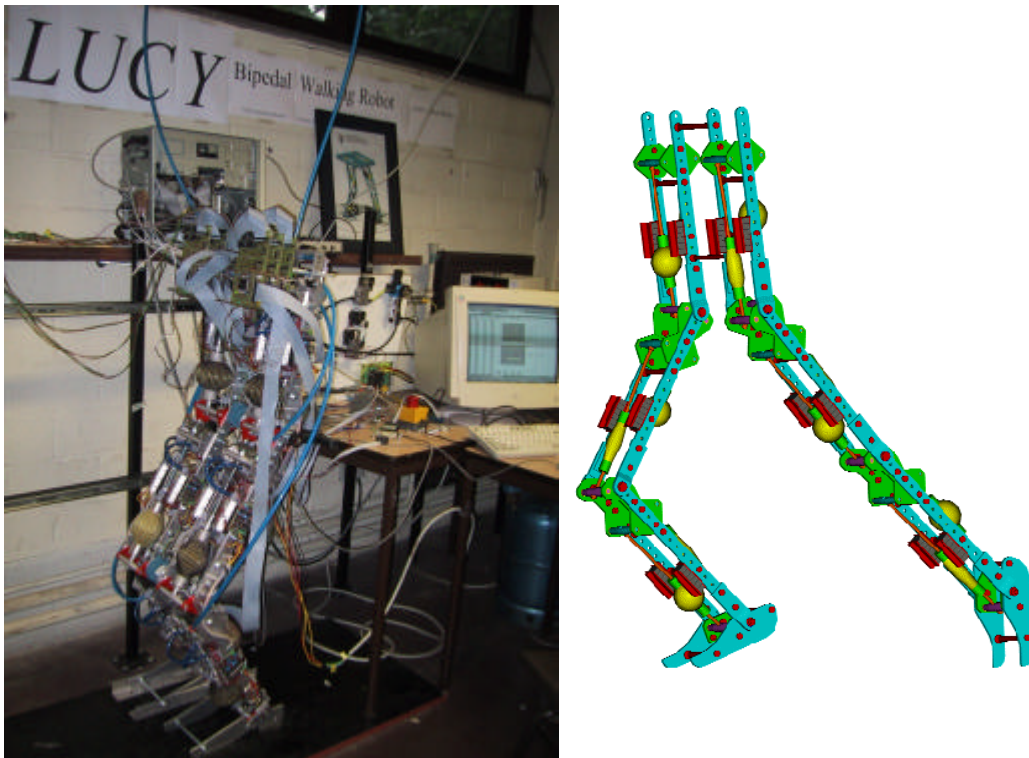
Pneumatische spieren kunnen echter zonder overbrengingsmechanisme aan de structuur worden gekoppeld. Dit vereenvoudigt niet alleen het design, maar bespaart ook op het gewicht.

2.2.4 Algemene beschrijving

Eén van de basisgedachten in het design van de robot is de modulariteit dewelke zich vertaalt in dezelfde configuratie van zowel onderbeen, bovenbeen als torso. Enkel de voet is anders. Dit maakt het mogelijk om de robotconfiguratie eenvoudig te veranderen voor verschillende experimenten en vereenvoudigt de design en de productie.

De structuur is gemaakt uit hoogwaardig aluminiumlegering, AlSiMg1 en bestaat uit twee benen en een bovenlichaam. Beide benen zijn identiek en hebben elk een bovenbeen, onderbeen en een voet. Ze zijn met elkaar verbonden door scharniergerichten en creëren zo de heup, de knie en de enkel. De heup is verbonden met een glijdingsmechanisme zodanig dat het zijdelings kantelen en roteren volgens de verticale as van de robot vermeden wordt en zijn bewegingsruimte wordt beperkt tot het sagitaal vlak. De robot dient immers om de principes van GPASsen te testen. Het doel is niet om een autonome robot te bouwen.

Het geheel weegt 28,2 kg en is 150 cm groot.



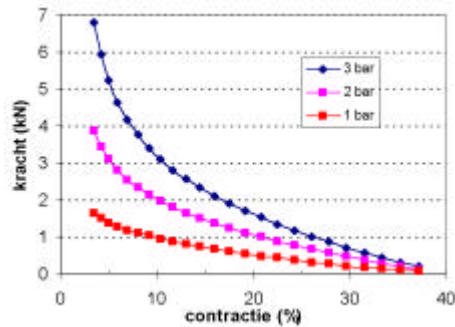
Figuur 2-3: de gerealiseerde robot Lucy en een designtekening van de mechanische structuur

2.2.5 Keuze spieren en aanhechtingspunten

De parameters van de spieren die de krachtskarakteristiek bepalen zijn initiële lengte L en slankheid R/L .

Voor een optimalisatie zouden deze parameters voor elke gewricht moeten variëren. Voor de eenvoud en modulariteit werd dit niet gedaan en zijn alle spieren dezelfde. De waarden zijn vastgelegd op $L=110$ mm en $R/L= 0,145$.

Deze spier heeft de volgende experimentele krachtsgrafiek.

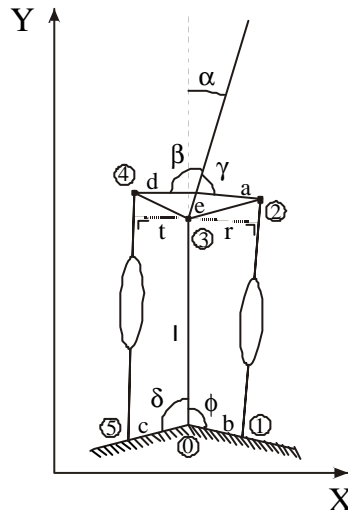


Grafiek 2-3: opgemeten krachtsgrafiek

Ook de aanhechtingspunten van de spieren zijn van groot belang. De punten worden gedefinieerd door de afstanden a , b , c en d en de hoeken β , γ , δ en ϕ op figuur 2-4.

De mechanische uitvoering van de aanhechtingspunten van de spieren die de afstanden d en a en de hoeken β en γ vastleggen, is uitgevoerd in staal en is gemakkelijk vervangbaar om in een verder stadium de invloed van de aanhechtingspunten te onderzoeken. Op de foto's zijn het de rode plaatjes aan de gewrichten zoals op figuur 2-6.

De plaatjes die de afstanden b en c en de hoeken δ en ϕ vastleggen, zijn voor de 3 gewrichten dezelfde.



Figuur 2-4: definitie aanhechtingspunten hefboommechanisme

	HEUP	KNIE	ENKEL
Lengte a (mm)	35	37	35
Lengte b (mm)	40	40	40
Lengte c (mm)	40	40	40
Lengte d (mm)	35	32	35
Lengte e (mm)	0	0	0
Lengte l (mm)	350	350	350
Hoek b ($^{\circ}$)	90	90	90
Hoek g ($^{\circ}$)	45	17	55
Hoek d ($^{\circ}$)	65	105	55
Hoek F ($^{\circ}$)	65	90	90
Lengte draadstang vooraan (mm)	188	181	187
Lengte draadstang achteraan (mm)	188	188	187

Tabel 2-1: gegevens aanhechtingspunten

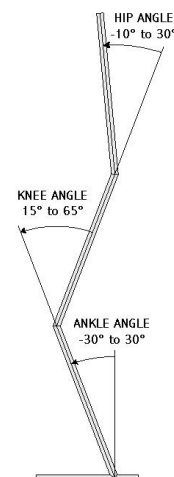
De afstand tussen cd en ab wordt verwezenlijkt door de spier en de overblijvende lengte door een draadstang. De lengte van de draadstang is belangrijk bij het monteren van de spieren, vermits het bepaalt hoeveel de contractie van de spier bedraagt bij een zekere hoek α . Neemt men een andere lengte dan zal ook de koppelkarakteristiek veranderen.

Om te verhinderen dat er singuliere punten optreden wordt de range van de hoeken beperkt. In dergelijke standen kunnen de assen immers verwrongen worden.

Deze zijn voor de

Heup: $-10^{\circ} \rightarrow 30^{\circ}$ ($-18^{\circ} \rightarrow 30^{\circ}$)
 Knie: $15^{\circ} \rightarrow 65^{\circ}$ ($8^{\circ} \rightarrow 68^{\circ}$)
 Enkel: $-30^{\circ} \rightarrow 30^{\circ}$ ($-15^{\circ} \rightarrow 10^{\circ}$)

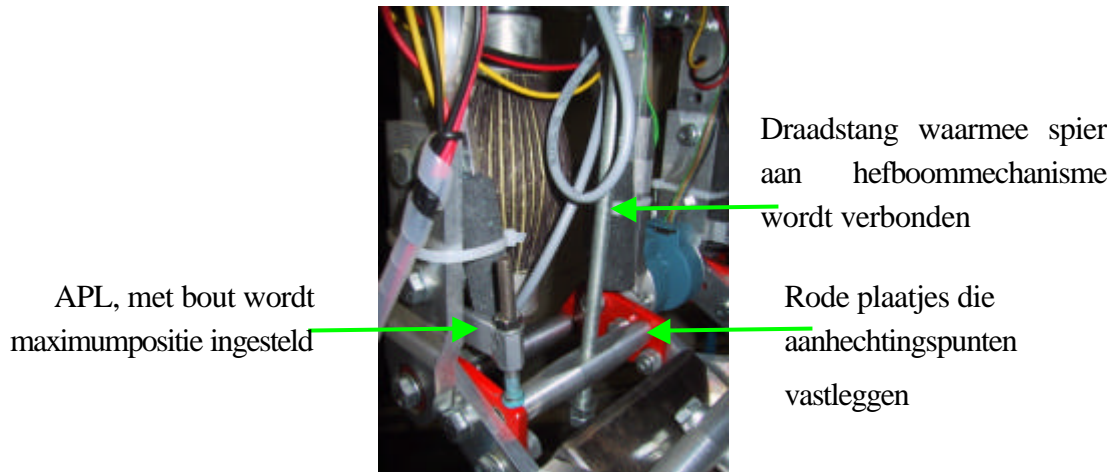
De waarde tussen haakjes is de gebruikte range bij een stappende mens [23]. Merk op dat de range van de enkelhoek beduidend beperkter is bij de mens. De voet van de mens rolt



Figuur 2-5: definitie gebruikte hoeken

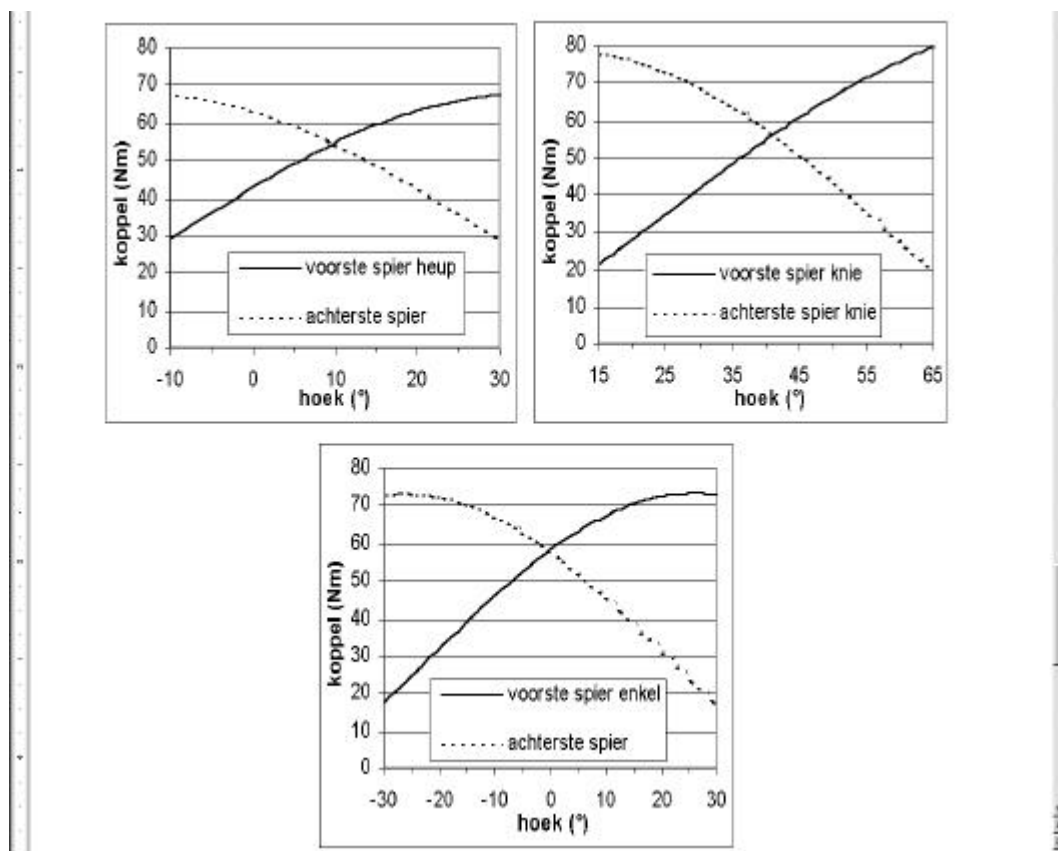
immers mee af bij het stappen, wat de uitvoering van de voetstructuur is star.

De beperking van de rotatie van een gelid wordt bekomen door instelling van de bouten van de Angular Position Limiter (APL). Op onderstaande foto (2-6) ziet men de APL alsook de rode plaatjes waarmee de instelling van de aanhechtingspunten vastliggen.



Figuur 2-6: rode plaatjes voor instelling aanhechtingspunten met APL

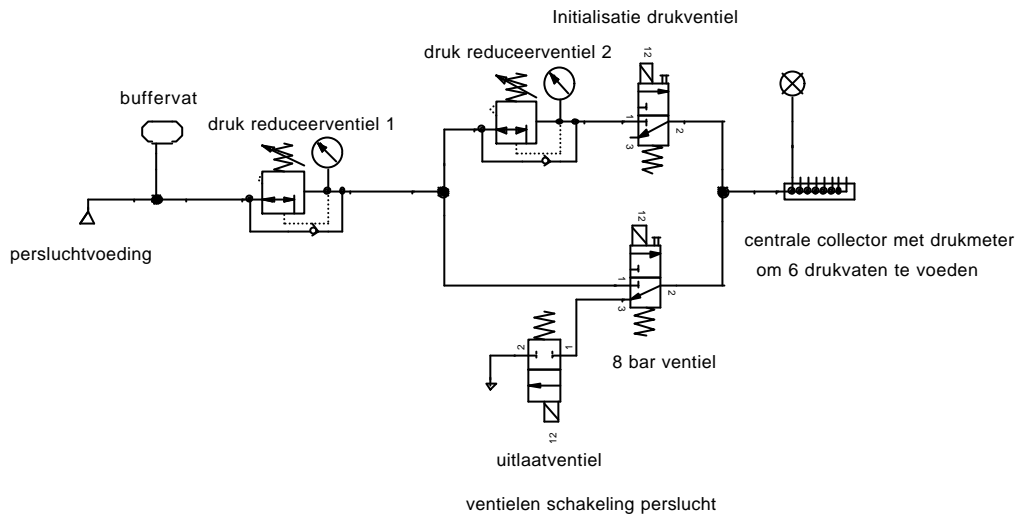
Men krijgt volgende koppelkarakteristieken voor een druk van 3 bar (rel).



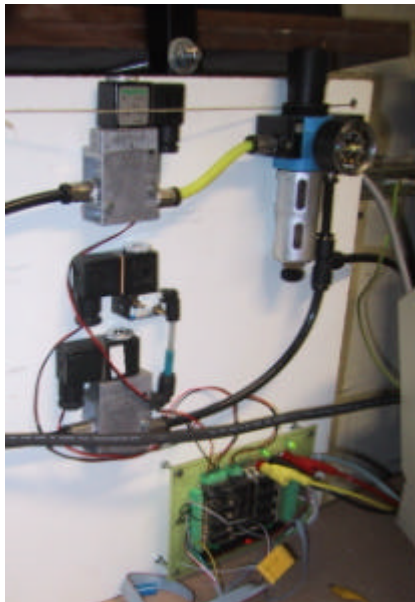
Grafiek 2-4: koppelkarakteristiek van de spieren bij een druk van 3 bar (rel)

2.2.6 Perslucht

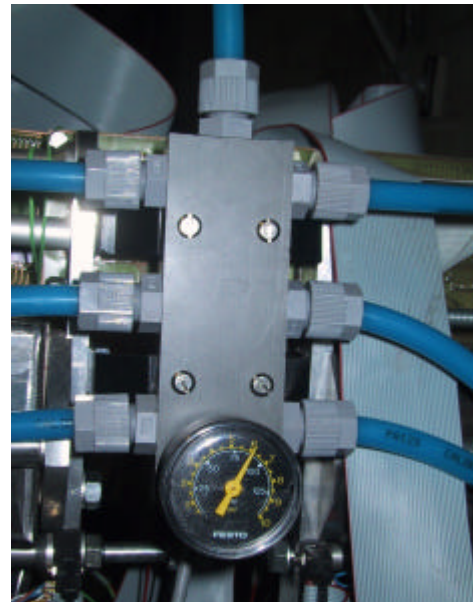
Schema 2-7 is de persluftschemakeling tot aan de centrale collector. Vanaf deze centrale collector worden 6 buffervaten gevoed, één voor elk gewricht. Elk buffervat voedt twee kleppenblokken: één voor de buigspier en één voor de trekspier. Door het schakelen van de kleppen van de kleppenblok wordt de druk in de spieren geregeld. Aan uitlaat van de kleppenblokken staan de geluidsdempers.



Figuur 2-7: schema persluftschemakeling



Figuur 2-8: ventielen persluftschemakeling



Figuur 2-9: centrale collector met drukmeter om 6 drukvaten te voeden

Deze schakeling kan zich in 3 modes bevinden: voeding afgesloten en ontluchte leidingen, leidingen op calibratiedruk en leidingen op voedingsdruk.

Mode 1) Voeding afgesloten/ontluchten leiding:

Dit is de veilige toestand van het systeem omdat de robot geen krachten kan uitoefenen zonder luchtdruk, bij foutsituaties wordt automatisch naar deze toestand gegaan.

- Initialisatiedrukventiel: onbekrachtigd
- 8 bar drukventiel: onbekrachtigd
- Uitlaatventiel: bekrachtigd

Mode 2) Calibratievoeding:

Bij de initialisatie van de robot moeten de druksensoren, die de druk in de spieren opmeten, worden gecalibreerd. Dit wordt gedaan aan de hand van twee drukken. Enerzijds de atmosferische druk door de uitlaatkleppen van de kleppenblok open te zetten, anderzijds door een gekende druk in de spieren te brengen. Deze druk wordt ingesteld met drukreducerventiel 2.

- Initialisatiedrukventiel: bekrachtigd
- 8 bar drukventiel: onbekrachtigd
- Uitlaatventiel: onbekrachtigd

Mode 3) Voeding:

Dit is de mode waarin de robot zich bevindt bij normale werking. De voedingsdruk wordt ingesteld met drukreducerventiel 1. Het “8 bar ventiel” moet een ventiel zijn met groot debiet omdat gans de voeding door deze klep stroomt.

- Initialisatiedrukventiel: onbekrachtigd
- 8 bar drukventiel: bekrachtigd
- Uitlaatventiel: bekrachtigd

Het elektrisch circuit voor de bediening van deze kleppen staat op een safety board, beschreven op het einde van dit hoofdstuk.

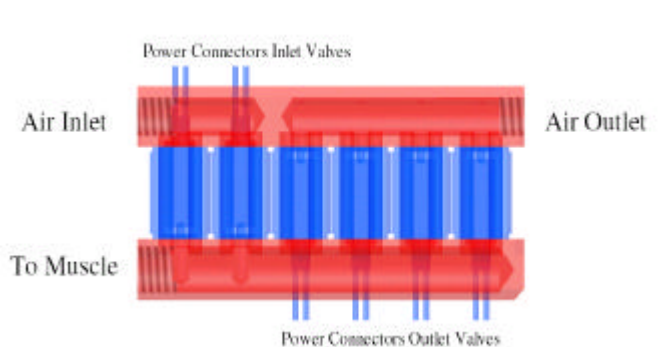
2.2.7 Kleppenblok

De kracht die de spieren uitoefenen is evenredig met de druk in de spieren. Een fijne drukregeling is dan ook noodzakelijk om tot een juiste hoek en stijfheid te komen. Uit [16] blijkt dat er best 6 aan/uit-ventielen gebruikt worden (Matrix 821 2/2). Waarvan er 2 dienen als inlaatklep en 4 als uitlaatklep.

De voedingsdruk is 8 bar, de gemiddelde druk in de spieren is 2,5 bar en de omgevingsdruk is 1 bar. Door het grotere drukverschil tussen voedingsdruk-spiersdruk zijn er minder ingangskleppen nodig dan uitlaatkleppen.

Om ingewikkelde aaneenschakelingen van persluchtleidingen te vermijden werden de twee inlaatkleppen en de vier uitlaatkleppen allen samen op één collector geplaatst. Om het gewicht van het geheel te beperken werden de aluminium eindplaatjes van de zes kleppen verwijderd en vervangen door een nieuw aluminium eindstuk. Eén eindstuk bevat de aansluiting van de persluchtleiding en de uitlaatdemper, het andere eindstuk is verbonden met de spier.

Het geheel van de twee eindstukken en de 6 kleppen vormt de kleppenblok, waarvan een CAD tekening op onderstaande figuur te zien is.



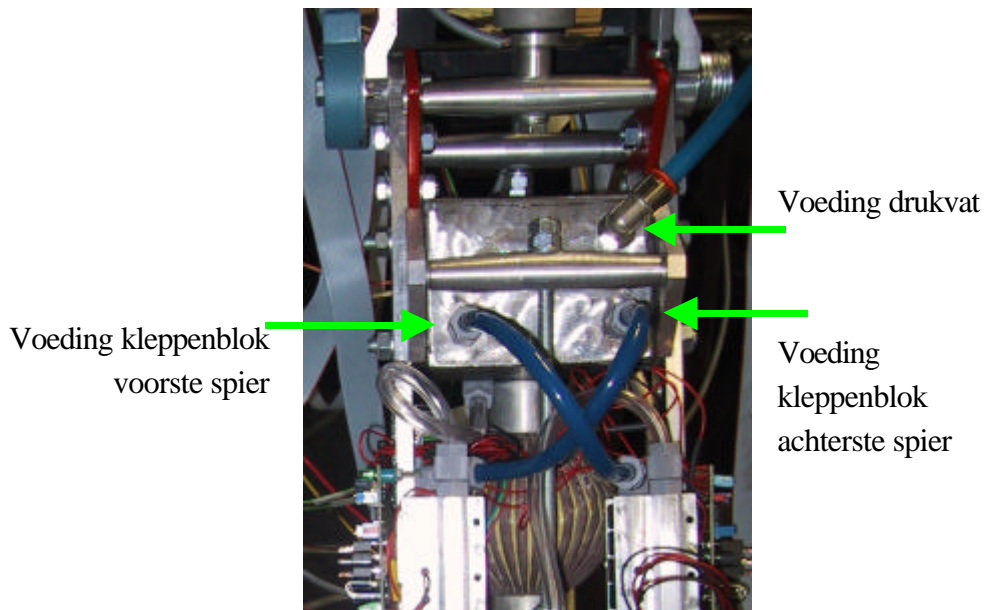
Figuur 2-10: CAD tekening kleppenblok

De goede werking van de kleppen werd belemmerd door vervuiling afkomstig van zowel de spier, zoals stukjes epoxy, als uit de toevoerleiding. De grootste problemen werden veroorzaakt door de magnetiserende deeltjes omdat ze worden gemagnetiseerd door de aanwezige spoel en dus aangetrokken worden. Om de kleppen hiertegen te beschermen werd aan voedingsinlaat en aan de leiding naar de klep een filtergaas geplaatst om het vuil tegen te houden.

2.2.8 Drukvat

Elk gewricht is uitgerust met een buffervat waarvan het volume ongeveer hetzelfde is als een spier. Dit is nodig omdat anders bij het openen van de klep de druk in de leiding te snel wegvalt. Dit resulteert in een slechte drukregeling.

De verschillende drukvaten worden gevoed door een centrale collector.

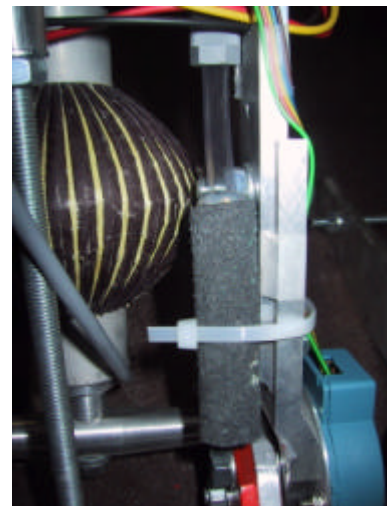


Figuur 2-11: drukvat

De luchtdrukleidingen onder voedingsdruk zijn blauw en staan onder voedingsdruk, de leidingen van de spier in soepele doorzichtige darm en staan op dezelfde druk als de spier.

2.2.9 Uitlaatdemper

De uitgang van elke kleppenblok bevat een geluidsdemper om het geluidsniveau te beperken. Deze demper zorgt wel voor extra stromingsweerstand van de lucht. Het dempend materiaal is een geperforeerde darm uit gerecycleerde autobanden, door de grote oppervlakte kan men het nodige debiet halen.



Figuur 2-12: uitlaatdemper

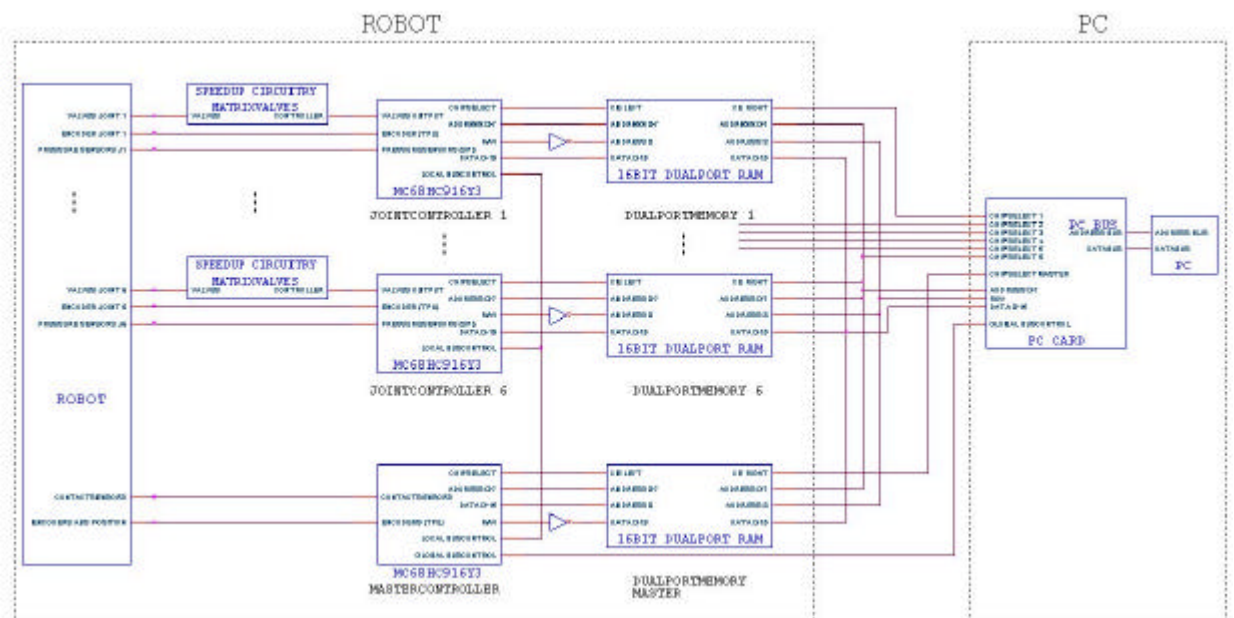
2.3 Elektronisch Design

Elk gewricht heeft zijn eigen microcontroller, er zijn dus zes microcontrollers nodig plus een zevende microcontroller die de master is van de communicatie.

De parameters waarmee de positie en stijfheid van een gewricht worden bepaald zijn de drukken in de strekspier en buigspier en de hoek van het gewricht.

Deze worden als inputgrootheden aangeboden aan de microcontroller. De microcontroller zal vervolgens de kleppen sturen van de verschillende spieren via de speed-up circuits.

De gewenste positie ontvangt de microcontroller van de PC via de PC-kaart en de dual ported RAM. Een gedetailleerde beschrijving vindt men in hoofdstuk 3.



Figuur 2-13: schema van elektronisch design

2.3.1 Microcontroller

Als microcontroller werd de MC68HC916Y3 van het merk Motorola gekozen.

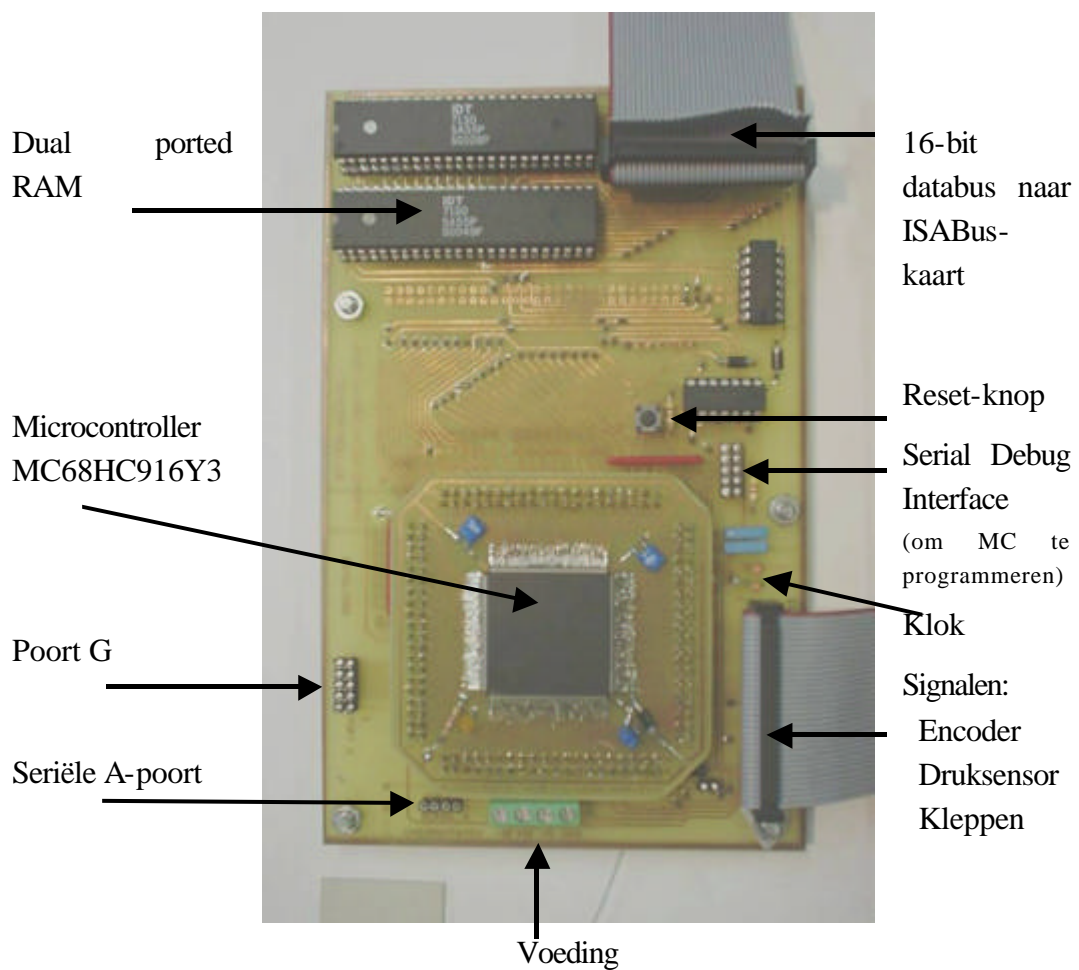
De belangrijkste redenen zijn:

- Er was reeds ervaring met Motorola microcontrollers binnen de vakgroep Werktuigkunde.
- 16-bit: 8-bit is te weinig, 32-bit is beste maar was nieuw en nog te duur
- 16 MHz kloksnelheid.

- 100 Kb flash EEPROM.
- De *Time Processor Unit (TPU)* is een intelligente semi-autonome microcontroller voor input/output opdrachten. De TPU functies vervangen software code van de *Central Processor Unit (CPU)* en belasten daardoor de CPU niet.

Programmatie van de controller dient te gebeuren in assemblercode omdat het uitvoeren van de code veel sneller is dan wanneer het programma wordt geschreven in een hogere taal als C. Deze code wordt door een computerprogramma geassembleerd en omgezet in machinecode, die op zijn beurt door een ander computerprogramma in de controller geladen wordt.

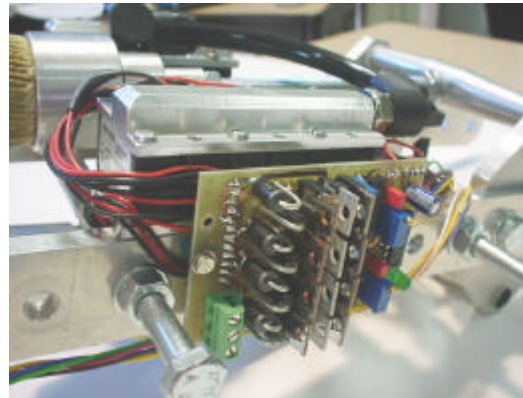
Het microcontroller moederbord werd ontwikkeld binnen de vakgroep Werktuigkunde om volledig de behoeften van de robot te dekken.



Figuur 2-14: microcontroller moederbord

2.3.2 Speed-up circuit

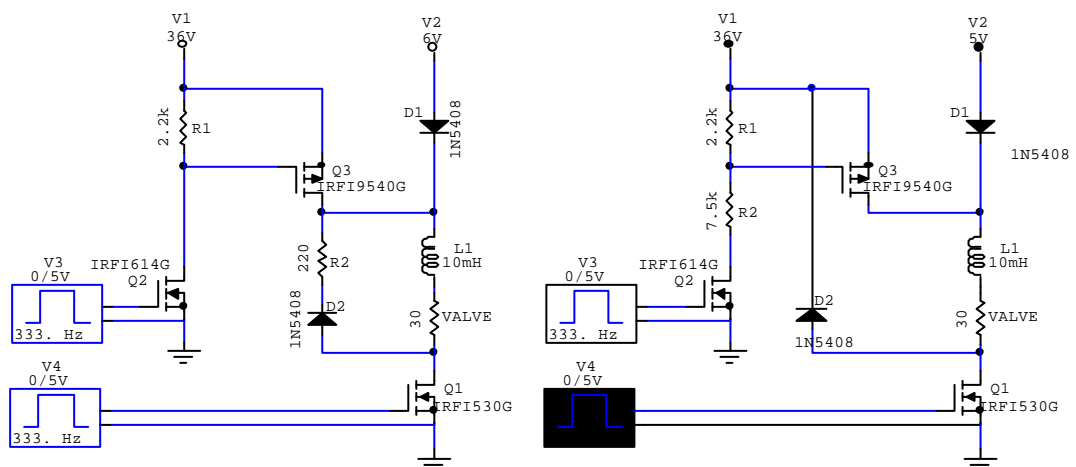
Daar het snel openen en sluiten van de kleppen noodzakelijk is om een goede drukregeling te bekomen is er een speed-up circuit tussen de microcontroller en de kleppen gemonteerd. Gedurende 1 ms wordt er een spanning van 36V over de kleppen gezet, waarna de spanning terugvalt tot 5V. Indien immers de 36V over de klep wordt aangehouden dan brandt de spoel door.



Figuur 2-15: speed-up circuit met erachter kleppenblok

Het elektronisch schema van figuur 2-16 werd ontwikkeld in [16], volgende verbeteringen werden aangebracht.

- Om de microcontroller beter te beschermen werd er aan het ingangssignaal een buffer voorzien en pull-up weerstanden om ongedefinieerde ingangen te vermijden.
- De stuurspanning voor mosfet IRF610 werd via een spanningsdelers beperkt daar een $V_{gs} > 20V$ de mosfet kan beschadigen.
- Om een snellere sluiting te bekomen werd de negatieve klem van de klep via diode 5408 verbonden met de 36V. Hierdoor wordt de spoel sneller ontladen en sluit de klep ook sneller. Uit simulaties in Circuitmaker bleek dat de sluiting hierdoor 58% sneller was.
- Oppervlakte printplaat werd gereduceerd van 136 cm² naar 37 cm².



Figuur 2-16: schema oud design en nieuw design speed-up circuit

2.3.3 Hoekmeting

De hoek wordt gemeten door een incrementele encoder (HEDM6540 van Agilent), welke een resolutie geeft van 8000 eenheden per omwenteling.

Dergelijke opnemer bestaat uit een optische schijf met lichte en donkere velden, die langs een opnemer beweegt. De opnemer bestaat uit 3 sensoren. De 2 eerste sensoren tellen het aantal licht-donker velden. Naargelang de ene blokgolf voor- of najlt ten opzichte van de andere draait men wijzersin of tegenwijzersin. De derde sensor dient om de teller van de andere pulsen te initialiseren. Dus het gewricht moet bij de initialisatie over gans het gebied bewegen om deze nuldoorgang te detecteren. Dit wordt gedaan door in het begin gans het been te strekken.

Vermits de robot 9 vrijheidsgraden heeft indien deze in de lucht hangt:

$$\begin{array}{r} \# \text{ geleidingen:} \quad 7 * 3 = \quad 21 \\ \# \text{ aantal pingewrichten:} \quad 6 * 2 = \quad -12 \\ + \text{-----} \\ \quad \quad \quad \quad \quad \quad \quad 9 \end{array}$$

Daarvan worden er reeds 6 gemeten door de hoekencoders in de gewrichten. Wil men de volledige positie kennen van de robot in de ruimte zal men nog 3 extra metingen moeten verrichten.

Dit zal in de toekomst gedaan worden door de master-microcontroller. Deze zal de absolute hoek, de horizontale en verticale positie van het bovenlichaam opmeten.

Voor de controle van het stappen zijn die metingen niet noodzakelijk omdat men ervan uitgaat dat bij stappen steeds één voet op de grond staat.

De encoders zijn verbonden aan de TPU van de microcontroller. Door dan de *Quadrature Decode TPU function (QDEC)* op te roepen wordt de verwerking van de signalen van de encoder verwerkt door de TPU en is de hoekpositie steeds beschikbaar in een register.



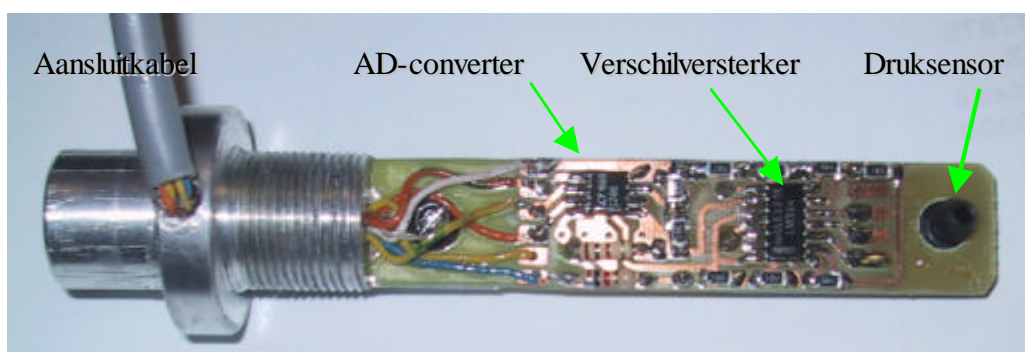
Figuur 2-17: hoekencoder zonder beschermkapje

2.3.4 Drukmeting

De druk wordt in de spieren gemeten. Een te lang verbindingsstuk tussen sensor en drukvat introduceert immers een tijdsvertraging op het gemeten druksignaal. Wanneer deze tijdsvertraging te groot wordt zal dit aanleiding geven tot een instabiele drukregeling. Bovendien geven ladingsverliezen in de leiding eveneens aanleiding tot valse metingen.

De druk wordt gemeten met absolute druksensoren (CPC100AFC van Honeywell). De uitgangsspanning die evenredig is met de druk wordt onmiddellijk door een verschilversterker versterkt en dan door een AD-converter (12-bit seriële output) omgezet in een digitaal signaal dat naar de microcontroller gaat. Door een dergelijke configuratie is de ruisopname tot een minimum beperkt.

Beide druksensoren zijn gekoppeld aan de *Queued Serial Peripheral Interface (QSPI)*, dit is een serieel bussysteem dat courant gebruikt wordt voor communicatie tussen chips en microcontrollers.



Figuur 2-18: druksensor met AD-converter

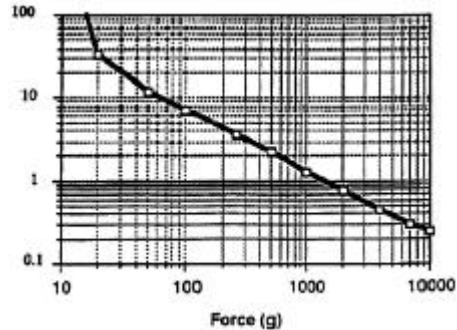
2.3.5 Krachtmeting voet

Force sensitive resistors (FSR) zijn weerstanden waarvan de weerstand daalt naargelang de druk die er wordt op uitgeoefend (Grafiek 2-5). Ze hebben weinig hysteresis en zijn ongevoelig voor warmte en schokken.

Ze worden onder andere gebruikt in pneumatische handen om te voelen wanneer er iets vastgepakt wordt. De Shadow Biped gebruikt FSR's voor een krachtmeting op de voet [12].

Door dergelijke sensoren aan tip en hiel te plaatsen van de voet kan men de positie van het Zero Moment Point (ZMP) opmeten. Dit is het punt waar het totale moment van alle reactiekrachten met de grond gelijk is aan nul. Valt dit punt buiten het steunvlak van de robot dan zal deze kantelen. De reactiekrachten worden bepaald door het zwaarteveld en de inertie-effecten. Bij stilstand komt dit

punt, daar er geen inertie-effecten zijn, overeen met het massamiddelpunt. Momenteel zijn deze sensoren nog niet geïnstalleerd. De elektronica kan overgenomen worden van de druksensoren waardoor ze kunnen gekoppeld worden aan de *Queued Serial Peripheral Interface (QSPI)*.



Grafiek 2-5: weerstand (W) daalt van FSR i.f.v. kracht

2.3.6 Safety Board

De maximum druk in de spieren bedraagt 4,5 bar absoluut. De voedingsdruk is echter 8 bar.

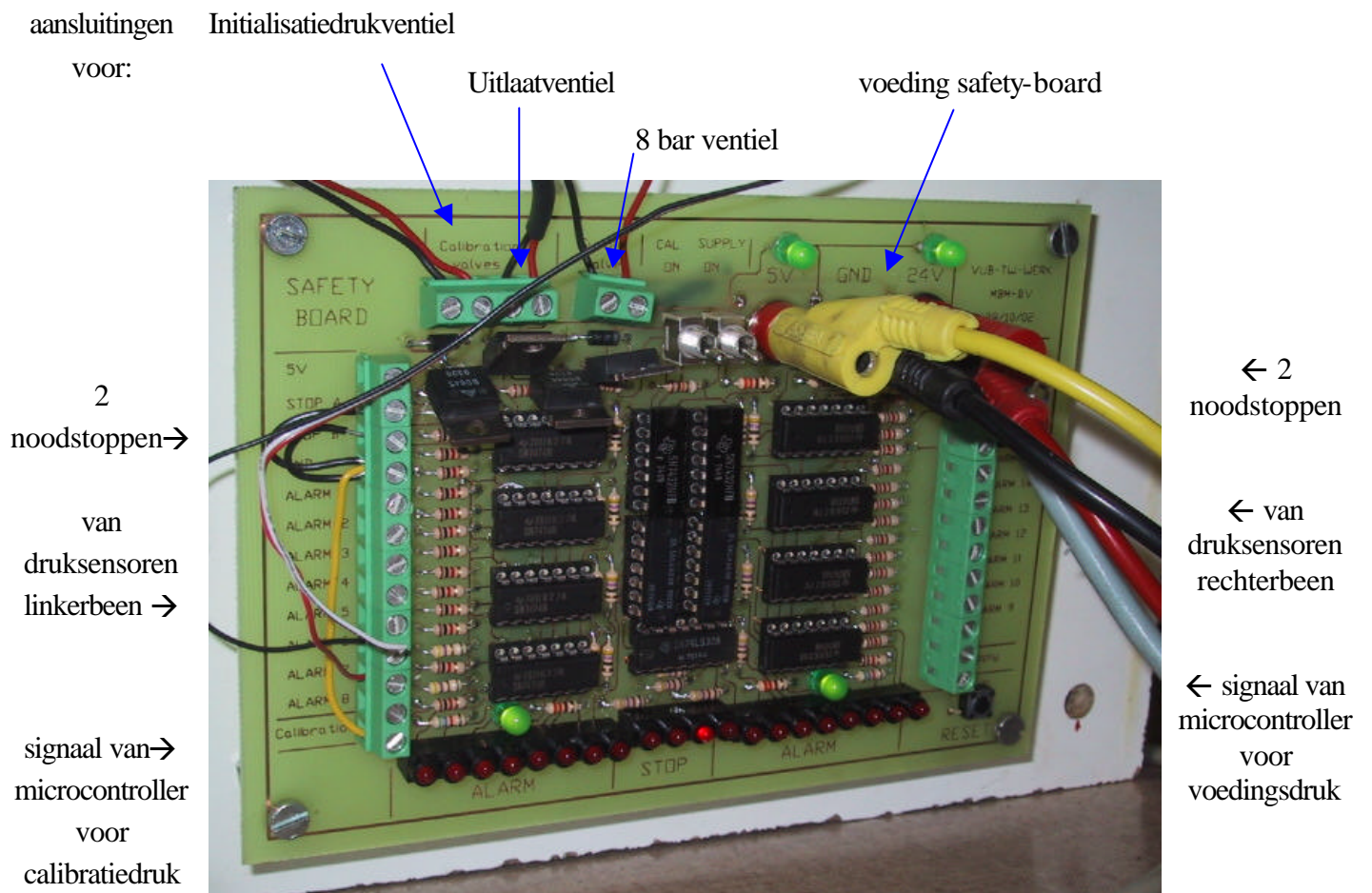
Bij normale werking wordt de druk in de spieren door de microcontroller begrensd tot 4 bar. Bij foutsituaties, zoals het vastlopen van de microcontroller, kan de voedingsklep blijven openstaan waardoor de spieren stuk kunnen gaan door een te hoge inwendige druk. Om dit te verhinderen is er een systeem, volledig onafhankelijk van de microcontrollers, die de voedingsdruk gaat sluiten en de leidingen gaat ontlichten indien er een druk wordt gedetecteerd van meer dan 4 bar in één van de spieren. Via een LED kan men nagaan in welke spier de fout zich heeft voorgedaan. Via de resetknop kan men de voedingsdruk terug herstellen.

Op deze *safety board* zijn eveneens 4 manuele noodstoppen aansluitbaar.

Een signaal van de master-microcontroller laat eveneens toe de voedingsdruk op te zetten of een lagere druk in te stellen.

Deze lagere druk wordt gebruikt om de druksensoren te calibreren, waarvan meer in het volgende hoofdstuk.

Er zijn dus drie modes waarin het systeem zich kan bevinden.



Figuur 2-19: safety board

Hoofdstuk 3

Controle

3.1 Inleiding

De controle van de robot gebeurt op twee niveaus: een controle op globaal niveau door de PC en een controle op lokaal niveau uitgevoerd door een microcontroller voor elk gewricht.

De PC produceert de trajecten die de robot moet volgen en de microcontroller zal aan de hand van de inputgegevens de druk in de spieren regelen om dit gevraagde traject te bereiken. Er is dus nood aan een snelle communicatie tussen de computer en de microcontrollers.

Dit hoofdstuk start dan ook met een bespreking van de communicatie, vervolgens wordt de globale controle en lokale controle besproken.

Vermits er zes gewrichten zijn, zijn er eveneens zes microcontrollers. Een zevende is verantwoordelijk voor de communicatie tussen PC en de microcontrollers te dirigeren.

3.2 Communicatie

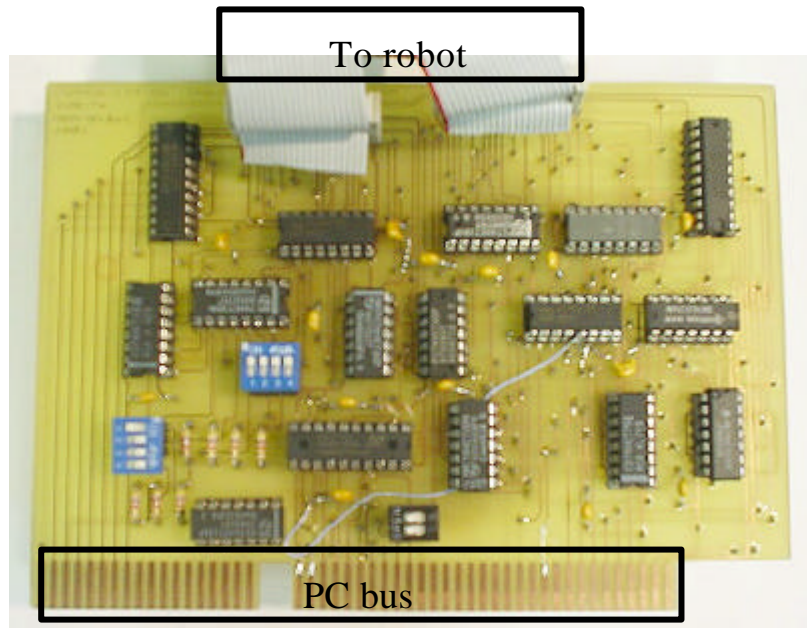
Omdat snelle communicatie tussen de PC en de microcontrollers absoluut noodzakelijk is werd een ISAbus-kaart voor de PC met bijhorend protocol ontworpen. De communicatiesnelheden van de datakaarten uit 2001 waren immers ontoereikend. Zo haalt de CANbus een bitrate van 1 Mbits/s indien de lengte van de kabel beperkt is tot 40 m. USB 1.0: 1,5 Mbit/s (low speed), USB 1.1: 12 Mbit/s (full speed)

De ISAbus-kaart haalt een snelheid van 34 Mbit/s.

Het recent ontwikkelde USB 2.0 is sneller (480 Mbit/s, high speed) en betrouwbaarder dan de ISAbus-kaart.

Voor het stappen is de ISAbus-kaart voldoende en werd dan ook deze weg verder uitgediept.

3.2.1 ISAbus-kaart

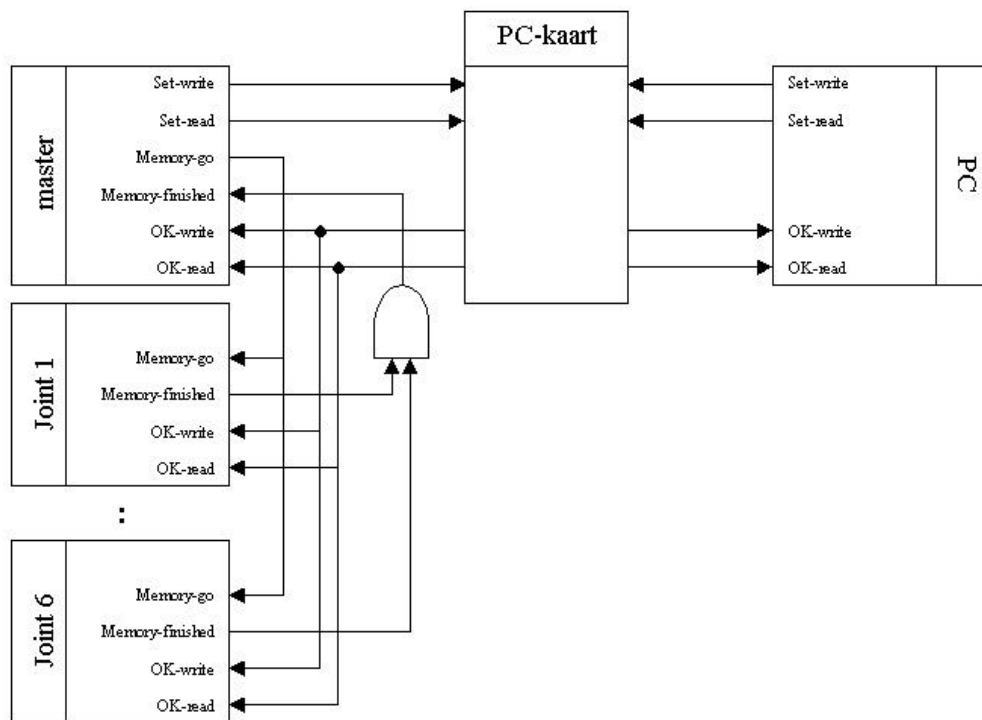


Figuur 3-1: ISAbus-kaart

Belangrijk in de communicatie is het gebruik van 16-bit dualported RAM als buffergeheugen tussen PC en microcontroller. Elke microcontroller heeft zijn eigen RAM-eenheid. Zowel de PC als de microcontroller kunnen data schrijven in het RAM-geheugen en data uitlezen. De RAM-eenheid laat niet toe om op hetzelfde moment een waarde te schrijven als te lezen. De PC en de microcontroller mogen wel op het zelfde moment beide een waarde lezen of beide een waarde schrijven. Er is dus een logica nodig die dat verwezenlijkt en deze bevindt zich op de ISAbus-kaart. De informatie-uitwisseling aangaande dit probleem gebeurt via enkele controlebits, waarvan het schema op figuur 3-2 staat.

Met de controlebits Set-Write en Set-Read kunnen zowel de master-microcontroller als de PC het datatransport vastpinnen op een bepaalde mode. Deze modes zijn:

- geen datatransport (zowel Set-Write als Set-Read bit hoog)
- schrijven in de RAM-eenheid (Set-Write hoog en Set-Read laag)
- lezen uit de RAM-eenheid (Set-Write laag en Set-Read hoog)
- initiatief overlaten aan de andere partij (Set-Write laag en Set-Read laag), dit betekent dat indien de PC deze beide bits laag houdt de microcontroller de master wordt.



Figuur 3-2: schema van de controlebits

In de toepassing van de stappende robot is de 7^{de} microcontroller steeds de master.

De RAM is verdeeld in twee: de ene helft staat in voor het datatransport van microcontroller naar PC (microcontroller schrijft en de PC leest) en de andere helft voor het datatransport van de PC naar de microcontroller (PC schrijft en de microcontroller leest).

De verdeling in twee wordt bekomen door de R/W-bit van de PC en microcontroller te verbinden met de hoogste adrespin van de RAM. Indien de PC data schrijft in een geheugenplaats van de RAM met een bepaald adres, dan moet de microcontroller daaruit lezen. Dit betekent dat de R/W-bit van de microcontroller met de negatie moet verbonden worden aan de hoogste adrespin van de RAM.

Indien er gegevens verzonden moet worden naar de RAM-eenheid dan zal de master-microcontroller de Set-Write-Bit hoog zetten. Indien er geen leesopdracht bezig is van de PC of microcontroller dan zal de PC-kaart de OK-Write-Bit hoog zetten. Dit signaal geeft aan de joint-microcontrollers de toelating om gegevens in de dualported RAM te schrijven.

Na 5 ms wordt de Set-Write-Bit terug laag gezet door de master-microcontroller.

Indien er waarden gelezen moeten worden uit de RAM-eenheid dan is er een gelijkaardig scenario, alleen wordt nu de Set-Read-Bit hoog gezet en wordt de validatie ervan gedaan door de OK-Read-Bit.

De Memory-Go-Bit en Memory-Finished-Bit worden in het begin gebruikt als trigger signaal om de verschillende initialisatie procedures goed te doen verlopen.

Ook de PC bezit de Set-Write-Bit en Set-Read-Bit tussen de PC en de ISAbus-kaart. Moet er van de computer data verzonden worden dan zal hij de Set-Write-Bit hoog zetten. Indien dit toegelaten is, dus Set-Read-Bit van microcontroller staat laag, dan zal de ISAbus-kaart de OK-Write-Bit hoog zetten. De microcontroller kan nu enkel zenden en niet ontvangen. De microcontroller kan enkel data ontvangen indien de PC de Set-Write-Bit terug laag zet.

De Set-Write-Bit en Set-Read-Bit generen eveneens een interrupt voor de PC op het kanaal IRQ5. IRQ staat voor interrupt request.

Bij een interrupt voert de PC een stukje code uit waarin hij eerst gaat controleren of hij moet lezen of schrijven. Naargelang de uitkomst gaat hij de dualported RAM van de zeven microcontrollers achtereenvolgens uitlezen of invullen. Hiervoor bevat de ISAbus-kaart twee incrementele tellers die enerzijds de RAM unit nummer telkens gaan verhogen (figuur 3-3, nr 5) en anderzijds het data-adres per RAM unit nummer (figuur 3-3, nr 6).

Het basisadres wordt ingevoerd bij de controle of het lezen of schrijven is.

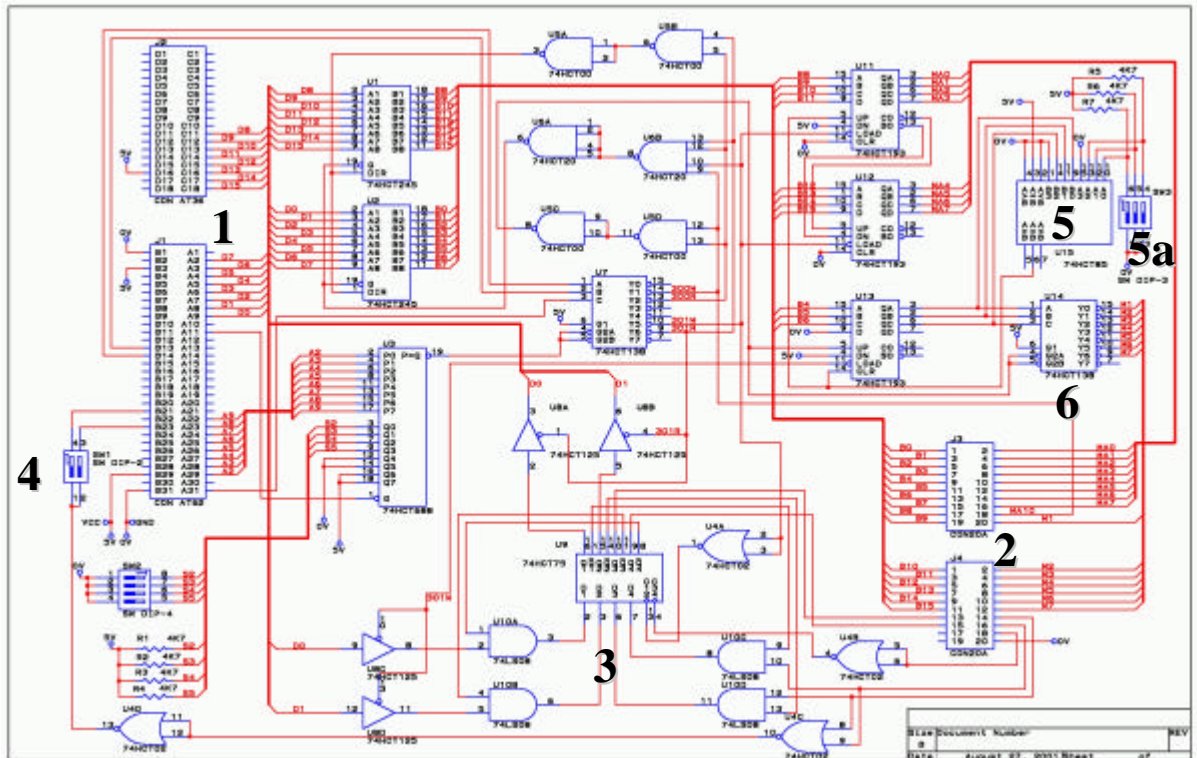
Stel als voorbeeld dat het basisadres X is, dan worden de microcontrollers als volgt beschreven of uitgelezen:

MC 1 adres X, MC 2 adres X,....

MC 1 adres X+1, MC 2 adres X+1,...

MC 1 adres X+2, MC 2 adres X+2,...

Op deze manier moet men maar éénmaal het adres ingeven en vervolgens kan men alle data verzenden, waardoor een snellere transmissie bekomen wordt.



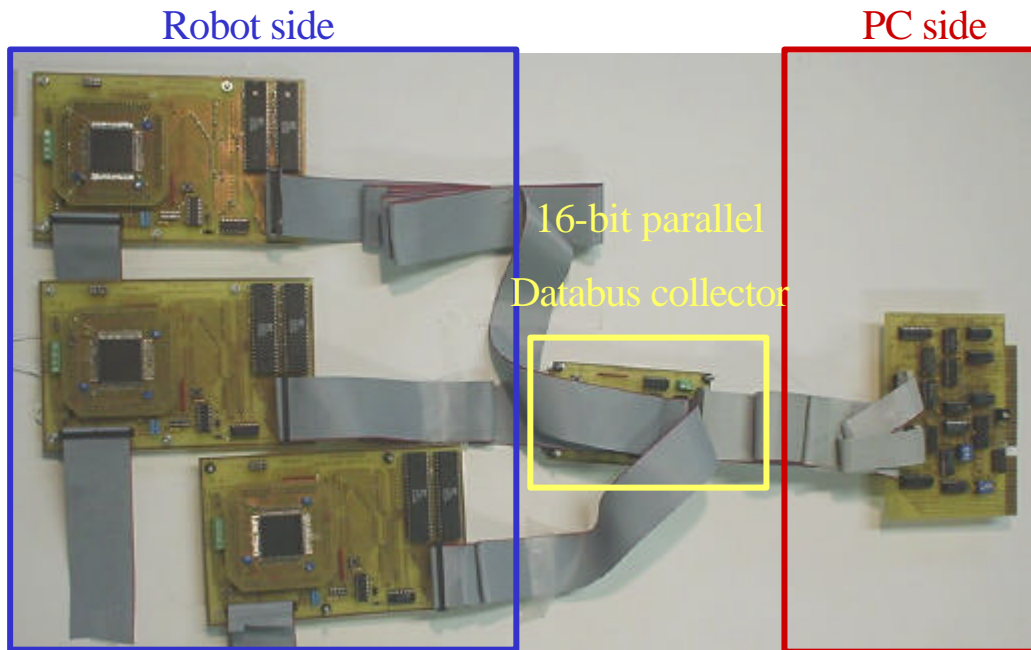
Figuur 3-3: schema ISAbus-kaart

- 1) ISAbus connector
- 2) Connector met de robot via databus collector
- 3) Logica controlebits
- 4) Schakelaars om interrupts op of af te zetten
- 5) Teller om RAM unit te verhogen met instelling via schakelaars (5a) hoeveel microcontrollers verbonden zijn (max 7)
- 6) Teller om geheugenadres te verhogen

De pin-outs van de connectoren staan in bijlagen.

3.2.2 Databus collector

De fysieke verbinding van de microcontrollers met de ISAbus-kaart gebeurt met flatcables. Deze worden verzameld op een 16-bit parallel Databus Collector. Deze bevat ook een reset-knop om alle microcontrollers tegelijkertijd te resetten.



Figuur 3-4: fysieke verbinding microcontrollers met ISAbus-kaart via databus collector

3.2.3 Driver

De programmatie van de controle op het hoogste niveau is in een Windows-omgeving. Windows vereist dat elk programma platform onafhankelijk moet zijn. Dus elke hardware afhankelijkheid is volledig incompatibel met het operating system. Om de link te maken tussen hardware en software zijn er drivers nodig.

Via een dergelijke driver is het mogelijk een bibliotheek te maken die de low-level toegang tot de hardware mogelijk maakt, terwijl het high-level programma platform onafhankelijk blijft.

De programmatie van de driver werd gedaan met de hulp van Windriver, een programma van Jungo [24]. De driver kan enerzijds lezen en schrijven in de I/O-adressen (hexadecimaal) 300 en 301 van de ISAbus en IRQ 5 detecteren.

Hierin dient adres 301 voor de controle of er gelezen of geschreven moet worden. De eerste 8 bits bevatten het beginadres van de RAM en de laatste 2 bits zijn de Read-Bit en Write-Bit en adres 300 wordt gebruikt voor de eigenlijke datatransport.

De belangrijkste instructies uit de bibliotheek zijn:

INTERRUPTLISTENER_Open: openen driver om interrupt te detecteren
INTERRUPTLISTENER_Close: sluiten driver om interrupt te detecteren
INTERRUPTLISTENER_Interrupt5Enable
: reageren bij interrupt aanzetten
INTERRUPTLISTENER_Interrupt5Disable
: reageren bij interrupt afschakelen
INTERRUPTLISTENER_Interrupt5IsEnabled
: nakijken of interruptreactie aan staat
SUPERDRIVER_Open
: openen driver toegang tot adressen 300 en 301
SUPERDRIVER_Close
: sluiten driver toegang tot adressen 300 en 301
SUPERDRIVER_ReadWord
: ontvangen van 16-bits getal
SUPERDRIVER_WriteWord
: zenden van 16-bits getal

Men moet wel nakijken of de gebruikte adressen en IRQ-kanaal niet reeds bezet zijn door een ander toestel. Desnoods moet men deze verplaatsen naar een ander adres.

Sommige kanalen zijn niet toegankelijk zoals deze voor de muis (IRQ 12), toetsenbord (IRQ 1), ...

3.2.4 Priority-level

De snelheid van communiceren hangt zowel af van de kloksnelheid van de computer als van de priority van het programma.

Windows is een multitasking systeem. De processor kan echter slechts één taak tegelijkertijd uitvoeren, terwijl er verschillende taken wachten om uitgevoerd te worden. Om te bepalen welke taak er het eerst wordt uitgevoerd krijgen de verschillende taken van de scheduler een priority. Dit is een getal van 1 tot 31, waarvan de taak met het hoogste getal het eerst zal worden uitgevoerd. De

behering van het toewijzen van de priority's berust bij de scheduler. Een taak van het programma waarmee de gebruiker op dat moment bezig is zal bijvoorbeeld een hogere priority krijgen dan een programma dat op dat moment door de gebruiker niet wordt gebruikt.

De programmeur kan echter ook de priority's aanpassen. Een programma voor bijvoorbeeld de schermbeveiliging zal een zeer lage priority krijgen, terwijl real-time programma's een hoge priority toegedeeld moeten krijgen voor een goede werking.

In het programma dat de communicatie verzorgt met de robot is een dergelijke verhoging van de priority noodzakelijk om op een snelle manier te communiceren.

De basis priority level wordt bepaald door zowel de prioriteitsklasse (priority class) als een prioriteitswaarde (priority value).

Door in te spelen op deze twee parameters verkrijgt men 31 mogelijkheden die vermeld staan in bijlage.

Wanneer er geen communicatie is wordt er een 7 toegekend aan het programma (*NORMAL_PRIORITY_CLASS THREAD_PRIORITY_NORMAL*), dit is wat een programma standaard gebruikt.

Van zodra de gebruiker de communicatie met de robot start wordt dit verhoogd naar 26 (*REALTIME_PRIORITY_CLASS THREAD_PRIORITY_HIGHEST*)

Er kan nog verhoogd worden naar 31 (*REALTIME_PRIORITY_CLASS THREAD_PRIORITY_TIME_CRITICAL*) Dit wordt in de literatuur echter afgeraden omdat het systeem dan instabiel kan worden.

Door de verhoging van de priority is de tijd tot eerste reactie na interrupt teruggedrongen van 60 ms naar 600 μ s.

Momenteel worden er om de 20 ms nieuwe stelwaarden gezonden en meetresultaten naar de computer gestuurd.

De gezonden waarden zijn: stelwaarde hoek en waarden voor het instellen van de regelaar.

De ontvangen waarden zijn: stelwaarde hoek (controle of hoek goed ontvangen is), werkelijk hoek, berekend drukverschil van regelaar, stelwaarde druk 1 en 2, werkelijk druk 1 en 2.

Per seconde ontvangt de PC dus 2450 16-bit gegevens.

3.2.5 Fouten in de communicatie

Vermits een goede communicatie tussen de computer en de microcontrollers onontbeerlijk is voor het goed functioneren van de robot, werden verschillende communicatietesten uitgevoerd.

Bij het maken van een communicatietest werd getracht volgende punten te onderzoeken:

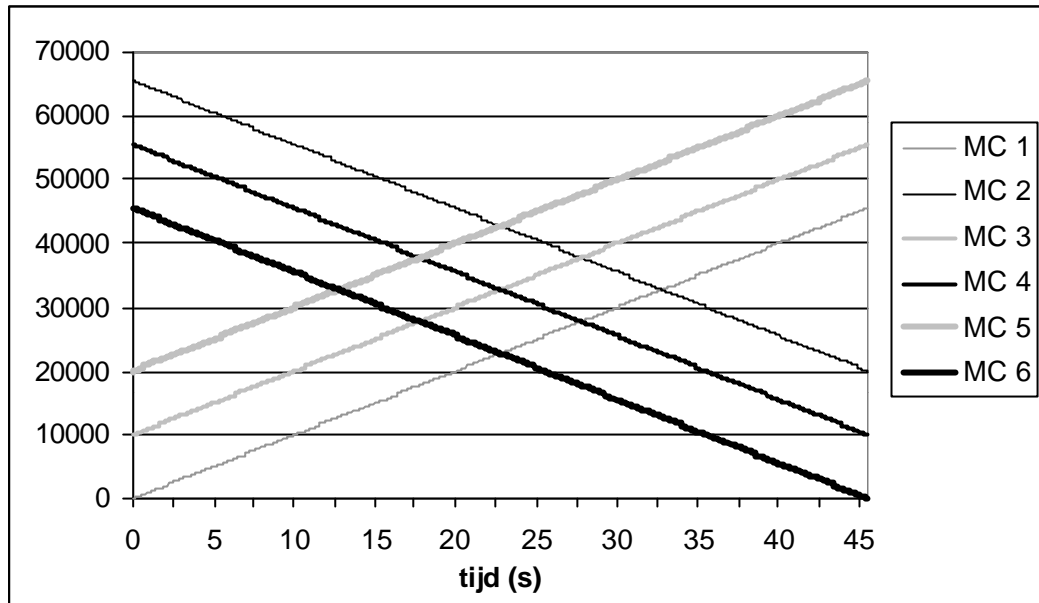
1. Zit de fout in de richting van computer naar robot of van robot naar computer
2. De lengte van de communicatielijn is tamelijk lang, zorgen capacitaire effecten voor fouten
3. Zijn individuele data- of adreslijnen slecht verbonden
4. Komt de data in het juiste adres terecht.

De uitgewerkte test bestond erin een tabel met testwaarden door te zenden van de PC naar de microcontrollers. Het enige wat de microcontrollers deden was die waarden verschillende keren terugzenden naar de PC. Deze ontvangen waarden werden dan geregistreerd. Hiermee kan men zien of de fout van type 1 is.

De tabel met testwaarden bestaat uit alle waarden tussen 0 en 2^{16} , zou er dan een zekere regelmatigheid zitten in de fouten dan kan men hieruit concluderen dat een bepaalde datacommunicatielijn onderbroken is.

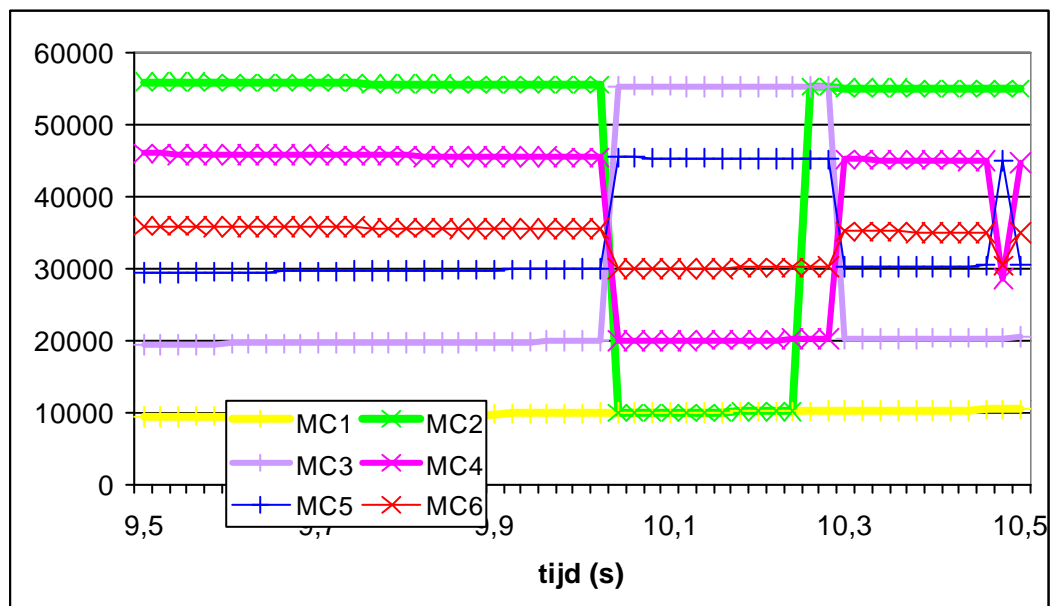
De tabel met testwaarden werd niet incrementeel doorgezonden. Maar om de communicatielijnen steeds op te laden en te ontladen werd eerst een 0 (hex: 0000) doorgezonden, vervolgens 65535 (hex: FFFF) waardoor alle lijnen worden opgeladen, terug ontladen door een kleine waarde 10000 door te zenden die verschillend is van 0, ... Omdat elke geheugenplaats dan een totaal verschillende waarde krijgt, kan men gemakkelijk nagaan of de data in de juiste adressen terecht zijn gekomen.

Er werd dus volgende tabel als testtabel gebruikt.



Grafiek 3-1: testtabel met testwaarden om communicatie te controleren

Indien de resultaten worden geanalyseerd valt er op te merken dat op gezette tijden de waarde bestemd voor de vorige microcontroller wordt ontvangen.



Grafiek 3-2: deel van de ontvangen waarden

De oorzaak hiervan is nog niet achterhaald. Het ontbreken van oscilloscopen met hoge tijdsresolutie om te meten in de computer zelf verhindert het onderzoek hiernaar.

Om dit probleem te omzeilen worden de data tweemaal doorgezonden.

Indien volgende tabel de gewenste data-uitwisseling voorstelt.

	Master	Heup R	Knie R	Enkel R	Heup L	Knie L	Enkel L
Adres : 0000	-	1	2	3	4	5	6

Dan gaat men deze als volgt communiceren.

	Master	Heup R	Knie R	Enkel R	Heup L	Knie L	Enkel L
Adres: 0000	1	1	3	3	5	5	5
Adres: 0002	2	2	2	4	4	6	6

De microcontrollers kijken in de grijze geheugenplaatsen. Treedt er nu een verschuiving op dan ontvangen de microcontrollers nog steeds de juiste waarde.

Dit is natuurlijk niet een optimale oplossing daar de communicatiesnelheid met de helft wordt gereduceerd. Door het ontbreken van de tijd, de moeilijkheid om het probleem te onderzoeken en de wetenschap dat er in de toekomst met USB 2.0 zal worden gecommuniceerd werd besloten dit probleem voorlopig niet verder te onderzoeken.

De lengte van de lijnen heeft ook een negatieve invloed, indien deze te lang wordt dan nemen de communicatiefouten aanzienlijk toe. De fouten zitten vooral in het stuk van de robot naar de computer en hiervan ondervindt de robot geen invloed. De communicatiefouten uit zich in scherpe pieken op de grafiek.

3.2.6 USB

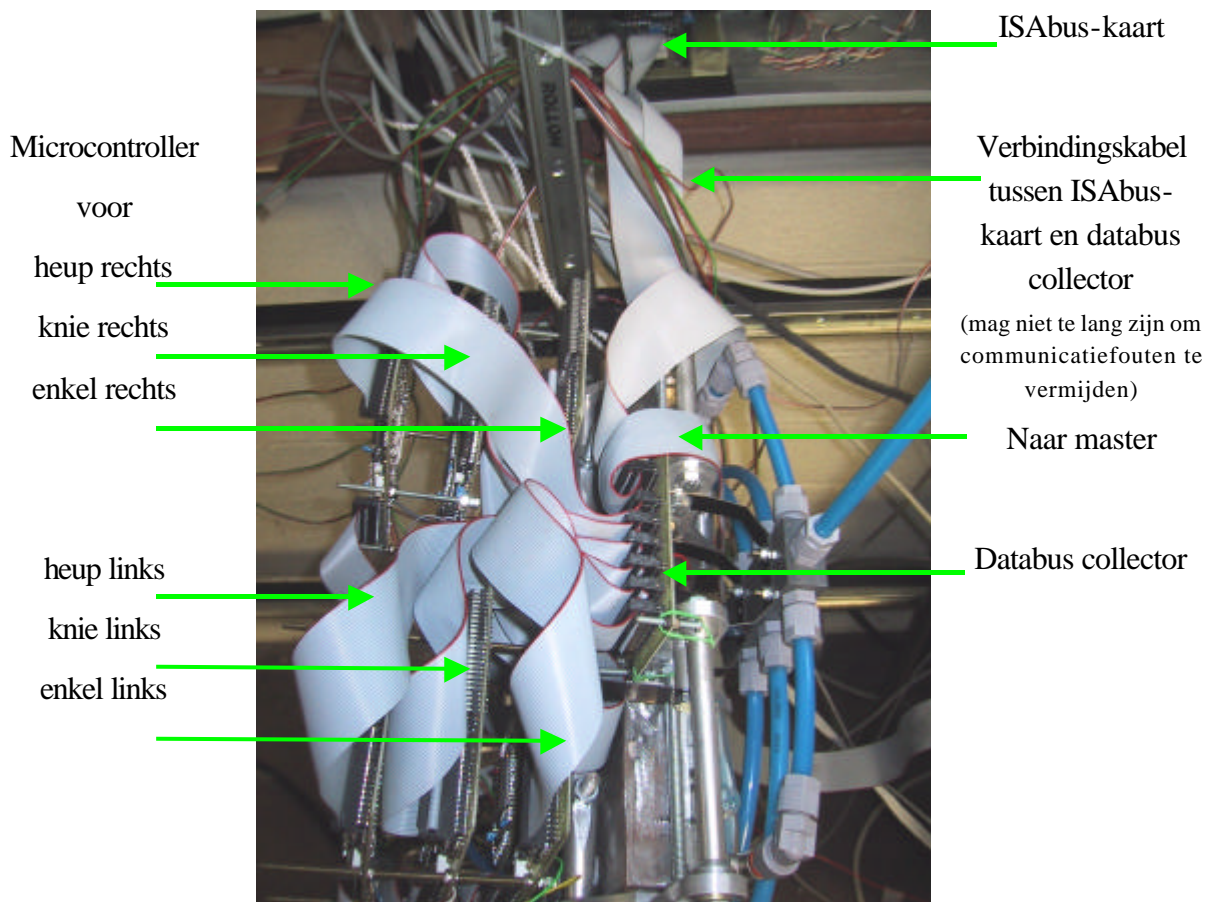
Het recent ontwikkelde USB 2.0-protocol is sneller (480Mbps) ten opzichte van de ISAbus-kaart (34 Mbps). De werkelijke datasnelheid hangt af van hoe druk de bus gebruikt wordt.

USB heeft als voordeel dat de meeste intelligentie in de computer steekt zodanig dat de randapparatuur eenvoudig en goedkoop kan gehouden worden.

Bijkomende voordelen zijn dat dit een seriële communicatie is, in plaats van een parallelle communicatie. Er werd immers vastgesteld dat indien men de kabel tussen PC en robot langer maakt dan 1,5 meter er teveel fouten sluipen in de communicatie. Deze beperkte lengte stelt problemen bij het stappen. Bij USB 2.0 mag men daarentegen de lengte opvoeren tot 5 m (d.m.v. 5 hubs 30 m). De draden zijn ook zeer flexibel.

Een laatste nadeel van de ISAbus-kaart is dat deze niet meer kan ingeplugd worden in de modernste computers omdat deze computers niet meer worden uitgerust met een ISAbus slot.

De Development Kit van de EZ-USB® FX2™-familie van Cypress bevat een microcontroller uitgerust met het USB 2.0 protocol. De chip bezit reeds de nodige intelligentie om te reageren op de verschillende events op de USB-poort. Hij kan zowel geprogrammeerd worden in C als assembler.



Figuur 3-5: bovenaanzicht robot met microcontrollers

3.3 Globale controle

De globale controle is opgesplitst in 2 programma's. Het eerste programma genereert het traject en slaat dit op in een Excel-bestand. Een tweede programma gaat de data van dit Excel-bestand zenden naar de robot en de meetresultaten opslagen in een nieuw Excel-bestand.

Omdat Excel een geschikt programma is om meetresultaten te verwerken en te visualiseren is er een sterke link mee.

3.3.1 Trajectgeneratie

Er zijn 2 mogelijkheden om een traject te generen uitgewerkt. Via discrete posities of via objectieve parameters.

In het eerste geval wordt de stand van de robot voor een aantal discrete posities in de tijd door de gebruiker vastgelegd. De computer berekent dan via een spline de tussenliggende punten.

Een tweede mogelijkheid is dat de gebruiker de objectieve parameters van de stap ingeeft en dat de computer hieruit het traject berekent [19].

De objectieve parameters zijn begin-, midden- en eindpositie van de enkels en de heup met de bijkomende eis dat in het begin en op het einde van de stap de snelheid nul moet zijn, dit om schokken bij het neerkomen in de testfase te vermijden.

Indien $x_r(t), y_r(t)$ de opeenvolgende posities voorstellen van de rechterenkel dan is $x_r(0), y_r(0)$ de beginpositie, $x_r(t_{\text{stap}}/2), y_r(t_{\text{stap}}/2)$ de middenpositie en $x_r(t_{\text{stap}}), y_r(t_{\text{stap}})$ de eindpositie met t_{stap} de tijdsduur van de stap die gekend zijn.

Volgende polynomen met onbekenden a, b, c, d, e en f, g, h, i, j worden gezocht:

$$x_r(t) = a.t^4 + b.t^3 + c.t^2 + d.t + e$$

$$y_r(t) = f.t^4 + g.t^3 + h.t^2 + i.t + j$$

Dit is een probleem met 2x5 voorwaarden in 2x5 onbekenden.

In matrixnotatie geeft dit.

$$\begin{bmatrix} x_r(0) \\ x_r\left(\frac{t_{stap}}{2}\right) \\ x_r(t_{stap}) \\ \dot{x}_r(0) \\ \dot{x}_r(t_{stap}) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ \left(\frac{t_{stap}}{2}\right)^4 & \left(\frac{t_{stap}}{2}\right)^3 & \left(\frac{t_{stap}}{2}\right)^2 & \left(\frac{t_{stap}}{2}\right) & 1 \\ t_{stap}^4 & t_{stap}^3 & t_{stap}^2 & t_{stap} & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 4t_{stap}^3 & 3t_{stap}^2 & 2t_{stap} & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix}$$

$$\begin{bmatrix} y_r(0) \\ y_r\left(\frac{t_{stap}}{2}\right) \\ y_r(t_{stap}) \\ \dot{y}_r(0) \\ \dot{y}_r(t_{stap}) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ \left(\frac{t_{stap}}{2}\right)^4 & \left(\frac{t_{stap}}{2}\right)^3 & \left(\frac{t_{stap}}{2}\right)^2 & \left(\frac{t_{stap}}{2}\right) & 1 \\ t_{stap}^4 & t_{stap}^3 & t_{stap}^2 & t_{stap} & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 4t_{stap}^3 & 3t_{stap}^2 & 2t_{stap} & 1 & 0 \end{bmatrix} \begin{bmatrix} f \\ g \\ h \\ i \\ j \end{bmatrix}$$

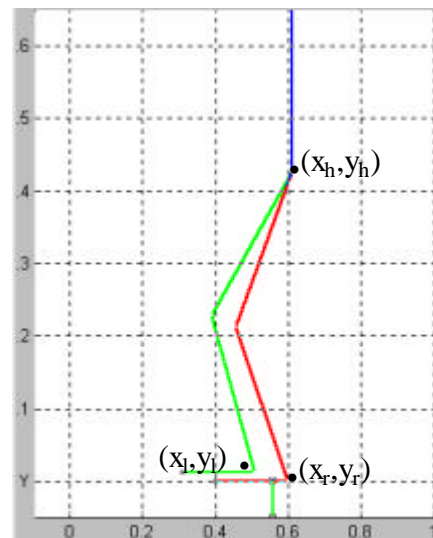
Deze matrix wordt geïnverseerd om de onbekende a, b, c, d, e en f, g, h, i, j te vinden.

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ \left(\frac{t_{stap}}{2}\right)^4 & \left(\frac{t_{stap}}{2}\right)^3 & \left(\frac{t_{stap}}{2}\right)^2 & \left(\frac{t_{stap}}{2}\right) & 1 \\ t_{stap}^4 & t_{stap}^3 & t_{stap}^2 & t_{stap} & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 4t_{stap}^3 & 3t_{stap}^2 & 2t_{stap} & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} x_r(0) \\ x_r\left(\frac{t_{stap}}{2}\right) \\ x_r(t_{stap}) \\ \dot{x}_r(0) \\ \dot{x}_r(t_{stap}) \end{bmatrix}$$

$$\begin{bmatrix} f \\ g \\ h \\ i \\ j \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ \left(\frac{t_{stap}}{2}\right)^4 & \left(\frac{t_{stap}}{2}\right)^3 & \left(\frac{t_{stap}}{2}\right)^2 & \left(\frac{t_{stap}}{2}\right) & 1 \\ t_{stap}^4 & t_{stap}^3 & t_{stap}^2 & t_{stap} & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 4t_{stap}^3 & 3t_{stap}^2 & 2t_{stap} & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} y_r(0) \\ y_r\left(\frac{t_{stap}}{2}\right) \\ y_r(t_{stap}) \\ \dot{y}_r(0) \\ \dot{y}_r(t_{stap}) \end{bmatrix}$$

Het traject van de rechterenkel ($x_r(t), y_r(t)$) is nu volledig gekend, op dezelfde wijze bekomt men deze voor de linkerenkel en de heup.

Indien men eist dat de hoek van de voeten en het bovenlichaam steeds dezelfde moeten zijn (bv voeten steeds horizontaal en bovenlichaam vertikaal zoals op nevenstaande figuur) dan kan men met geometrische formules voor elk moment de overeenkomstige hoeken bereken van enkel, knie en heup.



Figuur 3-6: staafmodel robot

3.3.2 Feedforward

Oorspronkelijk was het de bedoeling om een dubbele feedforward in de trajectcontrole in te bouwen. De eerste feedforward berekent de koppels nodig in de gewrichten, de tweede feedforward gaat uitgaande van de koppelkarakteristieken van de spieren en de gewenste koppels de nodige drukken bereken.

Met de gewenste hoeken wordt dus een schatting van de drukken naar de robot gezonden.

Vermits de resultaten zonder deze ingreep reeds bevredigend waren en een goede modelisatie van de robot nog niet beschikbaar was, is deze piste niet verder uitgewerkt. In de toekomst, vooral als er sneller zal gestapt worden, zal de PID regelaar vermoedelijk niet meer volstaan en zal een verdere uitdieping van de feedforward-methode nodig zijn.

De redenering achter de feedforward is als volgt:

De bewegingsvergelijkingen van een robot kunnen altijd geschreven worden onder de vorm:

$$H\ddot{q} + C(\dot{q}, q)\dot{q} + g(q) = \tau$$

Matrix H is de massamatrix, C bevat de coriolis- en centrifugaaltermen, g is te wijten aan de gravitatie en vector τ bestaat uit alle overige krachten/koppels (actuatie, wrijving, ...). q zijn de veralgemeende coördinaten.

Onderstel dat de beweging quasi-statisch gebeurt, dan mag men de traagheidskrachten en corioliskrachten verwaarlozen.

Indien men nu voor de actuatie $\tau = -K_p \tilde{q} - K_D \dot{\tilde{q}} - K_I \int_0^t \tilde{q} dt + \hat{g}$ neemt.

Met matrices K_p , K_I en K_D diagonaalmatrices met respectievelijk de proportionele, integrale en de differentiële factoren op de diagonalen.

\tilde{q} de fout op de positie ($q_{\text{werkelijk}} - q_{\text{gewenst}}$).

\hat{g} een model voor de gravitatie.

Als rustfout krijgt men dan:

$$\tilde{q} = K_p^{-1} (g - \hat{g})$$

Indien het model perfect is dan is de eindfout steeds nul, anders is ze evenredig met de fout op het model. Deze imperfecties van het model worden dan weggeregeld door de aanwezige PID regelaar.

De reden waarom dit noodzakelijk zal worden is dat zowel de belasting als de koppelkarakteristiek niet lineair zijn en dus voor de lokale PID regelaar moeilijk te regelen valt. Bedoeling is dat samen met de gewenste hoek ook een schatting van de nodige druk voor strek- en buigspier worden gestuurd naar de lokale controller. Deze zal dan aan de hand van een lokale PID regelaar de resterende fout kunnen weggeregelen.

De drukken worden als volgt berekend:

Het traject wordt door de gebruiker bepaald. De computer berekent voor elke positie de koppels (M) in de gewrichten. Hij kan dit doen indien de massa's en posities van de massamiddelpunten gekend zijn.

Men heeft dus als gekende parameters M en \mathbf{a} ; m_1 en m_2 liggen eveneens vast door de keuze van de spieren en de aanhechtingpunten.

Uit de formule $M = p_2 m_2(\mathbf{a}) - p_1 m_1(\mathbf{a})$ zijn p_1 en p_2 de overblijvende onbekenden.

Een tweede vergelijking is nodig om het probleem te kunnen oplossen, dit is de stijfheid [11].

$$K(=C^{-1}) = \frac{dM}{d\mathbf{a}} = \frac{d(M_2 - M_1)}{d\mathbf{a}} = \frac{dp_2}{d\mathbf{a}} m_2 + p_2 \frac{dm_2}{d\mathbf{a}} - \frac{dp_1}{d\mathbf{a}} m_1 - p_1 \frac{dm_1}{d\mathbf{a}}$$

Hierin stellen de termen $dp_i/d\mathbf{a}$ de verandering in stijfheid voor van veranderende druk in functie van de contractie, dewelke wordt beheerst door de actie van de kleppen en de thermodynamische processen van de lucht in de spier. Onderstelt men dat de kleppen gesloten blijven en beschouwt men isentropische compressie dan kan men volgende formule gebruiken.

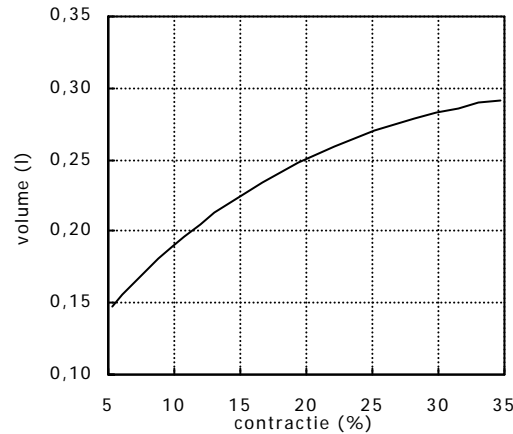
$$P_i V_i^g = P_{i0} V_{i0}^g \text{ met } P_i = P_{atm} + p_i$$

Met P_i en V_i absolute druk en volume van spier i ; P_{i0} , V_{i0} de absolute initiële druk en volume wanneer de kleppen van spier i gesloten zijn; p_i en p_{i0} de relatieve druk en de initiële relatieve druk; g de isentropische coëfficiënt en P_{atm} de atmosferische druk.

Vermits P_{atm} een constante is verkrijgt men.

$$\frac{dp_i}{d\mathbf{a}} = \frac{dP_i}{d\mathbf{a}} = -\mathbf{g}(P_{atm} + p_{i_0}) \cdot \frac{V_{i_0}^g}{V_i^g} \cdot \frac{dV_i}{d\mathbf{a}}$$

Onderstaande grafiek toont het volume van beide spieren in functie van de kniehoek.



Grafiek 3-3: berekend volume (ml) i.f.v. de hoek voor beide kniespieren

Uitgaande van deze grafiek en de koppelkarakteristiek kan men informatie bekomen aangaande het teken van $dm_i/d\mathbf{a}$ en $dV_i/d\mathbf{a}$.

De formule van de stijfheid wordt dan :

$$K(= C^{-1}) = -(k_1 p_{1_0} + k_2 p_{2_0} + k_{atm} P_{atm})$$

$$k_1 = m_1 \mathbf{g} \frac{V_{1_0}^g}{V_1^{g+1}} \left| \frac{dV_1}{d\mathbf{a}} \right| + \left| \frac{dm_1}{d\mathbf{a}} \right| > 0$$

$$\text{met } k_2 = m_2 \mathbf{g} \frac{V_{2_0}^g}{V_2^{g+1}} \left| \frac{dV_2}{d\mathbf{a}} \right| + \left| \frac{dm_2}{d\mathbf{a}} \right| > 0$$

$$k_{atm} = k_1 + k_2 - \left| \frac{dm_1}{d\mathbf{a}} \right| - \left| \frac{dm_2}{d\mathbf{a}} \right|$$

k_1 , k_2 , k_{atm} zijn dus afhankelijk van de hoek \mathbf{a} .

Uitgaande van

$$\begin{cases} M = p_2 m_2(\mathbf{a}) - p_1 m_1(\mathbf{a}) \\ K(= C^{-1}) = -(k_1 p_{1_0} + k_2 p_{2_0} + k_{atm} P_{atm}) \end{cases}$$

kan men p_1 en p_2 berekenen. Merk op dat het moment een gewogen verschil is van de drukken en de stijfheid een gewogen som van de drukken.

$$\begin{cases} p_1 = \frac{M \cdot k_2 + K \cdot m_2 + k_{atm} \cdot m_2 \cdot P_{atm}}{-m_1 \cdot k_2 - m_2 \cdot k_1} \\ p_2 = \frac{M \cdot k_1 - K \cdot m_1 - k_{atm} \cdot m_1 \cdot P_{atm}}{m_1 \cdot k_2 + m_2 \cdot k_1} \end{cases}$$

Samen met \mathbf{a} worden deze schattingen van drukken p_1 en p_2 naar de microcontroller gezonden.

De microcontroller genereert dan de drukken $(p_1 + \Delta p)$ en $(p_2 - \Delta p)$ in de spieren met Δp een waarde van de lokale PID regelaar om de resterende hoekfout weg te regelen.

Er moet eveneens rekening gehouden worden met de beperkte toelaatbare drukken in de spieren. Deze moet tussen 0,1 en 3 bar (relatief) liggen.

Indien bv $p_1 > 3$

Het koppel uitgeoefend door een gewricht wordt gegeven door:

$$M = M_2 - M_1 = p_2 m_2(\mathbf{a}) - p_1 m_1(\mathbf{a})$$

Indien men niet de druk p_1 instelt in spier 1, maar een druk van 3 bar relatief, dan verlaagt het koppel uitgeoefend door spier 1 met $(p_1 - 3) \cdot m_1(\mathbf{a})$, dus wil men hetzelfde koppel M behouden moet men dit ook aftrekken bij het koppel van spier

$$2: M_{2nieuw} = M_2 - (p_1 - 3) \cdot m_1(\mathbf{a})$$

De drukken worden dan:

$$p_{2nieuw} = \frac{M_{2nieuw}}{m_2(\mathbf{a})} = \frac{M_2 - (p_1 - 3) \cdot m_1(\mathbf{a})}{m_2(\mathbf{a})} = p_2 - \frac{(p_1 - 3) \cdot m_1(\mathbf{a})}{m_2(\mathbf{a})}$$

$$p_{1nieuw} = 3$$

Indien bv $p_1 < 0,1$

Indien men niet de druk p_1 instelt in spier 1, maar een druk van 0,1 bar relatief, dan verhoogt het koppel uitgeoefend door spier 1 met $(0,1 - p_1) \cdot m_1(\mathbf{a})$, dus wil men hetzelfde koppel M behouden moet men dit ook bijtellen bij het koppel van spier 2: $M_{2nieuw} = M_2 + (0,1 - p_1) \cdot m_1(\mathbf{a})$

De drukken worden dan:

$$p_{2nieuw} = \frac{M_{2nieuw}}{m_2(\mathbf{a})} = \frac{M_2 + (0,1 - p_1) \cdot m_1(\mathbf{a})}{m_2(\mathbf{a})} = p_2 + \frac{(0,1 - p_1) \cdot m_1(\mathbf{a})}{m_2(\mathbf{a})}$$

$$p_{1nieuw} = 0,1$$

Gevolg hiervan is dat het moment correct zal blijven maar dat de stijfheid niet meer de gevraagde waarde zal zijn.

3.3.3 Programmatie trajectgeneratie

Om de trajectgeneratie op een eenvoudige en gebruiksvriendelijke manier te laten verlopen is er in Matlab een grafische user interface (GUI) ontwikkeld.

Om het traject te definiëren zijn er twee pistes uitgewerkt: ofwel definieert men discrete posities ofwel berekent men het traject aan de hand van objectieve parameters.

3.3.3.1 Trajectprogramma a.d.h.v. discrete posities

Via het bewegen van glijknoppen of het invullen van hoeken in de invulvelden wordt de robot in een bepaalde positie geplaatst die wordt gevisualiseerd op de linkse tekening, zoals getoond op onderstaande screenshot. Indien deze stand voldoet kan men deze positie opslagen.

Om na te gaan of de robot niet voorover of achterover valt is de X-positie van het zwaartepunt getekend die rood kleurt indien deze buiten de voet valt van het ondersteunende been en groen wanneer deze binnen de voet valt.

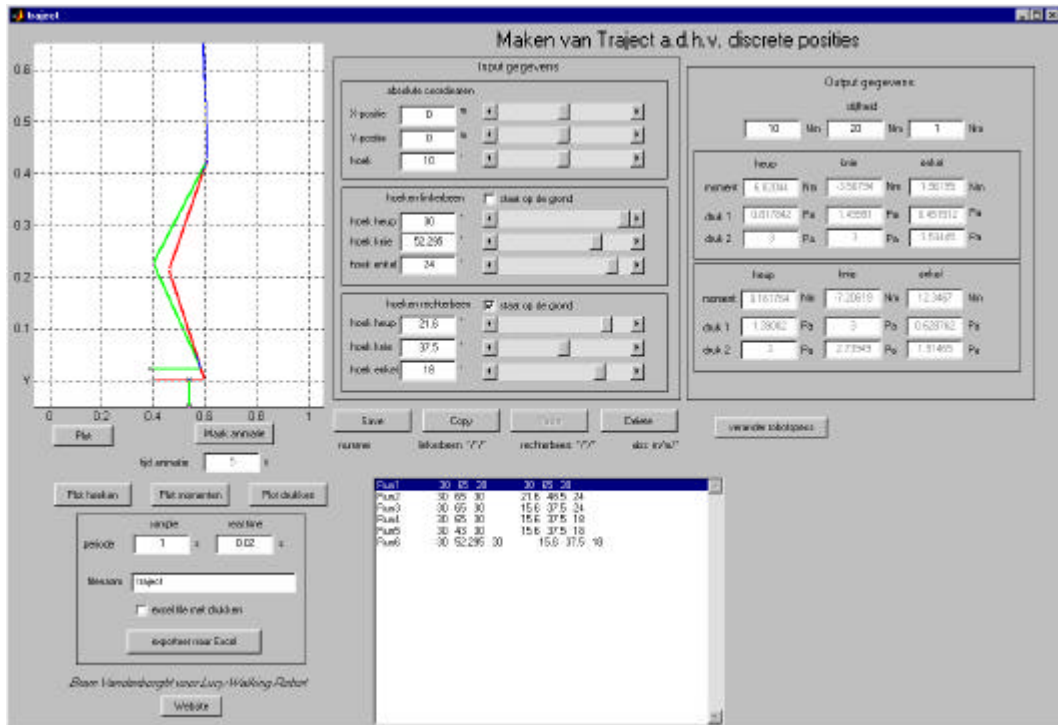
Men moet dit doen voor de verschillende opeenvolgende posities.

Stel dat men deze standen wenst voor elke halve seconde. De microcontroller moet echter elke 20 ms een nieuwe stelwaarde ontvangen. Deze 2 getallen moet men dan invullen en de computer berekent dan via een spline de waarde van de tussenliggende punten.

Via de knop “animatie” krijgt men een animatie van de beweging van de robot.

Door het drukken op de knop “Exporteer naar Excel” wordt dan een Excel-bestand gemaakt met de opeenvolgende hoeken en eveneens een schatting van de drukken in de spieren.

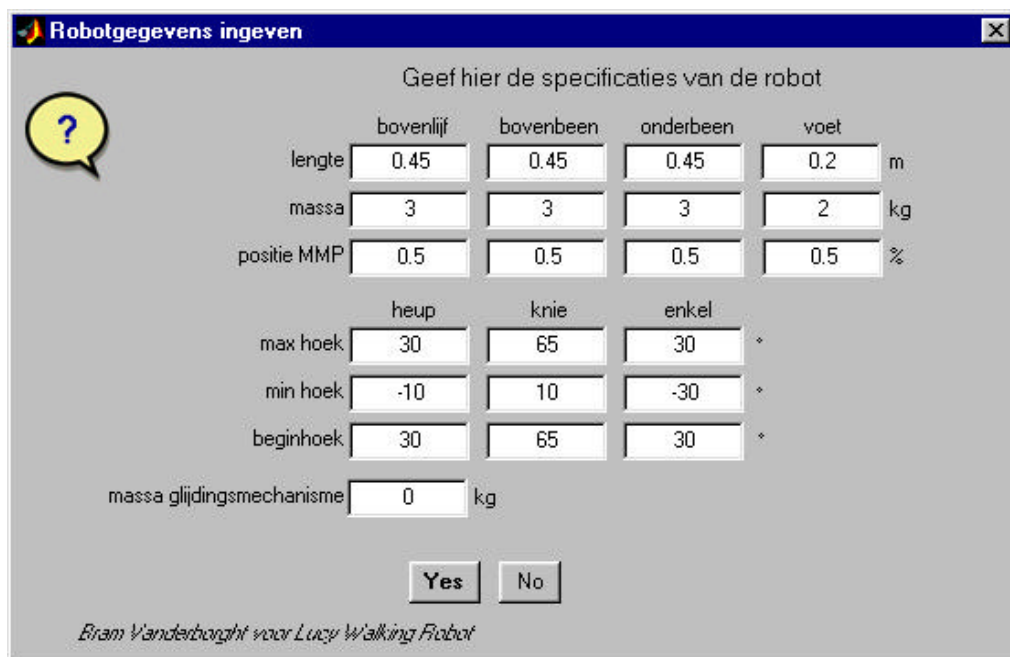
Twee grafieken geven eveneens het verloop van de hoeken in de tijd weer met erop aangeduid de punten van de standen die men had gevraagd alsook een drukverloop.



Figuur 3-7: screenshot hoofdvenster trajectprogramma a.d.h.v. discrete posities

Dit programma houdt eveneens rekening met de beperkte drukken die toelaatbaar zijn in de spieren.

Gegevens omtrent de robot: lengtes, massa's, positie massamiddelpunt; alsook maximum hoeken van de gewrichten zijn eenvoudig aanpasbaar en worden automatisch opgeslagen in een M-file.



Figuur 3-8: screenshot venster verandering robotgegevens

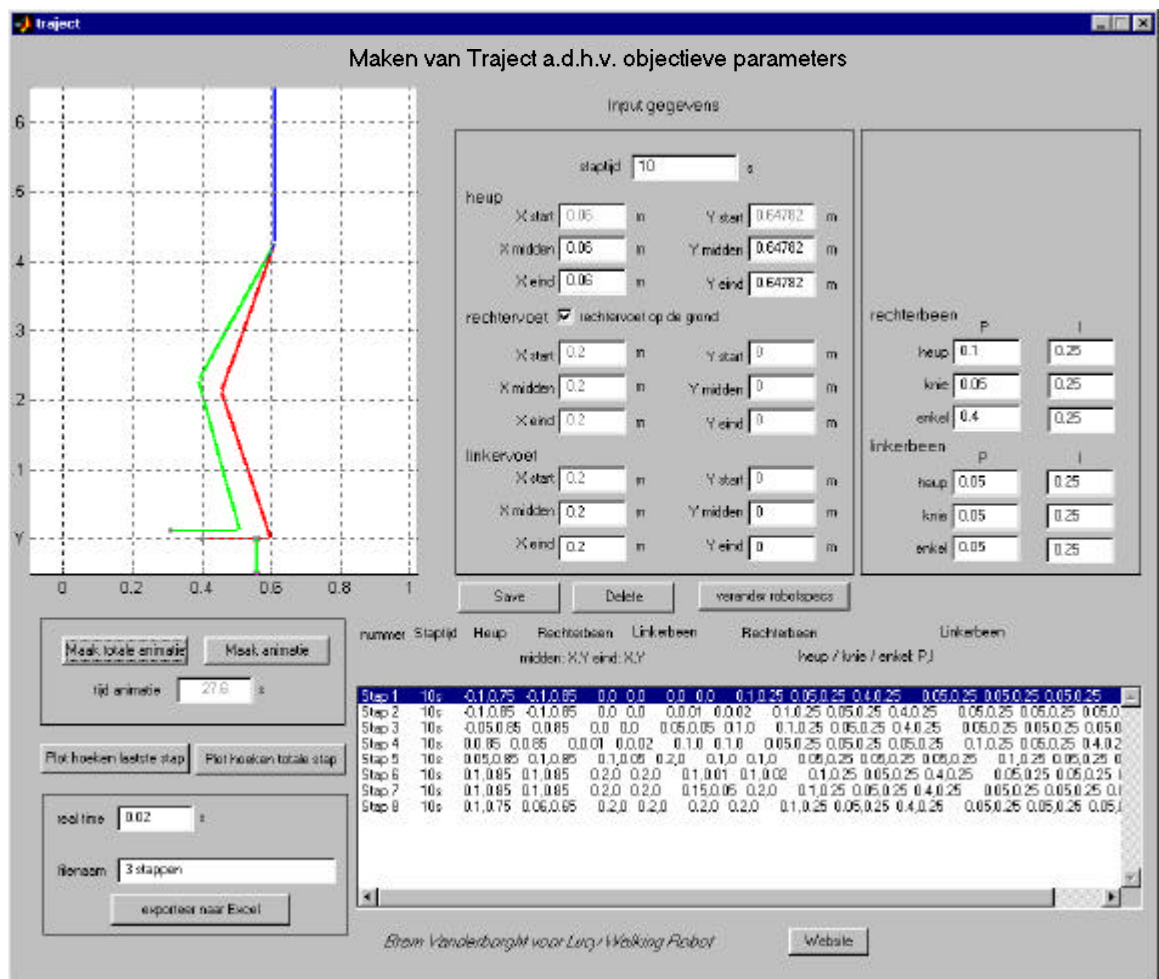
3.3.3.2 Trajectprogramma a.d.h.v. objectieve parameters

Een tweede strategie om het traject te bepalen is het ingeven van de objectieve parameters van de stap. Dit zijn de begin-, midden- en eindposities van de heup, linker- en rechterenkel en de eis dat in het begin en op het einde van de stap de snelheid 0 is. De computer berekent vervolgens de polynomen die de trajecten voorstellen.

Deze worden dan omgezet naar trajecten voor de verschillende scharnieren.

Indien deze stap gewenst is kan men deze data opslagen en worden de eindposities van deze stap automatisch overgezet naar de startposities van de volgende stap. Deze objectieve parameters worden eveneens in een lijst gezet zodanig dat de gebruiker een goed overzicht blijft behouden van de verschillende stappen. Men kan ook een animatie oproepen van gans de beweging.

Het programma houdt rekening met het beperkte bereik van de hoeken en er bestaat de mogelijkheid het verloop te visualiseren in grafiek.



Figuur 3-9: screenshot hoofdvenster trajectprogramma a.d.h.v. objectieve parameters

Bij het uitvoeren van testen met de robot bleek dat het nodig is om de P en I actie te veranderen naargelang het been ondersteunend werkt of niet. Omdat dan de belasting totaal verschillend is. Om hieraan te verhelpen wordt voor elke deelbeweging eveneens deze twee parameters meegegeven in het Excel-bestand. Bij de overgang van één waarde naar de volgende wordt dit niet bruusk opgelegd maar berekent de computer een lineaire overgang in 0,5 s.

Door het drukken op de knop “Exporteer naar Excel” wordt dan een Excel-bestand gemaakt met de opeenvolgende hoeken en eveneens de I-waarden en k1waarden (zie lokale controle). In dit Excel-bestand worden eveneens de lijst gezet van objectieve parameters. Deze lijst wordt echter niet gezien door het communicatie-programma, maar is zuiver informatief om later nog te weten welke beweging het bestand voorstelt.

3.4 Communicatieprogramma

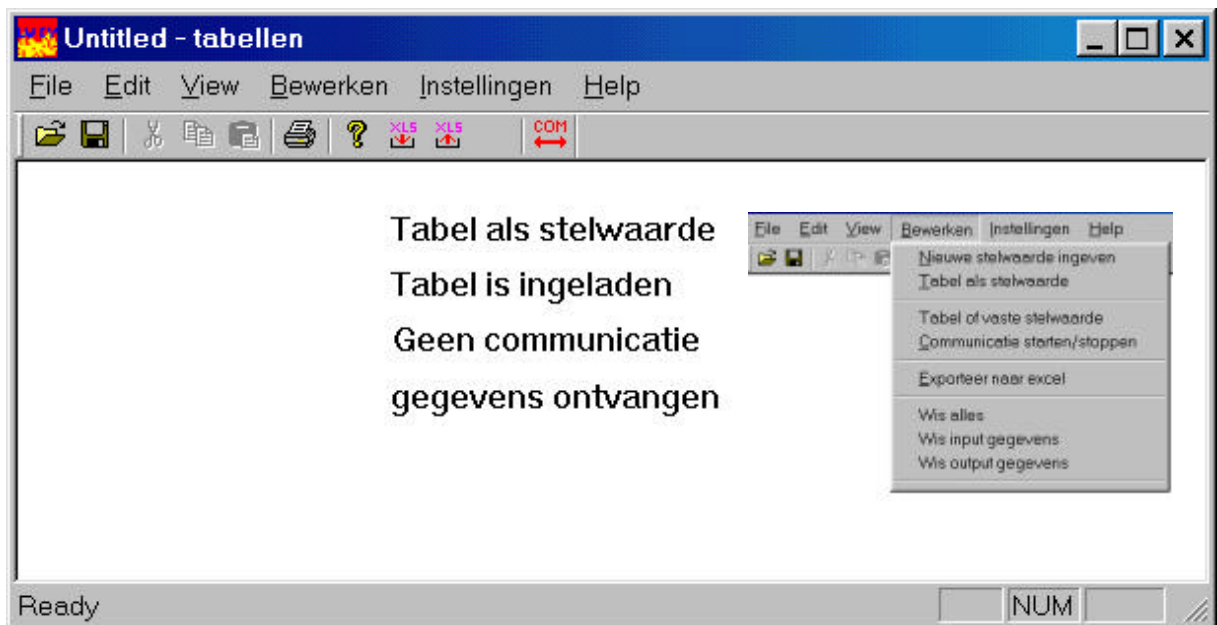
Het hoofddoel van dit programma is het opgelegde traject naar de microcontrollers te zenden en de meetresultaten te registreren.

Ook zijn er enkele ondersteunende routines voorzien om de robot te initialiseren en te testen.

Dit programma is geprogrammeerd in Microsoft Visual C++ 6.0.

Als traject kan ofwel een Excel-sheet worden opgeroepen ofwel kunnen vaste hoeken worden ingegeven.

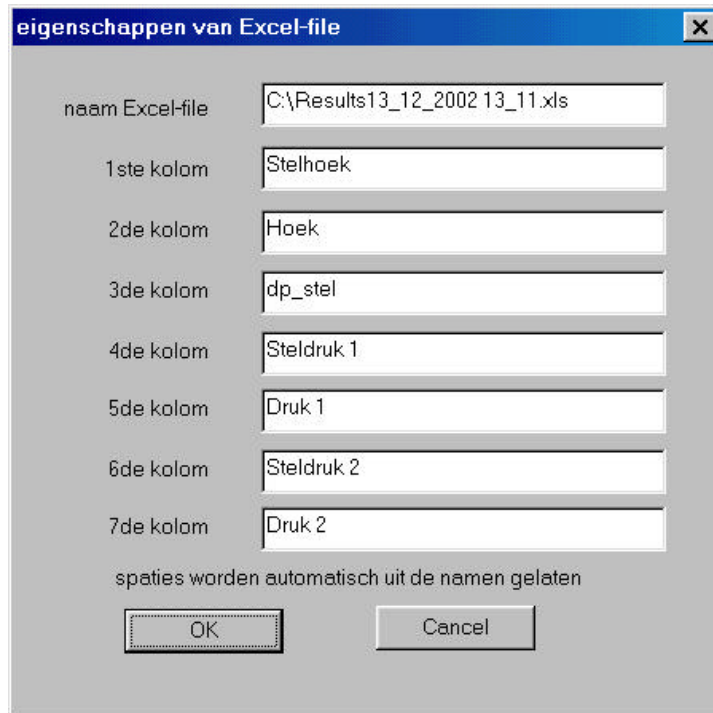
Door vervolgens op “communicatie starten/stoppen” te drukken worden de gegevens doorgezonden naar de microcontrollers en de meetresultaten geregistreerd.



Figuur 3-10: screenshot hoofdvenster communicatieprogramma

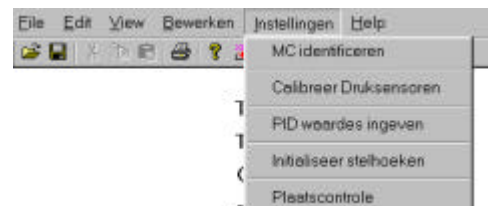
Deze kunnen dan worden gesaved in een .xls-file. Als voorstelling van naam wordt er gegeven: “results” gevolgd door de datum en uur van de proef. Deze kan de gebruiker echter wijzigen in onderstaand venster. Elk gewricht krijgt zijn eigen werkblad.

Dit laat toe om snel en eenvoudig verschillende experimenten uit te voeren.



Figuur 3-11: venster voor export naar Excel file

Verder zijn er nog enkele extra functies voorzien.



3.4.1 Identificeren van de microcontrollers.

Tijdens het ontwerp van de robot is er veel aandacht besteed aan de modulariteit. Dit wordt verder gezet in het gebruik van de microcontrollers, daar elke microcontroller zowel de functie kan aannemen van master als van de jointcontroller.

Dit wordt gedaan door een code te zenden naar de microcontrollers zodanig dat de microcontroller weet welke functie hij op zich moet nemen.

3.4.2 Calibreren van de druksensoren.

Door het handmatig maken van de elektronica rond de druksensoren verschillen ze onderling een beetje.

Daarom worden ze in het begin allen aan elkaar gelijk gesteld.

Dit doet men door eerst de atmosferische druk te meten, vervolgens de spieren te vullen met een verlaagde druk van ongeveer 3 bar en hierbij een tweede meting te verrichten.

De werkelijke waarde van deze twee drukken moet men dan manueel ingeven in de computer en de computer gaat dan naar de microcontrollers de waarden doorgeven waarmee ze de uitgelezen drukken moeten compenseren.

Indien men een extra ijkingsdruksensor zou hebben dan pas kan men spreken van een echte calibratie. Dit is in dit stadium nog niet echt nodig. Belangrijk is dat bij een bepaalde druk alle druksensoren dezelfde uitlezing geven.

In de toekomst zal het echter wel noodzakelijk zijn om een nauwkeurige druksensor aan te kopen, vooral voor de repeteerbaarheid van de proeven.

Unit	Value
atmosferische luchtdruk (bar)	1
atmosferische luchtdruk (atm)	1000
atmosferische luchtdruk (hPa)	750.06375
atmosferische luchtdruk (mm HG)	14.503824
atmosferische luchtdruk (PSI)	
Druk in leiding (bar)	3
Druk in leiding (atm)	3000
Druk in leiding (hPa)	2250.1912
Druk in leiding (mm HG)	43.511474
Druk in leiding (PSI)	

Figuur 3-12: programma laat toe om drukken in te geven in verschillende eenheden

3.4.3 Het instellen van de R,P,I,D-waardes.

Omdat het instellen van de R,P,I,D-waardes gebeurde met trial-and-error werden telkens ook deze waardes naar de microcontroller doorgezonden. Op deze wijze is het mogelijk om tijdens de beweging de regelaar aan te passen.

	rechterbeen			linkerbeen		
	heup	knie	enkel	heup	knie	enkel
R	3	3	3	3	3	3
P	0.4	0.05	0.4	0.25	0.25	0.25
I	0.25	0.25	0.25	0.25	0.25	0.25
D	0	0	0	0	0	0

Figuur 3-13: ingeven van de waardes voor regelaar

3.4.4 Plaats in programma

Het kan gebeuren dat er een microcontroller vastloopt. Om sneller te achterhalen waar de fout zich heeft voorgedaan zenden de microcontrollers op bepaalde plaatsen in het programma een code naar de dual ported RAM.

Deze codes kunnen door de computer worden uitgelezen en alzo kan men achterhalen in welke microcontroller het is foutgelopen en waar in het programma de fout zich heeft voorgedaan.

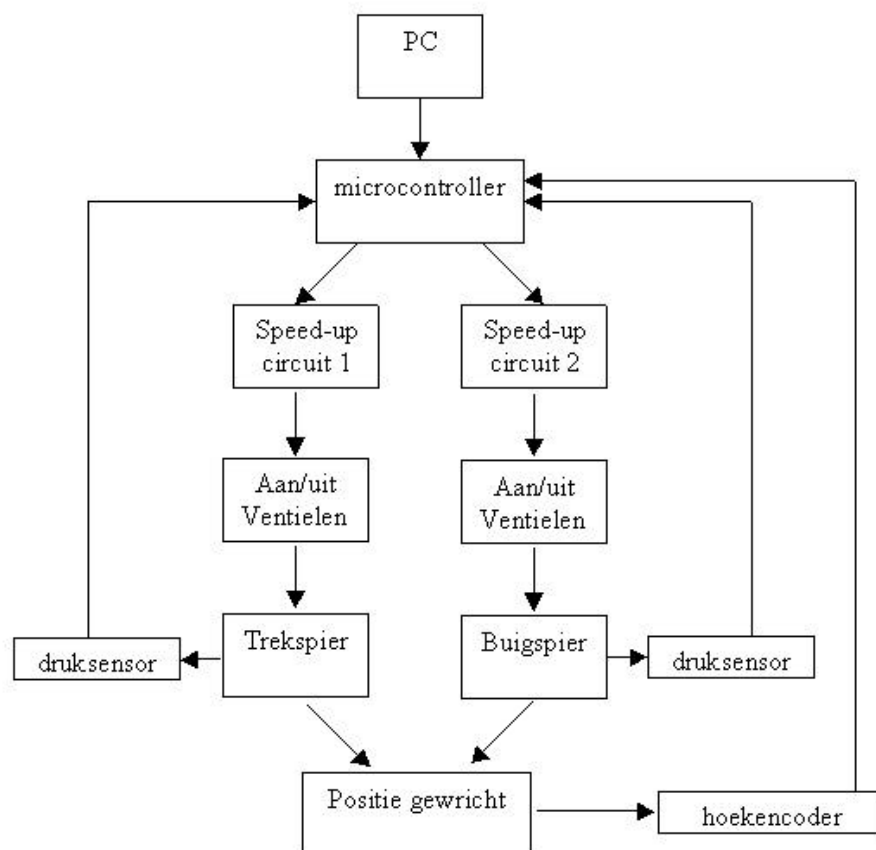
3.5 Lokale controle

De lokale controle wordt waargenomen door de microcontrollers, waarvan er één is voor elk gewricht. De microcontrollers bezitten een centrale plaats zoals weergegeven op onderstaand schema (3-14). De inputgegevens zijn: de druk in de strek- en buigspier, de hoek van het gewricht en via de RAMEenheid de gewenste hoek. Uitgaande van deze gegevens berekent de regelaar de stelwaarde voor de drukken. Deze drukken worden dan aangeboden aan de drukregelaar die via de speed-up circuits de ventielen gaat sturen.

De regelaar die de stelwaarde voor de drukken berekent is een PID regelaar.

De regelaar die de drukken gaat instellen is een bang-bang regelaar.

Beide regelaars worden eerst besproken, vervolgens hun implementatie in de microcontroller.



Figuur 3-14: schema van de opstelling

3.5.1 PID regelaar

De positie van de gewrichten en de stijfheid wordt bepaald door de druk in de spieren. De regelaar zal deze grootheden (P_1^{stel} en P_2^{stel}) moeten berekenen.

Men kan grosso modo stellen dat het verschil van de spierdrukken de positie van de arm bepaalt en dat de som van deze drukken de stijfheid van het systeem bepaalt. Rekening houdend met dit gegeven kunnen de ingangsvariabelen van het systeem als volgt gedefinieerd worden:

$$P_1^{stel} = P_m + \Delta P$$

$$P_2^{stel} = P_m - \Delta P$$

P_m is de gemiddelde druk $\left(P_m = \frac{P_1^{stel} + P_2^{stel}}{2} \right)$ en bepaalt de stijfheid van het systeem en ΔP bepaalt de positie van de arm. Werkt men bij een constante gemiddelde druk dan herleidt de taak van de regelaar zich tot de berekening van ΔP .

Als gemiddelde druk werd 2,5 bar genomen. Dit is het gemiddelde van de laagst mogelijke druk (1 bar) en de maximum toelaatbare druk (4 bar).

De regeling van ΔP gebeurt door middel van een PID-regelaar waarvan de I-actie variabel is.

De I-actie verkleint indien de drukregeling de gevraagde druk niet kan instellen. Indien dus het verschil tussen de gevraagde druk en de gewenste druk groter wordt zal de I-actie kleiner worden. Het is immers weinig zinvol iets te vragen dat de drukregeling toch niet kan verwezenlijken.

$$K_I = k \cdot I$$

$$k = \max \left[0, 1 ; 1 - R \cdot \max \left(\left| P_1 - P_1^{stel} \right| ; \left| P_2 - P_2^{stel} \right| \right) \right]$$

De I-actie kan alsoo verminderd worden met een factor 10. Dit wordt adaptieve I waarde genoemd. Is immers het verschil tussen P en P^{stel} te groot dan zakt k van 1 tot minimaal 0,1.

Dit geeft dus voor k een getal tussen 0,1 en 1.

Indien $e_k = a_{\text{stel}} - a_{\text{werkelijk}}$ en p de druk is, dan kan men de formule van de PID regelaar omvormen.

$$\Delta p_k = K_p e_k + \frac{K_D}{T_S} \Delta e_k + K_I T_S \sum_{i=0}^k e_i$$

$$\Delta p_{k-1} = K_p e_{k-1} + \frac{K_D}{T_S} \Delta e_{k-1} + K_I T_S \sum_{i=0}^{k-1} e_i$$

$$\Delta \Delta p_k = \Delta p_k - \Delta p_{k-1}$$

$$\Delta \Delta p_k = K_p (e_k - e_{k-1}) + \frac{K_D}{T_S} (e_k - 2e_{k-1} + e_{k-2}) + K_I T_S e_k$$

$$\Delta \Delta p_k = (K_p + \frac{K_D}{T_S} + K_I T_S) e_k - (K_p + \frac{2K_D}{T_S}) e_{k-1} + (\frac{K_D}{T_S}) e_{k-2}$$

$$\Delta \Delta p_k = (\text{kwaarde1} + K_I T_S) e_k - (\text{kwaarde2}) e_{k-1} + (\text{kwaarde3}) e_{k-2}$$

$$\Delta p_k = \Delta \Delta p_k + \Delta p_{k-1}$$

$$\text{kwaarde1} = K_p + \frac{K_D}{T_S}$$

$$\text{kwaarde2} = K_p + \frac{2K_D}{T_S}$$

$$\text{kwaarde3} = \frac{K_D}{T_S}$$

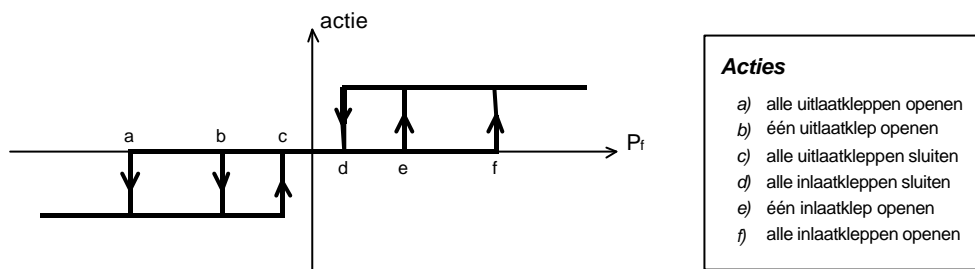
De gebruiker moet de waarden: R , I , K_p en K_d ingeven. De computer zal dan de waarden die de regelaar nodig heeft (R , I , kwaarde1 , kwaarde2 en kwaarde3) hieruit berekenen en doorsturen naar de microcontroller. Door dit zo te doen worden de microcontrollers ontlast omdat ze enkele berekeningen minder moeten doen.

Bij de implementatie van bovenstaande vergelijkingen in assembler dient men rekening te houden met over- en underflows, bit-precisie en de volgorde waarin men de bewerkingen uitvoert.

3.5.2 Bang-bang regelaar

De stelwaarden van de druk worden dan doorgegeven naar de bang-bang controller die geoptimaliseerd is geweest in [16]. Deze resultaten werden dan ook integraal overgenomen.

Naargelang het drukverschil tussen de gevraagde en de gewenste waarde (=foutsignaal P_f) worden een verschillend aantal inlaatkleppen of uitlaatkleppen geopend.



Figuur 3-15: actie van de regelaar in functie van het foutsignaal P_f

Als waarden werd er gekozen:

	$P \leq 2,5$ bar	$P > 2,5$ bar
a	-0,120 bar	-0,120 bar
b	-0,025 bar	-0,030 bar
c	-0,02 bar	-0,025 bar
d	0,02 bar	0,025 bar
e	0,025 bar	0,030 bar
f	0,120 bar	0,120 bar

Tabel 3-1: instelling drukken bang-bang regelaar

3.5.3 Implementatie in de microcontroller

Vooreerst is begonnen met het schrijven van een bibliotheek essentiële routines om bijvoorbeeld drukken en hoeken uit te lezen, gegevens te verzenden, kleppen te sturen, initialisatieprocedures uit te voeren, ... Deze code is geprogrammeerd in de EEPROM en moet na herinschakelen van de stroom niet meer worden geherprogrammeerd.

Er werden verschillende testprogramma's ontwikkelt om de verschillende hardware-componenten afzonderlijk te kunnen testen.

De belangrijkste testprogramma's zijn:

- Testqspi.asm Testen van de druksensoren
Geeft de uitlezing van de druksensor continu op het scherm weer.
- Testtpu.asm Testen van de kleppen
Stuurt één voor één alle kleppen.
- Testenco.asm Testen van de hoekencoder
Geeft na detectie van de derde puls, die de teller initialiseert, continu de hoek op het scherm weer.
- Testcomm.asm Testen van de communicatie via ISAbus-kaart
Programma voor hoger beschreven communicatietest.
- TestserA.asm Testen seriële A-poort
Herschrijft op het scherm wat gebruiker heeft doorgezonden naar microcontroller via seriële A-poort.
- TestserB.asm Testen seriële B-poort
Herschrijft op het scherm wat gebruiker heeft doorgezonden naar microcontroller via seriële B-poort.

3.5.4 Code voor master

De code voor de master is zeer eenvoudig. Na de initialisatie om de druksensoren te calibreren en het nulpunt van de hoekencoder te detecteren wordt er volgende sequentie gestart. Set-Write-Bit hoog (5 ms), vervolgens terug laag (5 ms), Set-Read-Bit hoog (5 ms) en vervolgens terug laag (5 ms).

De master zelf zendt nog geen data. In de toekomst zullen dit de absolute hoek, X,Y-posities van bovenlichaam en de reactiekrachten van de grond op de voet zijn.

3.6 Code voor de joints

Na de initialisatie wordt er om de 0,488 ms een stuk code uitgevoerd die de stand van de kleppen zal definiëren. De 0,488 ms is dus de samplingtijd.

De reductie van deze samplingtijd van 2,197 ms ([16]) naar 0,488 ms gaf een significante stabiliteitverbetering. De maximale duur van de berekeningen per cyclus bedraagt 0,160 ms. Dus in feite kan men de samplingfrequentie nog opvoeren.

De kortste openingstijd van een klepje bedraagt echter 0,7 ms (voor een klepje zonder veer), dus sneller bemonsteren heeft weinig zin.

Gedurende een cyclus worden volgende stappen doorlopen:

De microcontroller weet welke de stelwaarde van de hoek is.

Eerst zal hij de druksensoren en hoekencoder opmeten, hieruit worden via een PID-regelaar de stelwaarden van de drukken bepaald. Indien nodig worden ze gelimiteerd op 1,1 bar als onderste grens en 4 bar als bovenste grens. De onderste grens zorgt voor een minimumdruk nodig om nog de druk te kunnen regelen, vermits de atmosferische druk 1 bar bedraagt. De bovenste grens is 0,5 bar lager dan de veiligheidsdruk waarbij de leidingen ontluicht worden. De bang-bang controller zal via het openen en sluiten van de kleppen deze druk realiseren.

Op het einde vindt de communicatie met de computer plaats.

De microcontroller gaat eerst nakijken of de OK-Write-Bit of OK-Read-Bit hoog staan. Indien de OK-Read-Bit hoog staat zal hij de data van de computer ontvangen. Deze zijn stelwaarde hoek en de data die hij nodig heeft voor de PID-regelaar.

Indien de OK-Write-Bit hoog staat zal hij zijn meetresultaten naar de computer zenden, deze zijn: stelwaarde hoek (controle of hoek goed ontvangen is), werkelijk hoek, berekend drukverschil van regelaar, stelwaarde druk 1 en 2, werkelijk druk 1 en 2.

Hoofdstuk 4

Resultaten

Het doel van deze thesis was om een hardware-matige werkende robot te maken en bestond er niet in om een optimale regelaar te ontwerpen. Als eerste fase werd een PID regelaar geprogrammeerd als low-level controller in de microcontrollers. Deze had dan ook vooral de bedoeling de goede werking van de hardware na te gaan.

Ten tweede heeft men bij geavanceerdere technieken als feedforward en sliding mode controle een goede modelisatie nodig (massa's, traagheidsmomenten,...) dewelke tijdens de bouw nog te snel veranderde.

De PID controller werd geprogrammeerd in assembler-code. Dit heeft ten opzicht van een programmatie in een hogere taal als C het voordeel van veel sneller te zijn, maar is veel moeilijker wegens de 16-bit precisie. Geavanceerdere mathematische bewerkingen zullen moeilijker te programmeren zijn in assembler-code, waardoor er zal moeten overgegaan worden op hogere programmeertalen die een langere rekentijd vergen. Dit kan tot gevolg hebben dat de sampling-frequentie verlaagd dient te worden, hetgeen minder goed is. Alhoewel een deel van de berekeningen door de PC op zich genomen kan worden.

Gedurende de experimenten is de PID regelaar op verschillende momenten verbeterd om bepaalde problemen op te lossen. In de volgende alinea wordt een overzicht gegeven van de verschillende problemen met hun oplossingen.

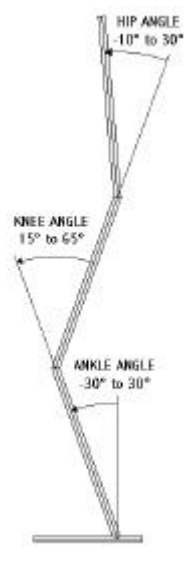
4.1 Registratie van de meetresultaten

De meetresultaten staan in een Excel-bestand, waar elk gewricht zijn eigen werkblad heeft.

De meetgegevens zijn de stelwaarde hoek, werkelijke hoek, berekend drukverschil van regelaar, stelwaarde druk 1 en 2 en werkelijke druk 1 en 2.

Per seconde worden 2450 16-bits gegevens ontvangen. Dit resulteert in grote Excel-bestanden. Zo levert een bestand dat 60 seconden beweging weergeeft een Excel-file van 1,35 Mb op.

Onderstaande figuur (4-1) legt de definitie van de gebruikte hoeken vast.



Figuur 4-1: definitie gebruikte hoeken

4.2 Verhogen van de samplingfrequentie

Oorspronkelijk werd er gewerkt met een samplingfrequentie van 2,197 ms [16]. Dit was enkel geschikt indien het been niet belast werd en vrij in de lucht hing.

Door de samplingtijd te reduceren naar 0,488 ms gaf dit een significante stabiliteitverbetering waardoor ook het been belast kon worden.

4.3 Afwezigheid grondcontactsensoren

Momenteel zijn er nog geen sensoren op de voet aangebracht die detecteren of de voet al dan niet op de grond staat. Vooral in closed loop leidt dit tot problemen. De robot stelt momenteel de drukken in uitsluitend op basis van de gevraagde stelhoek, terwijl een eerste bezorgdheid moet zijn of het ondersteunende been stabiel op de grond staat.

Bij quasi-statische controle kan men zo detecteren of het massamiddelpunt binnen de voet valt van het ondersteunende been, terwijl bij dynamisch bewegen of het zero moment point (ZMP) erbinnen is gelegen.

Bij de experimenten is gebleken dat wanneer de voet, door een externe invloed, foutief op de grond komt dan toch probeert de gevraagde hoekposities te bereiken. Dit doet hij door de koppels op te voeren. Wanneer de voet dan loskomt door overwinning van de wrijvingskracht dan heeft dat zeer grote oscillaties tot gevolg.

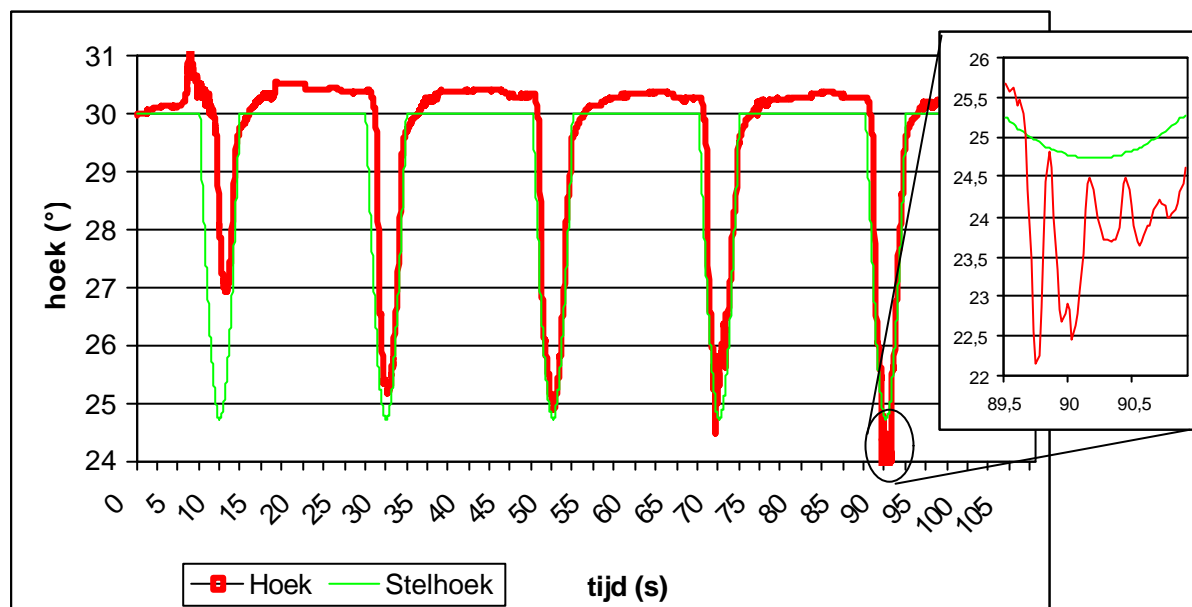
4.4 Variërende parameters PID regelaar

De acties van de PID-regelaar werden manueel ingesteld. Onderstaande voorbeelden tonen dat voor de verschillende deelbewegingen de parameters anders gekozen moeten worden. Alhoewel de programma's van de robot zo zijn gemaakt dat deze acties eenvoudig veranderd kunnen worden, blijft het toch een tijdrovende methode.

4.4.1 Invloed I-actie op heup

Voor de drie gewrichten is een integrerende actie steeds nodig omdat de stelwaarde een permanente verandering ondergaat.

Onderstaande grafiek (4-1) geeft de heuphoek weer bij de cyclus rechtkomen terug zakken, waarbij na elke cyclus de I-actie stijgt (0,25-0,5-1-2-5) en de P-actie constant blijft (0,5).



Grafiek 4-1: heuphoek bij stijgende I-actie (0,25-0,5-1-2-5)

Bij de eerste cyclus is de I-actie klein wat resulteert in een traag systeem. Wordt de I-actie verhoogd dan ziet men dat de stelwaarde beter wordt gevolgd, tot het systeem begint te oscilleren.

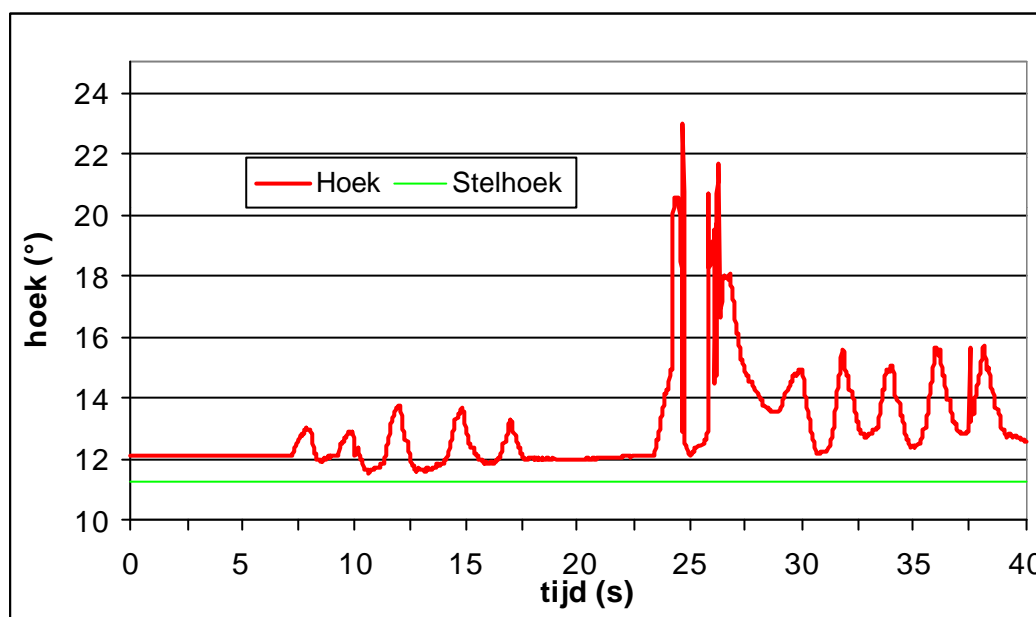
Uit bovenstaande grafiek blijkt dat $I=1$ de beste oplossing is. Deze is echter te instabiel voor externe perturbaties indien het andere been niet op de grond steunt. Voor een aanvaardbare stabiliteit is men verplicht $I=0,25$ te nemen.

4.4.2 Invloed P-actie op enkel

Het variëren van vooral de P-actie naargelang het belastingsgeval bleek noodzakelijk. Een lage P-actie is nodig wanneer de belasting laag is, bijvoorbeeld in de enkel wanneer de voet in de lucht hangt.

Een grote P-actie is vereist wanneer een hoge stijfheid vereist is, bijvoorbeeld wanneer de voet op de grond staat.

Op onderstaande grafiek (4-2) is de enkelhoek weergegeven van een op de grond staande voet met externe oscillaties. Gedurende de eerste 20 seconden was er een grote P-actie (0,4) en een lagere P-actie (0,05).

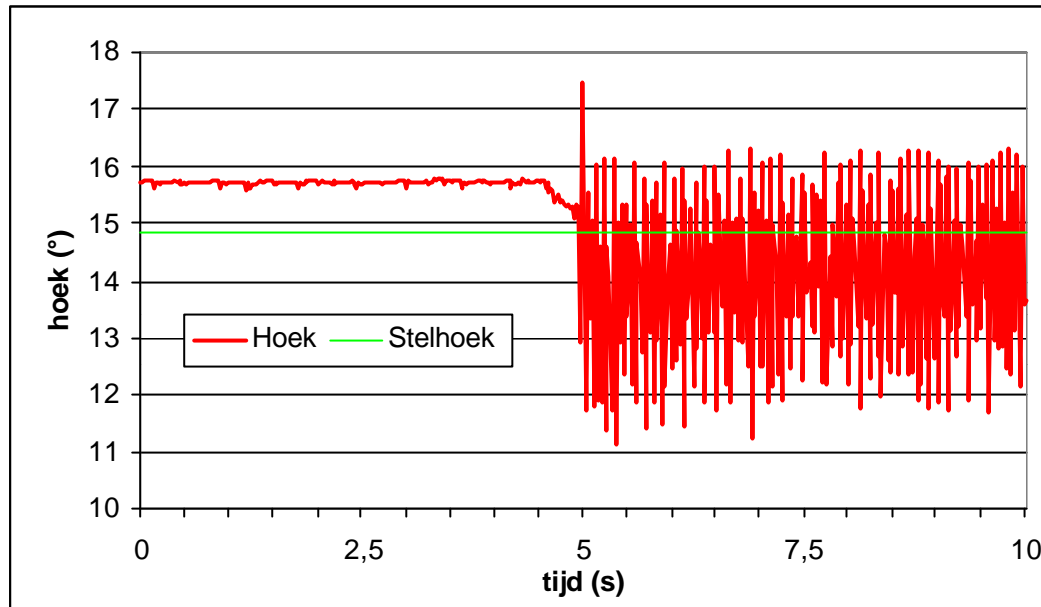


Grafiek 4-2: enkelhoek van een op de grond staande voet met grote P-actie (0-20 s) en kleine P-actie (20-40 s)

Men ziet dat door de verhoogde stijfheid de oscillaties kleiner zijn.

Dus hiervoor is een hoge P-actie vereist.

Indien de voet in de lucht hangt is de belasting van het enkelgewricht zeer klein want bestaat enkel uit het gewicht van de voet. Een hoge P-waarde (0,4 i.p.v. 0,05) zoals op onderstaande grafiek (4-3) tussen 5 en 10 seconden leidt tot sterke oscillaties. Een video-opname hiervan staat op de bijlage CD.

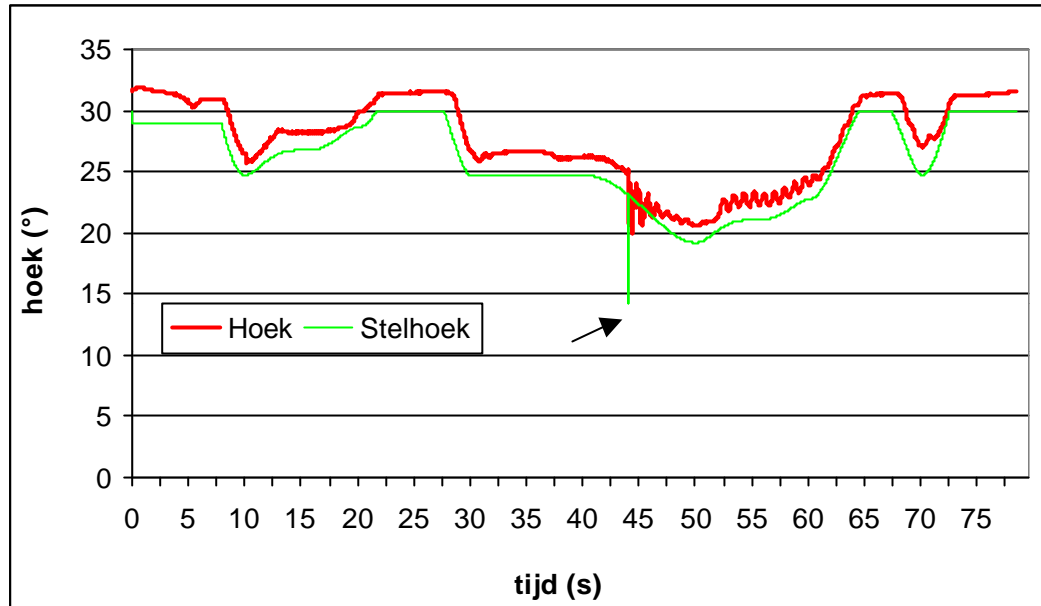


Grafiek 4-3: enkelhoek van niet op de grond steunende voet met kleine P-actie (tussen 0-5 s) en grote P-actie (tussen 5-10 s)

Het veranderen van de parameters van de regelaar mag niet bruusk gebeuren omdat dit leidt tot onaanvaardbare oscillaties.

4.5 Communicatiefouten

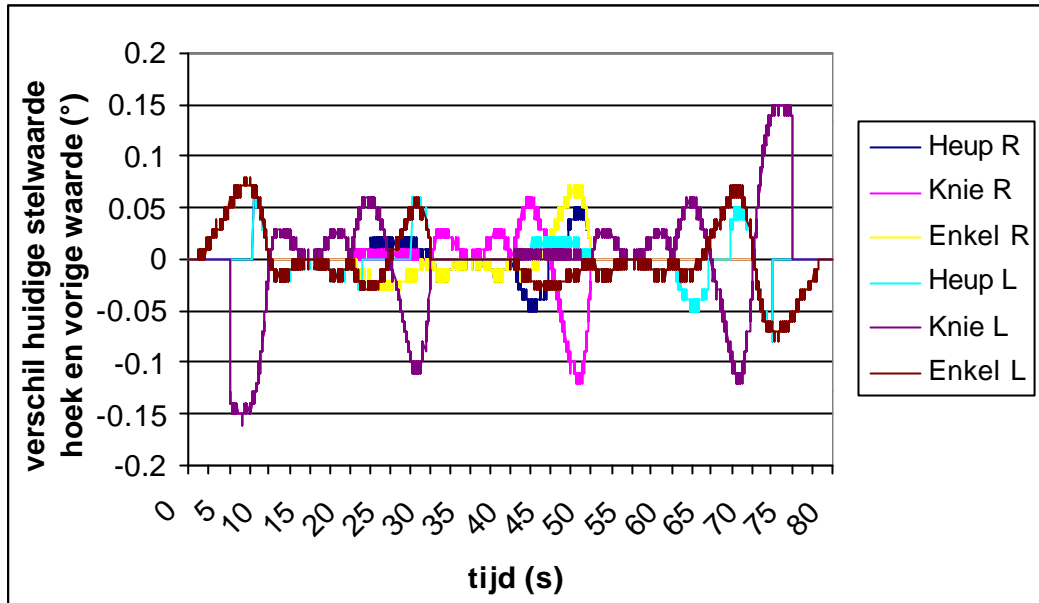
Op grafiek 4-4, die de heuphoek voorstelt bij een stapbeweging, heeft zich een communicatiefout voorgedaan op 44 s. De ontvangen stelwaarde voor de heuphoek bedroeg $14,23^\circ$ i.p.v. de correcte $23,15^\circ$. Een dergelijke communicatiefout doet het systeem sterk oscilleren zoals de rode grafiek weergeeft na de fout.



Grafiek 4-4: communicatiefout leidt tot oscillaties

Om dergelijke ongewenste communicatiefouten te vermijden vergelijkt de microcontroller de ontvangen stelwaarde voor de hoek met de vorige waarde. Indien dit verschil groter is dan $0,4^\circ$ wordt deze nieuwe stelwaarde genegeerd.

De waarde $0,4^\circ$ is gekozen als volgt. Grafiek 4-5 toont het verschil tussen 2 opeenvolgende stelwaarden voor een stapbeweging. Dit verschil gaat tot maximum $0,15^\circ$. Bij andere bewegingen ligt dit in dezelfde range. De limiet moet echter minstens het dubbele zijn van $0,15^\circ$. Omdat als één waarde wordt genegeerd het verschil $0,3^\circ$ kan zijn. Indien bijvoorbeeld de drie opeenvolgende waarden zijn: $14,00^\circ$ - $14,15^\circ$ - $14,30^\circ$, dus telkens $0,15^\circ$ als verschil, en de limiet $0,2^\circ$. Indien de tweede ontvangen waarde door een communicatiefout $23,15^\circ$ is, dan wordt deze terecht genegeerd. Dit betekent dat de eerste waarde, zijnde $14,00^\circ$, wordt behouden. Bij de ontvangst van de derde waarde wordt de correcte waarde $14,30^\circ$ ontvangen. De microcontroller zal echter deze waarde onterecht verwerpen omdat het verschil $0,3^\circ$ bedraagt. Daarmee dat de limiet minstens moet liggen op tweemaal het grootste verschil. Wanneer de snelheid van de beweging verhoogd wordt zal de limiet aangepast moeten worden.



Grafiek 4-5: verschil tussen 2 opeenvolgende stelwaarden voor een stapbeweging

4.6 Drie stappen

Vermits de communicatiekabel beperkt is tot 1,5 meter is de bewegingsruimte van de robot beperkt. De lengte is echter voldoende om van rustpositie recht te komen, een halve stap te zetten, vervolgens een volledige stap te zetten en terug de voeten naast elkaar te plaatsen om uiteindelijk terug te keren naar rustpositie.

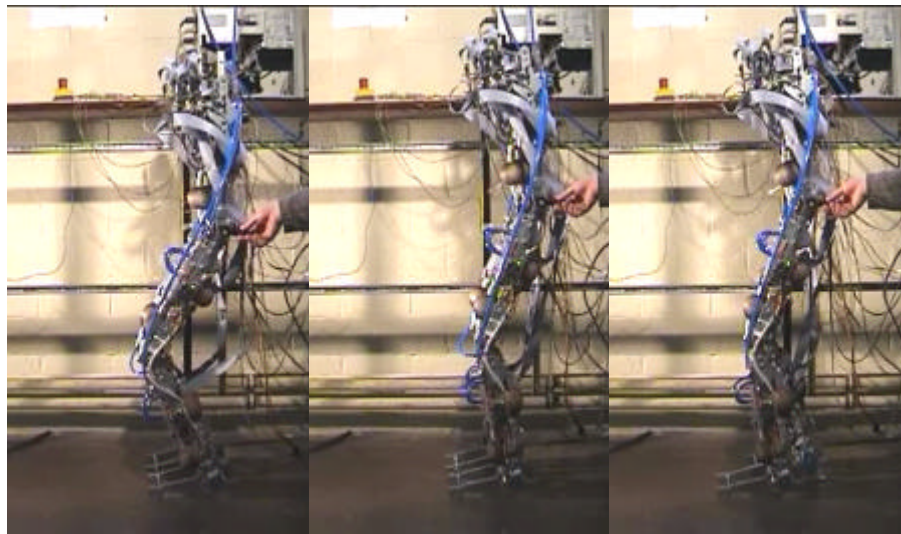
4.6.1 Traject via objectieve parameters

In onderstaande tabel staan de verschillende opeenvolgende posities van de heup, linker- en rechterenkel die werden ingegeven in het programma dat de trajecten genereert.

Tijd (s)	X heup (m)	Y heup (m)	X rechterenkel (m)	Y rechterenkel (m)	X linker enkel (m)	Y linker enkel (m)	Omschrijving
0	-0,14362	0,64782	0	0	0	0	rustpositie
5	-0,05	0,74891	0	0	0	0	rechtkomen
10	-0,05	0,85	0	0	0	0	Rechtstaand
15	-0,05	0,85	0	0	0	0,01	Opheffen linkervoet
20	-0,05	0,85	0	0	0	0,02	Opheffen linkervoet
25	0	0,85	0	0	0,05	0,05	Stap met staphoogte 0,05m
30	0,05	0,85	0	0	0,1	0	Neerkomen voet met staplengte 0,1m
35	0,05	0,85	0	0,01	0,1	0	Opheffen rechervoet
40	0,05	0,85	0	0,02	0,1	0	Opheffen rechervoet
45	0,1	0,85	0,1	0,05	0,1	0	Stap met staphoogte 0,05m
50	0,15	0,85	0,2	0	0,1	0	Neerkomen met staplengte 0,02m
55	0,15	0,85	0,2	0	0,1	0,01	Opheffen linkervoet
60	0,15	0,85	0,2	0	0,1	0,02	Opheffen linkervoet
65	0,15	0,85	0,2	0	0,15	0,05	Stap met staphoogte 0,05m
70	0,15	0,85	0,2	0	0,2	0	Neerkomen voet met staplengte 0,1m
75	0,15	0,75	0,2	0	0,2	0	Terugkeren naar rustpositie
80	0,06	0,64782	0,2	0	0,2	0	Rustpositie

Tabel 4-1: opeenvolgende posities 3 stappen-beweging
(vet betekent voet in beweging)

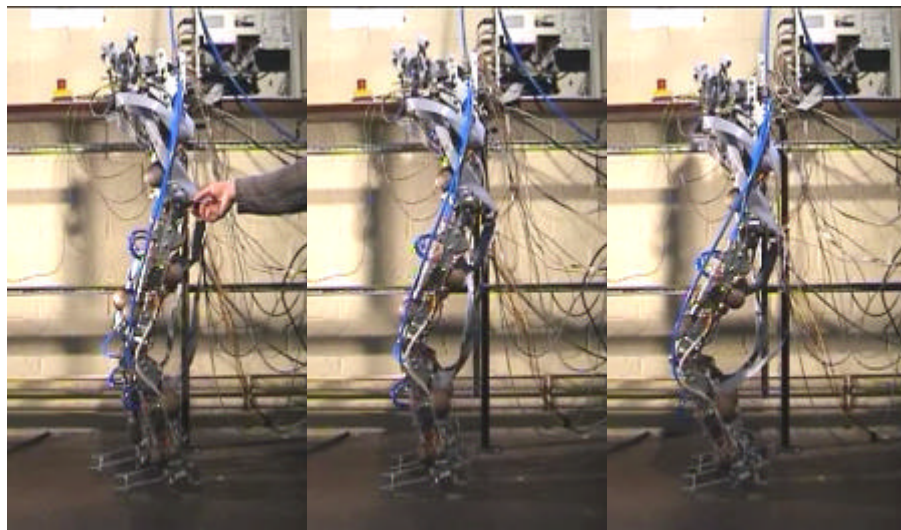
Op volgende foto's staan de verschillende fasen van de 3 stappen-beweging. Enkele video-opnames van deze beweging staan op de bijlage CD.



rustpositie (0 s)

rechtstaand (10 s)

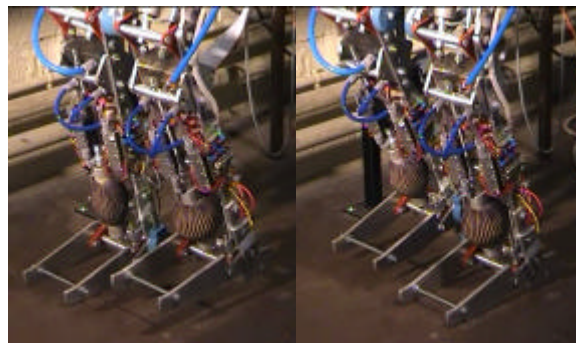
1^{ste} stap (30 s)



2^{de} stap (50 s)

3^{de} stap (70 s)

rustpositie (80 s)



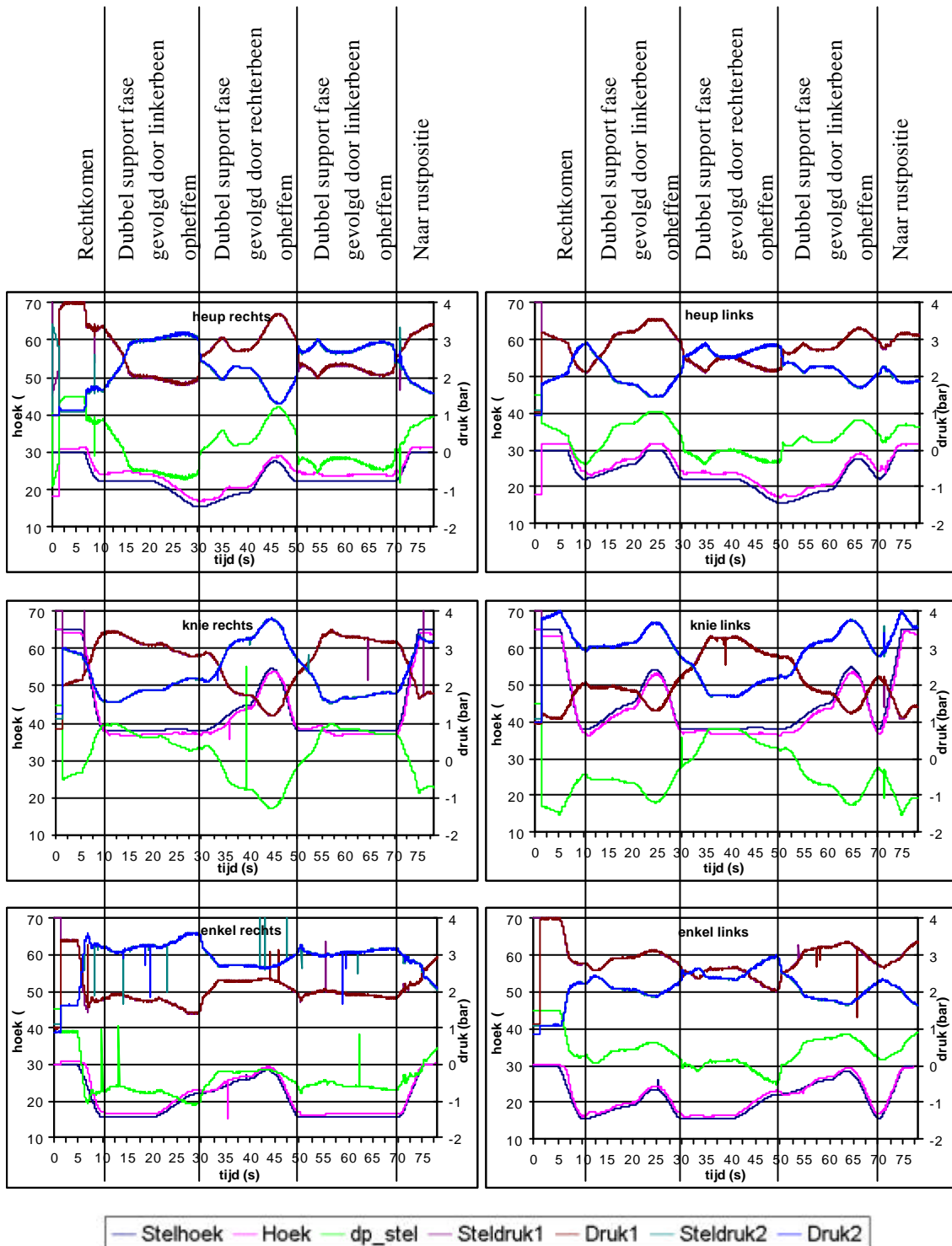
Opheffen voet 0,02 m

Staphoogte 0,05 m

Figuur 4-2: opeenvolgende posities 3 stappen beweging

4.6.2 Meetresultaten

Op onderstaande grafieken staan de resultaten van de verschillende gewrichten.



Grafiek 4-6: meetresultaten voor 3 stappen-beweging

Op de grafieken (4-6) van de meetresultaten merkt men soms zeer grote discrete pieken, vooral in de grafiek van “enkel rechts”. Dit zijn fouten in de communicatie van robot naar PC. Deze fouten hebben echter geen nefaste invloed op de robot en worden dan ook niet weggefilterd. De gebruiker blijft hierdoor een idee van de kwaliteit van de communicatie behouden.

4.6.3 Bespreking resultaten

Het meest kritieke deel van de beweging is de overgang van single support (vb linkervoet op grond) via double support (beide voeten op de grond) naar single support (rechtervoet op grond). Voor de 3 stappen-beweging is deze overgang na 30 s en 50 s. De drukken ondergaan dan een sterke verandering wegens de belastingswissel, zoals de grafieken aangeven. Staat het been op de grond dan moet de druk van de strekspier van de knie (druk 1) groter zijn dan deze van de buigspier (spier 2), hangt echter het been in de lucht dan moet de druk van de buigspier groter zijn dan van de strekspier. Deze overgang moet dan ook zeer geleidelijk gebeuren. Daarom wordt eerst de voet geleidelijk opgetild tot een hoogte van 0,02 m in 10 s, in de eerste seconden heeft men dan dubbel support gedurende dewelke de drukken sterk variëren omdat de belasting van het ene been naar het andere been moet overgaan. Gedurende de volgende 10 s wordt dan de stap verder afgewerkt.

De hoekfout blijft echter steeds binnen 1°, wat zeker aanvaardbaar is en verklaard wordt door de lage I-actie zoals aangetoond is in 2.4.1.

De grootste hoekfouten in de gewrichten treden op bij het van de grond komen van de voet.

Voor een goede trajectcontrole is een stijve verbinding nodig. Vroeger was dan ook “hoe stijver hoe beter” de regel [8]. Juist om redenen van dynamisch stappen en energierecuperatie werd van dit idee afgestapt. Dan mag ook de controle niet aan dit “hoe stijver hoe beter” principe blijven hangen door vast te houden aan zuivere trajectcontrole, maar moet onderzocht worden in hoeverre men van dit traject mag afwijken. Voor het stappen is bijvoorbeeld het zeer precies neerkomen van de voet op een bepaalde plaats helemaal niet belangrijk. Belangrijker is dat de hoeveelheid energie dat ervoor nodig is beperkt blijft en de robot stabiel is.

Jerry Prat (MIT) doet zelfs geen trajectcontrole meer, maar beschouwt 5 aandachtspunten zodat de Famingo Spring kan stappen [8]:

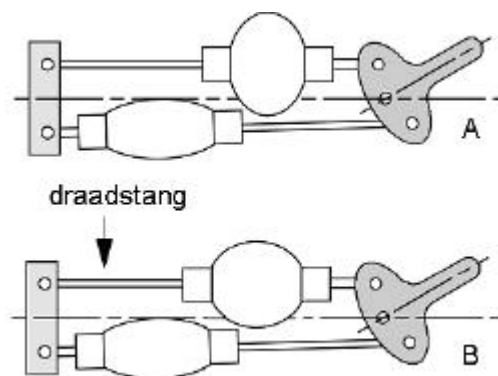
- Hoogte moet stabiel zijn
- Pitch moet stabiel zijn
- Snelheid moet stabiel zijn
- Het zwaaiende been moet zodanig bewegen dat de voeten in zulke posities worden geplaatst zodanig dat de stabiliteit van hoogte, snelheid en pitch gegarandeerd zijn
- Verwisseling van ondersteunend been moet gebeuren op gezette ogenblikken

Door handregeling van deze intuïtieve controleparameters bekwam men een stappende robot.

4.7 Invloed van koppelkarakteristiek

$M = p_2 m_2(\mathbf{a}) - p_1 m_1(\mathbf{a})$ bepaalt het koppel uitgeoefend door een gewricht. Indien men de koppelkarakteristieken $(m_1(\mathbf{a}), m_2(\mathbf{a}))$ foutief kiest dan kan het zijn dat het benodigde koppel niet opgewekt kan worden daar de drukken in de spieren begrensd zijn tussen 1,1 bar en 4 bar.

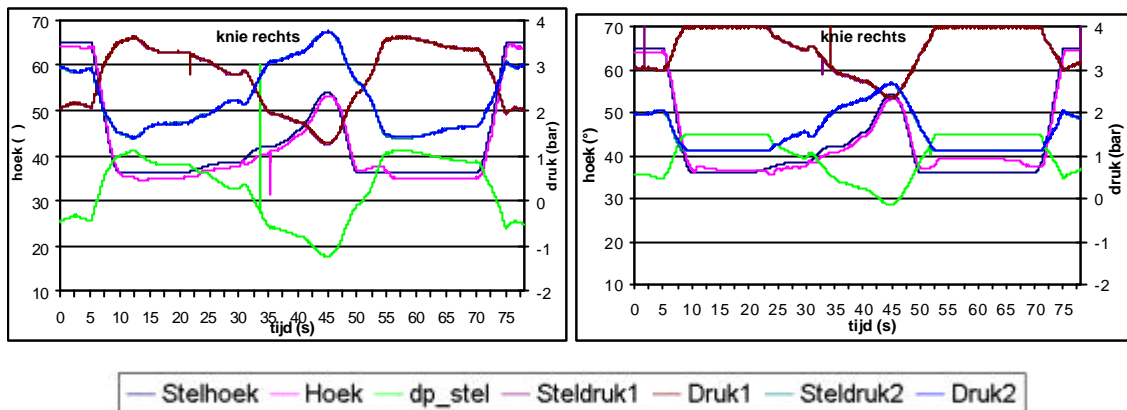
Men kan de koppelkarakteristieken onder andere veranderen door variatie van de lengte van de draadstangen zoals op onderstaande figuur (4-3). Dit betekent dat bij eenzelfde hoek de contractie anders is, dus ook de ontwikkelde kracht en koppel.



Figuur 4-3: verandering koppelkarakteristiek door aanpassen lengte draadstang

Bij de proef, waarvan de grafiek aan de linkerkant (4-7) is weergegeven, werd de draadstang die de spier met het hefboommechanisme verbindt ingesteld zoals berekend.

Vervolgens werd de draadstang foutief ingesteld, waarvan de resultaten in de rechtergrafiek staan.



Grafiek 4-7: meetresultaten kniehoek bij correct en foutief ingestelde draadstang

Bij de proef met de foutief ingestelde draadstang ziet men dat tussen 55 en 70 s de gevraagde hoek niet bereikt kan worden.

De gevraagde hoek is lager dan de werkelijke hoek, dit betekent dat de knie teveel is doorgebogen. Om de knie te strekken moet de strekspier (spier 1) meer druk krijgen en de buigspier (spier 2) minder druk. Spier 1 zit echter op zijn maximum (4 bar) en spier 2 op zijn minimum (1,1 bar). Door een correcte instelling van de draadstangen kan men dit probleem oplossen en bekomt men de linkergrafiek.

Het instellen van de lengte van de draadstang is dus essentieel. De berekende koppelkarakteristieken vormen hierbij een goede richtlijn om de draadstangen in te stellen.

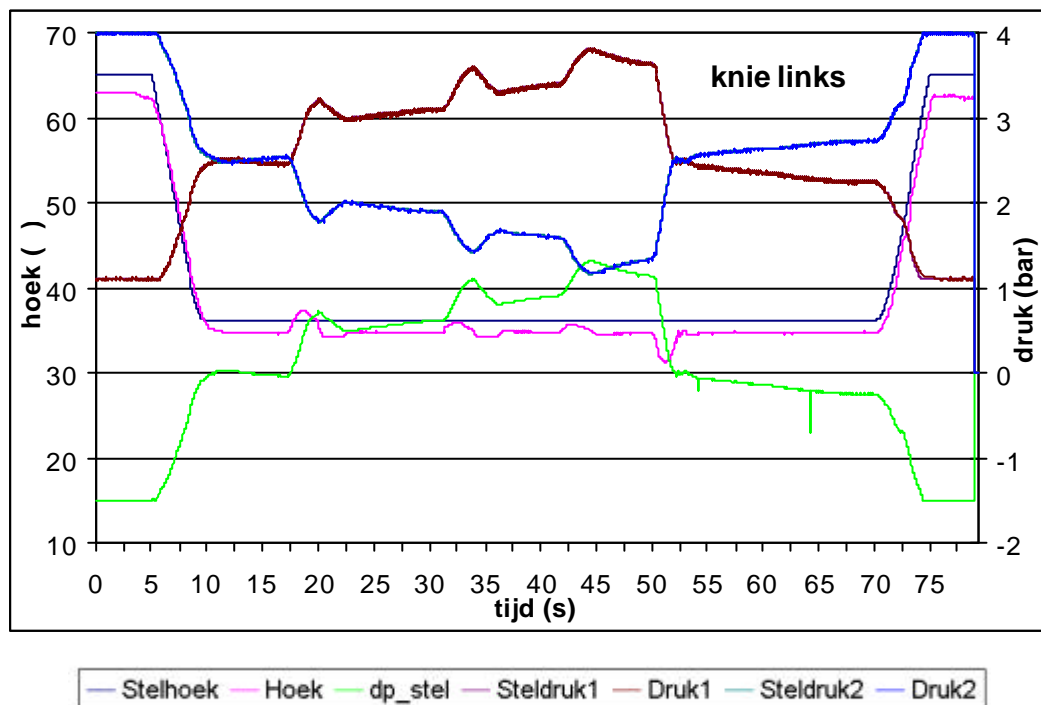
4.8 Aanbrengen van extra belasting

In deze proef kwam de robot recht vanuit rustpositie naar een heuphoogte van 0,85 m in 10 s, vervolgens bleef deze stand gedurende 60 s aangehouden om dan terug te zakken naar rustpositie. In de hoogste stand werd driemaal extra gewicht (10,01 kg - 5,22 kg - 4,035 kg) aan de heup gehangen en vervolgens de 3 gewichten tegelijkertijd er terug afgehaald.

Merk op dat dit een toename is van 70% van het eigengewicht van de robot.

Onderstaande grafiek (4-8) geeft de kniehoek weer. Deze stijgt steeds enkele graden door om zich vervolgens terug te herstellen. Het verschil in druk (dp_{stel}) wordt groter bij toenemend gewicht.

Dit kan men halen uit de formule $M = p_2 m_2(\mathbf{a}) - p_1 m_1(\mathbf{a})$ waar indien het vereiste moment groter wordt door de toenemende belasting het verschil tussen p_1 en p_2 groter moet worden.



Grafiek 4-8: meetgegevens knie bij toenemende belasting

Hoofdstuk 5

Conclusie en toekomstperspectieven

5.1 Conclusie

Deze thesis handelde over de bouw van een tweepotige robot geactueerd met pneumatische actuatoren. Deze artificiële spieren hebben een hoge kracht/gewichtsverhouding en een inherente natuurlijke soepelheid en bieden daarom interessante perspectieven als actuator in een stappende robot. De spieren ontwikkelen echter alleen een trekkracht en geen drukkracht. Er zijn dan ook twee antagonistische opgestelde spieren nodig om een bidirectioneel roterend gewricht te vormen. Interessant hierbij is dat positie en stijfheid van het gewricht onafhankelijk van elkaar kunnen ingesteld worden door drukregeling.

Door het gebruik van hoogwaardig aluminium, geïntegreerde kleppenblokken met on-off kleppen en de pneumatische spieren is het gewicht van deze 1,5 m hoge robot beperkt tot 28,2 kg. Door de overdimensionering van de actuatoren bezit de robot de mogelijkheid nog een belangrijk extra gewicht te torsen.

Belangrijk in het design is de modulariteit dewelke resulteert in een identieke bouw van bovenlichaam, bovenbeen en onderbeen. De bewegingsruimte van de robot is beperkt tot het sagitaal vlak, dit maakt het mogelijk om gradueel de hardware en controle-strategieën te implementeren en te testen.

De controlearchitectuur bestaat uit een hardware deel en een software deel.

Alle elektronische randapparatuur, nodig voor de besturing, werd zelf gemaakt. Hierdoor kan de elektronica beter worden afgestemd aan de vereisten van de robot. Door de complexiteit en het aantal waarin ze moesten gemaakt worden (7 microcontroller moederbords, 12 druksensoren, 12 speed-up circuits,...) was het bouwen, testen en debuggen ervan het belangrijkste werk. Extra aandacht werd besteed aan een snelle drukregeling door het implementeren van speed-up circuits die het snel openen en sluiten van de 72 kleppen mogelijk maakt.

De robot werd vervolgens gewricht per gewricht geassembleerd waarbij telkens de goede werking van de hardware werd nagegaan door het gewricht eenvoudige bewegingen te laten uitvoeren. Enkele video-opnames van de verschillende stadia staan op de CD-rom in bijlage.

Het software gedeelte bestaat uit drie grote delen:

- Lokale controller voor de microcontrollers (geschreven in assembler). Elk gewricht wordt gecontroleerd door een aparte microcontroller die de stelwaarde voor de hoek van de PC ontvangt. Uitgaande van de inputgegevens, druk in strek- en buigspier en de hoek, worden nieuwe stelwaarden voor de drukken berekend door een PID regelaar. De drukken in de spier bepalen immers de positie en stijfheid van het gewricht. Een nauwkeurige drukregeling is dan ook noodzakelijk, dit wordt geregeld door een bang-bang regelaar.
- Globale controller met gebruiksvriendelijke user interface voor de trajectgeneratie van de geleidingen (geschreven in Matlab). Als globale controle werd gekozen voor een trajectgeneratie aan de hand van objectieve parameters. De objectieve parameters zijn begin-, midden- en eindpositie van de enkels en de heup met de bijkomende eis dat in het begin en op het einde van de stap de snelheid nul moet zijn.
- Communicatieprogramma met bijhorende driver tussen globale en lokale controle (geschreven in Visual C++). Een snelle communicatie tussen globale en lokale controle is noodzakelijk, om deze reden werd binnen de vakgroep een eigen ISABus-kaart ontwikkeld. Hiervoor werd een driver geschreven. Deze driver dient om het communicatieprogramma toegang te verlenen tot de hardware van de ISABus-kaart. Het communicatieprogramma stuurt het door de globale controller gecreëerde traject door naar de robot en registreert de meetgegevens.

De controlearchitectuur van de robot Lucy is afgewerkt om te komen tot een quasi-statisch gecontroleerde tweepotige robot. Bij een quasi-statische beweging beweegt men traag genoeg zodanig dat men de coriolis- en traagheidseffecten mag verwaarlozen. De regeling hiervan is eenvoudiger en is als eerste fase geïmplementeerd om de goede werking van de hardware aan te tonen. De robot is in staat om bewegingen uit te voeren als rechte lijnen, heen opheffen en stappen te zetten.

Aan de hand van video-opnames, geplaatst op de CD rom, kan men de beweging bekijken.

5.2 Toekomstperspectieven

Twee belangrijke hardware wijzigingen moeten nog doorgevoerd worden:

- De ISAbus-kaart vervangen door een microcontroller dewelke het USB 2.0-protocol ondersteunt (chip EZ-USB FX2 van Cypress). Momenteel worden de mogelijkheden van deze microcontroller onderzocht. Het moet toelaten sneller en betrouwbaarder te communiceren, wat voor de controlestructuur van zeer groot belang is.
- Het aanbrengen van contactsensoren op de voeten. Deze moeten toelaten om te detecteren of de voet op de grond staat en de ligging van het Zero Moment Point (ZMP).

Wil men de robot sneller laten bewegen en overgaan van statisch gebalanceerd naar dynamisch stappen dan zal men zowel de globale als de lokale controle moeten verbeteren.

Momenteel wordt er reeds binnen de vakgroep onderzoek verricht naar het genereren van tijdsafhankelijke trajecten voor de actuatoren, zodanig dat een stap/looppatroon gevolgd wordt dat identiek voldoet aan een opgegeven set van 'objectieve parameters'. De ontwikkelde methodologie laat toe in real-time dergelijke trajecten te genereren voor sets van objectieve parameters, die per stap kunnen gewijzigd worden, waardoor de robot zich op een dynamisch stabiele wijze op een ongestructureerd terrein kan voortbewegen [19].

De lokale controle moet het opgelegde traject volgen, hierbij moet onderzocht worden in hoeverre het werkelijke traject mag afwijken van het opgelegde traject om volgende objectieven te bekomen: gewenste voortbeweging van de robot, op een stabiele manier en dit met een lage energieconsumptie.

Onvermijdelijk zal het nodig zijn een goede modelisatie van de robot te maken zoals het opmeten van massa's, massamiddelpunten en dynamische actuatoorkarakteristieken. Dit om de sturingsstrategieën te optimaliseren en via simulaties te kunnen valideren. Door het sterk niet-lineair karakter van zowel de koppelkarakteristieken van de spieren als de belasting zal het nodig zijn feedforward in de lokale controller in te bouwen.

Hoofdstuk 6

Symbolenlijst

C	soepelheid van de spier	-	rad/Nm
F	spierkracht	-	kN
f	dimensieloze krachtsfunctie van de spier	-	-
K	stijfheid van de spier	-	Nm/rad
L	initiële lengte van de spier	110	mm
l_o	ogenblikkelijke lengte van de spier	-	mm
M	koppel van de spier	-	Nm
m	koppelkarakteristiek van de spier	-	-
P	absolute druk	-	bar
p	relatieve druk : $p = P - P_{atm}$	-	bar (rel)
P_{atm}	omgevingsdruk	1	bar
P_m	absolute gemiddelde spierdruk	-	bar
R/L	slankheid van de spier	-	-
r	hefboomsarm	-	m
t	tijd	-	s
V	volume van de spier	-	m ³
x	X-positie	-	m
y	Y-positie	-	m
a	hoek van het gewricht	-	°
å	contractie van de spier	-	-
g	isentropische coëfficiënt	1,4	-

Hoofdstuk 7

Literatuurlijst

7.1 *Robotica*

[1] <http://world.honda.com/ASIMO/>

[2] <http://www.sony.com.au/aibo/>

[3] Nicholls E., "Bipedal Dynamic Walking in Robots", Honours Dissertation, 1998

[4] Loffler K., Gienger M. & Pfeiffer F., "Simulation and control of a biped jogging robot". In 4th international Conference on Climbing and Walking Robots, pages 867{874), Karlsruhe, Germany, 2001.

[5] Asada H. & Kanade T., "Design of Direct-Drive Mechanical Arms", ASME J. of Vibration , Acoustics, Stress, and Reliability in Design 105(3), pp. 312-316.

[6]. van der Linde R. Q., "Design, analysis and control of a low power joint for walking robots, by phasic activation of McKibben muscles", IEEE Trans. Robotics & Automation (Aug. 1999). V15(4), pp.: 599-604.

[7] Paluska D., "Design of a Humanoid Biped for Walking Research" Master's Thesis, Mechanical Engineering Department, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2000

[8] Pratt J., Gill A. & Williamson, Matthew M., "Series Elastic Actuators", Proceedings of IROS '95, Pittsburgh, PA

[9] Pratt J., Pratt G., "Intuitive Control of a Planar Bipedal Walking Robot" Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '98), Leuven, Belgium 1998.

[10] Nishino D., Yamaguchi J. & Takanashi A., "Realization of dynamic biped walking varying joint stiffness using antagonistic driven joints". In IEEE International Conference on Robotics and Automation, Leuven, Belgium, 1998.

[11] F. Daerden. "Conception and Realization of Pleated Pneumatic Artificial Muscles and their Use as Compliant Actuation Elements", PhD thesis, VUB Brussel, 1999.

[12] <http://www.shadow.org.uk>.

[13] Davis S. & Caldwell D., "The bio-metric design of a robot primate using pneumatic muscle actuators", In 4th international Conference on Climbing and Walking Robots, pages 197-204, Karlsruhe, Germany, 2001.

[14] Verrelst B., Van Ham R., Daerden F. & Lefeber D., "Design of a biped actuated by pleated pneumatic artificial muscles", Proceedings of the 5th International conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Paris, France, Sept. 2002, pp 195-202

[15] Verrelst B., Van Ham R., Daerden F. & Lefeber D., "Design of a biped actuated by pleated pneumatic artificial muscles", Proceedings of the 5th International conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Paris, France, Sept. 2002, pp 195-202

[16] Vandenhoudt J., "PWM-sturing van een antagonistisch paar pneumatische artificiële spieren", Master thesis, VUB Brussel, 2002

[17] Van Ham R., Daerden F., Verrelst B., Lefeber D. & Vandenhoudt J., "Control of pneumatic artificial muscles with enhanced speed up circuitry",

Proceedings of the 5th International conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Paris, France, Sept. 2002, pp 195-202

[18] Verrelst B., Daerden F., Lefeber D., Van Ham R.& Fabri T., “Introducing pleated pneumatic artificial muscles for the actuation of legged robots: a one-dimensional set-up”, In CLAWAR 2000: 3th International Conference on Climbing and Walking Robots, Madrid, Spain.

[19] Huang Q., Yokoi K., Kajita S., Kaneko K., Arai H., Koyachi N.& Tanie K., “Planning Walking Patterns for a Biped Robot”, IEEE Trans. on Robotics and Automation, Vol.17, No.3, June 2001.

[19] Vermeulen J., Lefeber D. & Verrelst B., “Control of foot placement, forward velocity and body orientation of a one-legged hopping robot”, accepted for publication in : Robotica, Cambridge University Press, Volume 21, pp....., 2003

[20] Collins S. H., Wisse M. & Ruina A., “A Two Legged Kneed Passive Dynamic Walking Robot”, International Journal of Robotics Research (July 2001), V20(7), p.p.: 607-615.

[21] Verrelst B., Van Ham R., Vanderborght B., Lefeber D. & Daerden F., “Lucy: a Walking Biped”, 6th National Congress on Theoretical and Applied Mechanics, Ghent, May 26-27, 2003, paper 74

[22] Verrelst B., Van Ham R., Vanderborght B., Lefeber D. & Daerden F., “Experimental Results on the First Movements of the Pneumatic Biped Lucy”. Accepted for publishing, CLAWAR 2003: 6th International Conference on Climbing and Walking Robots, Catania, Italy.

[23] Rose J. & Gamble J., “Human Walking”, Williams and Wilkins, 1994

7.2 Visual C++

[24] <http://www.jungo.com>

Nanko R., “Programmeren Visual C++”, Easy Computing, 2001

Klander L., “Core Visual C++”, Prentice Hall, 1999

<http://www.codeguru.com>

Visual studio 6.0 MSDN

7.3 Matlab

<http://www.mathworks.com>

7.4 Microcontroller

M68HC16 Family CPU16 Reference Manual. © Motorola Inc, 1997.
(CPU16RM/AD)

MC68HC16Y3/MC68HC916Y3 User's Manual. © Motorola Inc, 1997.

QSM Queued Serial Module Reference Manual (QSMRM/AD)

TPU Time Processor Unit Reference Manual (TPURM/AD)

Motorola Semiconductor Programming Note :

Quadratic Decode TPU Function (QDEC) (TPUPN20/D)

Discrete Input/Output TPU Function (DIO) (TPUPN18/D)

Hoofdstuk 8

Inhoudstabel bijlagen

Daar het aantal pagina's bijlagen te veel is om af te drukken op papier zijn ze verzameld op een CD-ROM. Het aantal pagina's staat tussen haakjes.

8.1 Programma code

8.1.1 Assembler code

Master (10)

Joint (25)

Bibliotheek routines

File met headings (4)

Code (51)

hulpprogramma (1)

Testprogramma's :

Testqspi (2)

Testtpu (4)

Testenco (6)

TestserA (1)

TestserB (1)

Testcomm (5)

8.1.2 Matlab

Trajectprogramma a.d.h.v. objectieve parameters (53)

Trajectprogramma a.d.h.v. discrete posities (78)

Verandering van robotspecificaties (20)

8.1.3 C++

Communicatieprogramma

<u>Implementation file</u>	<u>Header file</u>
CalibreerDrukSensoren.cpp (4)	CalibreerDrukSensoren.h (1)
interruptlistener_lib.cpp (3)	interruptlistener_lib.h (1)
MaakExcelFile.cpp (2)	MaakExcelFile.h (1)
MainFrm.cpp (2)	MainFrm.h (1)
PIDwaarden.cpp (2)	PIDwaarden.h (1)
RijBijvoegen.cpp (1)	RijBijvoegen.h (1)
superdriver_lib.c (3)	superdriver_lib.h (1)
tabellen.cpp (3)	tabellen.h (1)
tabellenDoc.cpp (3)	tabellenDoc.h (3)
tabellenView.cpp (18)	tabellenView.h (2)

8.2 *Elektrisch circuit*

Microcontroller
Druksensor
Speed-up circuit
Safety Board
16-bit parallel Databus Collector
ISAbus-kaart
Power board

8.3 *Elektrisch schema*

Algemeen elektrisch schema
Druksensor
Safety board

8.4 *Varia*

Connector pinout van gebruikte kabels

Lijst IRQ kanalen

Connector pinout ISABus-kaart

Priority levels

8.5 *Video-opnames*

Film 1: spieren op constante druk. Film toont dat de spieren een inherente soepelheid bezitten. (9 september 2002)

Film 2: 1 knie geactueerd, been hangt in de lucht. (27 november 2002)

Film 3: 1 knie geactueerd, been staat op de grond (17 februari 2003)

Film 4: 2 knieën geactueerd (14 maart 2003)

Film 5: 3 stappen beweging, fast forward (4x sneller) (19 mei 2003)

Film 6: 3 stappen beweging zijaanzicht (19 mei 2003)

Film 7: 3 stappen beweging vooraanzicht (19 mei 2003)

Film 8: detail van de voeten bij 3 stappen beweging (19 mei 2003)

Film 9: aanhangen van extra gewicht (19 mei 2003)

Film 10: noodstop (19 mei 2003)

Film 11: Instabiliteit voet (19 mei 2003)

8.6 *Website*

De website van Lucy Walking Robot is te vinden op www.vub.ac.be/lucy. Gedurende de werkzaamheden werd deze site regelmatig aangevuld met foto's en video-opnames over de laatste ontwikkelingen.

Op zondag 1 juni 2003 bedroeg het bezoekersaantal 1583, met bezoekers uit 50 verschillende landen.

De site is eveneens te vinden via de gebruikelijke zoekmachines (google, altavista, yahoo,...)

Verder staan er op volgende sites een link naar de website van Lucy:

Androidworld bipedal projects:

<http://www.androidworld.com/prod28.htm>

Delft Bio-robotics Lab

http://www.wbmt.tudelft.nl/mms/dbl/dbl_links_body.htm

FZI-IDS Walking Machine Catalogue

http://gate1.fzi.de/ids/public_html/index2.htm

The Legged Robot Builder Home Page

<http://www.joinme.net/robotwise/index.htm>

Biped Robot Research in the World

<http://staff.aist.go.jp/s.kajita/bipedsite-e.html>

KIMURA Lab

<http://www.kimura.is.uec.ac.jp/faculties/legged-robots.html>

8.7 CD rom

Als de internet-pagina “bijlagen” niet automatisch op het scherm komt, dient u het bestand “bijlagen.HTM” manueel te openen.
