

Het ontwerp en gebruik van een user-interface voor de xy-tafel

Citation for published version (APA):

Moltzer, M. J. (1996). *Het ontwerp en gebruik van een user-interface voor de xy-tafel: t.b.v. PGO-cursus 4 trimester 2.2 WMT- en onderzoekstroom*. (DCT rapporten; Vol. 1996.038). Technische Universiteit Eindhoven.

Document status and date:

Gepubliceerd: 01/01/1996

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

**Het ontwerp en gebruik van een
user-interface voor de xy-tafel**

*t.b.v. PGO-casus 4 trimester 2.2
WMT- en onderzoekstroom*

Meike Moltzer
Stageverslag WFW 96.038

Het ontwerp en gebruik van een user-interface voor de xy-tafel

t.b.v. PGO-casus 4 trimester 2.2 WMT- en onderzoekstroom

Auteur: Meike Moltzer
Begeleider: dr.ir. F.E. Veldpaus
Voorjaar 1996

VAKGROEP FUNDAMENTELE WERKTUIGBOUWKUNDE
FACULTEIT WERKTUIGBOUWKUNDE
TECHNISCHE UNIVERSITEIT EINDHOVEN

Samenvatting

Een van de PGO-casussen in het tweede jaar voor de WMT- en onderzoekstroom bestaat uit het modelleren van de xy-tafel en het ontwerpen van een P(I)D-regelaar hiervoor. Hiervoor moeten experimenten worden gedaan aan de echte xy-tafel met verschillende stuursignalen en regelaars. Het doel van de casus is dat studenten ontdekken wat het verschil is tussen het gedrag van een model en het werkelijke gedrag van een systeem. Hiertoe moeten ze eerst zelf een model opstellen voor de xy-tafel waarmee zij vervolgens gaan werken om een regelaar te ontwerpen.

Om de studenten direct met de xy-tafel om te laten gaan, zonder eerst zelf te moeten programmeren, is een op hen gerichte user-interface nodig. Dit programma, xytafel.m, moet de studenten de mogelijkheid bieden zelf diverse ingangssignalen en regelaars te proberen.

Door het kiezen van een geschikt stuursignaal met behulp van de user-interface kunnen de studenten een fysisch model van de xy-tafel na één of meerdere experimenten bepalen. In het programma xytafel.m is er de keuze uit vijf typen stuursignalen: een puls, een stap, een lineair of een harmonisch signaal of een combinatie van voorgaande signalen.

Het fysisch model dat wordt opgesteld door de PGO-groepen komt in enige mate overeen met een model zoals dat bekend is van de xy-tafel.

Voor de regelaar die vervolgens op basis van dit model ontworpen moet worden kan door de studenten met het programma xytafel.m gekozen worden uit een P-regelaar, een PD-regelaar of een PID-regelaar. Voor de parameters van deze regelaars moeten geschikte waarden worden gekozen.

De wijze waarop de studenten deze waarden bepalen en instellen gebeurt niet via het systematisch plaatsen van de polen, maar met behulp van "trial-and-error". De instellingen die uiteindelijk als beste worden beoordeeld door de studenten komen sterk overeen met de beste instellingen die gevonden zijn op een meer systematische manier.

Inhoudsopgave

Samenvatting

Inhoudsopgave

1. Inleiding

- 1.1 Opdrachtschrijving en doel van het onderzoek
- 1.2 Onderzoekspunten

2. Uitgangssituatie

- 2.1 Beschrijving xy-tafel
- 2.2 Beschrijving programmeeromgeving
- 2.3 Doel van de casus

3. Opzet van het programma xytafel.m

- 3.1 Programmastructuur
- 3.2 Keuzemogelijkheden van het stuursignaal
 - 3.2.1 Puls
 - 3.2.2 Stap
 - 3.2.3 Lineair
 - 3.2.4 Harmonisch
 - 3.2.5 Combinatie van signalen
- 3.3 Keuzemogelijkheden voor de regelaar
 - 3.3.1 P-regelaar
 - 3.3.2 PD-regelaar
 - 3.3.3 PID-regelaar
 - 3.3.4 Instellen van de regelaar

4. Gebruik van het programma xytafel.m

- 4.1 Gebruik van de keuzemenu's
- 4.2 Vergelijking met casus 5 ontwerpstroom

5. Resultaten bereikt door studenten met het programma xytafel.m

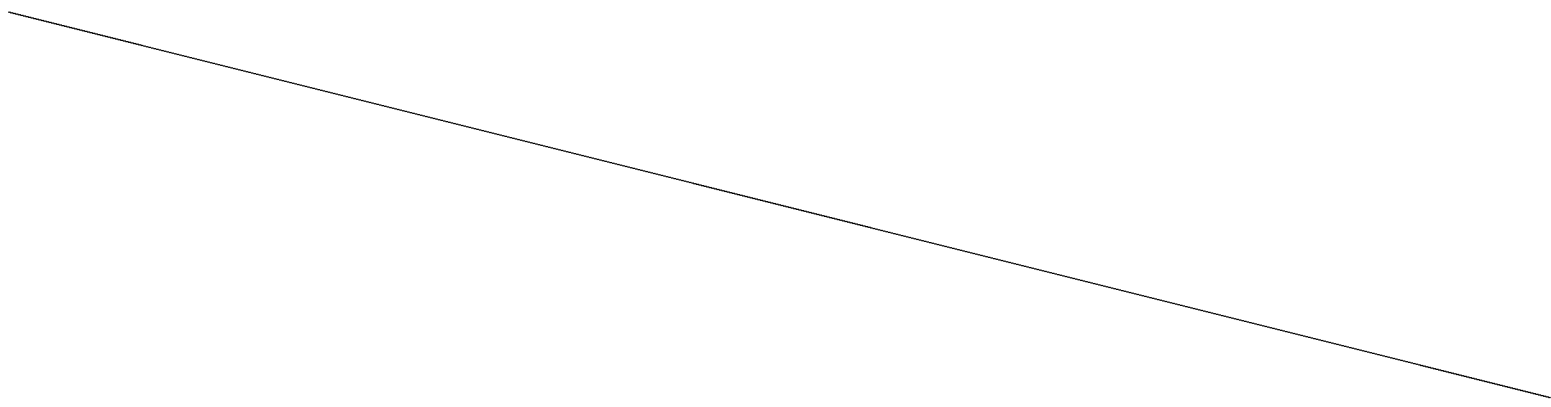
- 5.1 Identificatie systeem
 - 5.1.1 Te behalen resultaten
 - 5.1.2 Manier van aanpak door studenten t.b.v. identificatie
 - 5.1.3 Behaalde resultaten m.b.t. identificatie van het systeem
- 5.2 Instellingen regelaar
 - 5.2.1 Mogelijke optimale instellingen
 - 5.2.2 Werkwijze en resultaten van studenten
- 5.3 Vergelijking met casus 5 ontwerpstroom

6. Conclusies en aanbevelingen

- 6.1 Programma xytafel.m
- 6.2 Casusomschrijving en aansluiting collegestof

Literatuur

Bijlagen: A, B, C, D



1. Inleiding

1.1 Opdrachtomschrijving en doel van het onderzoek

Op de faculteit werktuigbouwkunde van de T.U. Eindhoven krijgen de studenten als onderdeel van de studie *Probleem Gestuurd Onderwijs (PGO)*: uitgaande van een zekere basiskennis krijgt een projectgroep van acht studenten de opdracht een praktisch probleem uit te werken.

Een van de opdrachten in het tweede jaar voor de WMT- en onderzoekstroom (zie ook bijlage A) bestaat uit het modelleren van de xy-tafel en het ontwerpen van een P(I)D-regelaar hiervoor. Hiertoe moeten ook experimenten worden gedaan aan de echte xy-tafel met verschillende stuursignalen en regelaars.

De xy-tafel wordt bestuurd met behulp van een computer. Om de studenten direct met de xy-tafel om te laten gaan, zonder dat zij eerst zelf moeten programmeren is een op hen gerichte user-interface nodig.

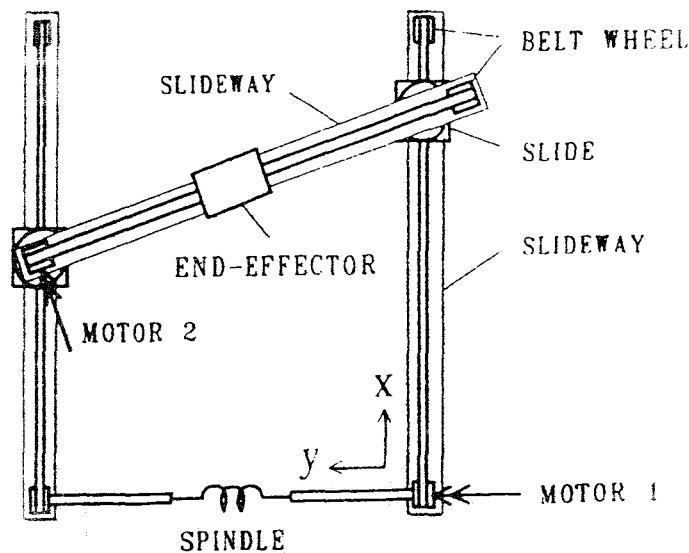


fig. 1.1 Schematische voorstelling xy-tafel

Dit programma moet de studenten de mogelijkheid bieden zelf diverse ingangssignalen en regelaars te proberen. Eisen die verder aan het programma worden gesteld zijn: het moet gemakkelijk in gebruik zijn, het mag niet zomaar 'vastlopen' en het moet beschadiging van de xy-tafel door verkeerd gekozen ingangen voorkomen.

In dit rapport wordt het ontwerp en het gebruik van het programma voor de user-interface, xytafel.m, verder uitgewerkt. Voor de bespreking van de wijze waarop het programma gebruikt is wordt gebruik gemaakt van de eindverslagen die de PGO-groepen hebben ingeleverd. Bovendien wordt gebruik gemaakt van de opgedane ervaringen bij het begeleiden van de PGO-groepen aan de xy-tafel.

De hoofdstukken 2 en 3 handelen over de opzet van het programma, in hoofdstuk 4 en 5 wordt bekeken op welke wijze het programma is gebruikt en met welke resultaten. Ten slotte worden in hoofdstuk 6 enkele conclusies getrokken over het gebruik van het programma en worden aanbevelingen gedaan voor een volgende casus met de xy-tafel.

1.2 Onderzoekspunten

Om tot een geschikte user-interface te komen zijn de volgende stappen doorlopen:

- Omschrijven wat er aanwezig is (programmeeromgeving, xy-tafel).
- Nagaan welke keuzen de studenten moeten kunnen maken bij het oplossen van de casus.
- Uitwerken van de structuur en keuzemogelijkheden van het programma.

Vervolgens zijn de volgende zaken bekeken:

- Hoe wordt het programma gebruikt door de studenten.
- Welke resultaten bereiken de studenten.
- Vergelijking met een vergelijkbare casus van de ontwerpstroom.

2. Uitgangssituatie

2.1 Beschrijving xy-tafel

De xy-tafel is een onderzoekopstelling die bestaat uit een karretje dat in een horizontaal vlak op geleidingen loopt. Figuur 1.1 geeft een schematische weergave van het systeem. De besturing van het systeem wordt verzorgd met behulp van een computer.

Tijdens het uitvoeren van de experimenten zal voor de torsie veer een veer met een oneindige veerstijfheid worden gekozen. Het systeem is hierdoor te beschrijven met een model met twee graden van vrijheid, omdat scheefstand zoals getekend is in figuur 1.1 niet voor komt.

Voor de xy-tafel zijn in deze situatie de volgende bewegingsvergelijkingen op te stellen:

$$\begin{aligned}M_x \ddot{x}(t) + B_x \dot{x}(t) &= U_x(t) + F_{wx} \\ M_y \ddot{y}(t) + B_y \dot{y}(t) &= U_y(t) + F_{wy}\end{aligned}$$

Hierin zijn:

x en y de twee vrijheidsgraden van de xy-tafel

M_x en M_y de massa's in x - en y -richting

B_x en B_y de parameters van de visceuze wrijving

$U_x(t)$ en $U_y(t)$ de stuursignalen

F_{wx} en F_{wy} de coulombse wrijvingen

2.2 Beschrijving programmeeromgeving

Voor het programmeren is gebruik gemaakt van twee verschillende programmeeromgevingen: Matlab (versie 3.5) en Borland C++. Alleen de programma's die direct de xy-tafel besturen zijn geschreven in C++, de overige files zijn geschreven in Matlab.

Er is gekozen voor Matlab omdat dit een erg eenvoudige en voor de studenten bekende omgeving is.

Bovendien wordt bij het onderwijs in de regeltechniek op de faculteit werktuigbouwkunde gebruik gemaakt van Matlab.

Het gebruik van de C++-programma's is nodig omdat Matlab-programma's in een on-line regel-situatie niet snel genoeg zijn. Voor de uitwisselbaarheid van informatie tussen Matlab, C++ en de xy-tafel wordt gebruik gemaakt van standaardbibliotheken in C.

2.3 Doel van de casus

Het doel van de casus is dat studenten ontdekken wat het verschil is tussen een model en de werkelijkheid. Hiertoe moeten ze eerst zelf een model opstellen voor de xy-tafel waarmee zij vervolgens gaan werken om een regelaar te ontwerpen.

De end-effector van de xy-tafel moet volgens de casusomschrijving een cirkel beschrijven met een straal van 0.25 [m], met een omloopsnelheid van 0.25 [m/s] en een maximale fout op de straal van 0.1 [mm].

De complete casusomschrijving is opgenomen in bijlage A.

3. Opzet van het programma xytafel.m

3.1 Programmastructuur

Met het programma xytafel.m moeten twee zaken bekeken kunnen worden:

1. De respons van de xy-tafel van een (bekend) opgelegd stuursignaal [N]
2. Het effect van een toegepaste regelaar

Eerst moet door de gebruiker gekozen worden wat men wil bekijken en vervolgens in welke richting (x en/of y) men dit wil bekijken en wat de parameters zijn van stuursignaal of regelaar. Ten slotte wordt het experiment uitgevoerd en kunnen de resultaten bekeken worden en opgeslagen worden in een datafile op diskette.

Het traject dat de gebruiker doorloopt ziet er schematisch als volgt uit:

stuursignaal of regelaar? → richting & parameters? → experiment → bekijken → opslaan

Bovenstaand schema zou ook gebruikt kunnen worden als de structuur van het programma. Dit zou echter wel de volgorde van handelingen van de gebruiker vastleggen. Om de gebruiker hier in vrij te laten is gekozen voor de volgende (globale) programmastructuur:

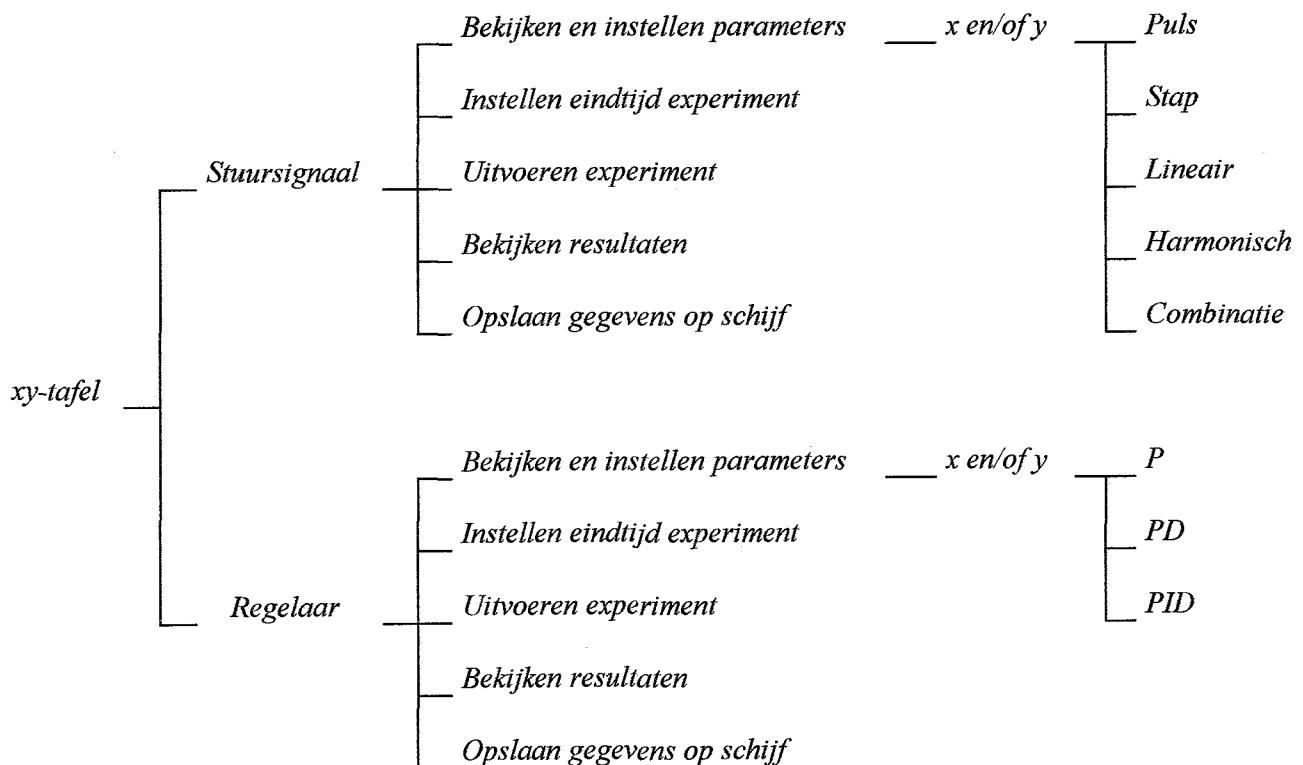


fig. 3.1 Programmastructuur xytafel.m

Bij deze programmastructuur wordt gebruik gemaakt van keuzemenu's waarbij de gebruiker steeds moet reageren op een vraag die de computer stelt. Na het uitvoeren van een commando of het afsluiten van een menu wordt steeds teruggekeerd naar het één nivo hogere menu. (In de bijlage is de programmastructuur opgenomen met de namen van de files.)

3.2 Keuzemogelijkheden voor het stuursignaal

Zoals reeds opgemerkt zijn voor de xy-tafel de volgende bewegingsvergelijkingen op te stellen:

$$\begin{aligned}M_x \ddot{x}(t) + B_x \dot{x}(t) &= U_x(t) + F_{wx} \\ M_y \ddot{y}(t) + B_y \dot{y}(t) &= U_y(t) + F_{wy}\end{aligned}$$

Bij het uitvoeren van de experimenten blijkt dat B_x en B_y nul verondersteld kunnen worden. Er moeten dus in elke richting ten minste twee vergelijkingen worden opgesteld om de twee onbekenden M en F_w voor die richting te bepalen. Door het kiezen van een geschikt stuursignaal $U(t)$ kunnen deze na één of meerdere experimenten bepaald worden. In het programma xytafel.m is er de keuze uit de volgende vijf signalen:

1. Puls
2. Stap
3. Lineair
4. Harmonisch
5. Combinatie van bovenstaande signalen

Er is gekozen voor een beperkt aantal signalen om de keuze voor de gebruiker beperkt te houden. Hierbij is gekozen voor bovenstaande stuursignalen omdat deze duidelijk herkenbaar zijn voor de gebruiker en eenvoudig te realiseren zijn met de xy-tafel.

In de volgende paragrafen worden de mogelijkheden van bovenstaande stuursignalen verder uitgewerkt. Hierbij wordt, ten behoeve van de eenvoud, gebruik gemaakt van slechts een vrijheidsgraad: z . Deze vrijheidsgraad komt overeen met x of y .

3.2.1 Puls

De puls-functie ziet er als volgt uit:

$$U(t) = a * \delta(t)$$

Hierbij is $\delta(t)$ de eenheidspuls, dat wil zeggen dat het oppervlak onder de puls gelijk is aan 1 en de hoogte van de puls oneindig hoog is.

De motoren van de xy-tafel kunnen geen oneindig grote kracht leveren. Om de eenheidspuls te benaderen is de hoogte van de puls zo groot mogelijk genomen. De maximaal te leveren kracht in x -richting en pulsduur zijn gelijk aan 246 [N] resp. aan 1/246 [s], in de y -richting aan 41 [N] resp. aan 1/41 [s]. De parameter a is het aantal keer dat de eenheidspuls achter elkaar wordt gezet, bijvoorbeeld: $10 * \delta(t)$ duurt in de x -richting 10/246 [s], $U_x(t)$ is gedurende die tijd gelijk aan 246 [N].

Door een puls aan te bieden aan de xy-tafel kunnen met slechts één experiment in elke richting alle modelparameters uit het eerder gegeven model bepaald worden. In dit experiment wordt eerst een constante, bekende kracht $U(t)$ aangeboden. Gedurende deze tijd versnelt de slede, op de slede werken nu de bekende $U(t)$ en de onbekende coulombse wrijving. Voor de slede geldt:

$$M\ddot{z}(t) = U(t) + F_w$$

Vervolgens wordt het stuursignaal gelijk aan nul. Voor de slede geldt dan:

$$M\ddot{z}(t) = F_w$$

Omdat een benadering voor de versnellingen eenvoudig uit de grafiek van de snelheid tegen de tijd af te leiden is zijn er nu twee vergelijkingen met de twee onbekenden M en F_w ontstaan.

3.2.2 Stap

De stap-functie is:

$$U(t) = a * \varepsilon(t)$$

Hierbij is $\varepsilon(t)$ de eenheidsstap met $\varepsilon(t)=0$ voor $t<0$ en $\varepsilon(t)=1$ voor $t>0$. De parameter a waarmee $\varepsilon(t)$ wordt vermenigvuldigd is hiermee de waarde van de constante kracht die vanaf $t=0$ [s] op het systeem staat.

De stap-functie kan op drie manieren gebruikt worden. Ten eerste kan hiermee gecontroleerd worden of een via een andere methode gevonden waarde van de (statische) coulombse wrijving klopt. Als voor a een kleinere waarde dan die van de coulombse wrijving wordt gekozen dan beweegt de slede niet, wordt er een grotere waarde gekozen dan komt de slede wel in beweging.

Als de waarde van de wrijvingskracht al bekend is dan kan door het gebruiken van de stap-functie de massa bepaald worden. Als de slede in beweging komt geldt:

$$M = \frac{U(t) + F_w}{\ddot{z}(t)}$$

Ten slotte kan men ook twee maal een verschillende stap-functie kiezen. Onder de voorwaarde dat het systeem bij beide stap-functies in beweging komt zijn hiermee twee vergelijkingen met twee onbekenden op te stellen, nl. twee maal $M\ddot{z}(t) = U(t) + F_w$ met twee bekende, verschillende waarden voor $\ddot{z}(t)$ en $U(t)$.

3.2.3 Lineair

Het lineaire signaal is gedefinieerd als:

$$U(t) = a * t * \varepsilon(t)$$

Door de opgelegde kracht lineair toe te laten nemen is direct te bepalen op welke tijdstip het systeem in beweging komt, welke kracht er op dat tijdstip wordt aangeboden en dus wat de statische wrijvingskracht is.

Omdat de wrijvingskracht bekend is en de toegevoerde kracht en de versnelling op elk moment te bepalen zijn is hieruit ook de massa te berekenen met $M\ddot{z}(t) = U(t) + F_w$.

3.2.4 Harmonisch

Het harmonische signaal is:

$$U(t) = a * \cos(\omega t + \varphi) * \varepsilon(t)$$

Met het harmonisch signaal is te bepalen of de wrijving in de positieve en de negatieve bewegingsrichting aan elkaar gelijk is. Ook kan hiermee onderzocht worden of de wrijving bij het in beweging komen gelijk is aan de wrijving bij het tot stilstand komen van de slede. Bij dit laatste dient wel rekening gehouden te worden met de massatraagheid van de slede.

De grootte van de wrijving kan bepaald worden door het aflezen van het tijdstip waarop de slede tot stilstand of in beweging komt en de bijbehorende kracht te bepalen.

3.2.5 Combinatie van signalen

Het afremmen van de end-effector kan het best onderzocht worden met behulp van een combinatie van signalen. De end-effector staat aan het begin van het experiment stil. Door een positieve waarde voor de parameter van de stap-functie en een negatieve waarde voor de lineaire parameter wordt steeds een kleinere kracht aangeboden. Als de aangebrachte kracht kleiner wordt dan de wrijvingskracht komt de slede (bij lage snelheden) tot stilstand. Met deze methode is globaal de waarde van de wrijving te bepalen. Deze methode werkt alleen bij lage snelheden in verband met de massa-traagheden van de sleden.

3.3 Keuzemogelijkheden voor de regelaar

Het blokschema van het te regelen systeem, de xy-tafel, is:

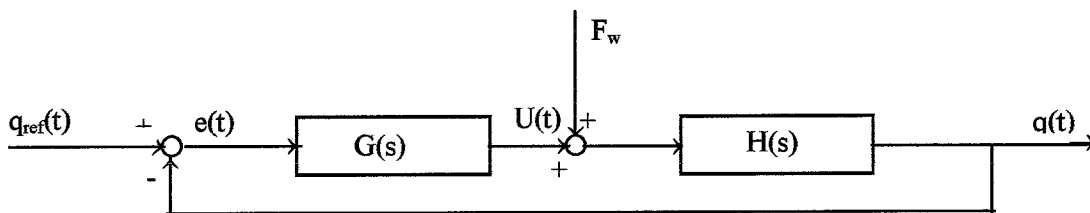


fig. 3.2 Blokschema geregelde xy-tafel

In dit systeem is $q_{ref}(t) = [x_{ref}(t) \ y_{ref}(t)]^T$ de kolom met de gewenste positie van de end-effector van de xy-tafel. In het geval van de PGO-casus is het gewenste trajectorie een cirkel met een straal van 0.25 [m] en een omloopsnelheid van 0.25 [m/s], dus $x_{ref}(t) = 0.25\cos(t)$ en $y_{ref}(t) = 0.25\sin(t)$.

De uitgang $q(t) = [x(t) \ y(t)]^T$ is de kolom met de werkelijke positie van de end-effector.

Uit de fout $e(t) = q(t) - q_{ref}(t)$ wordt met de regelaar $G(s)$ het stuursignaal $U(t) = [U_x(t) \ U_y(t)]^T$ berekend.

Op de xy-tafel werken wrijvingskrachten $F_w = [F_{wx} \ F_{wy}]^T$. De som van de stuursignalen en de wrijvingskracht is de ingang van de overdracht van de xy-tafel $H(s)$.

Na Laplace-transformatie geldt $L\{q(t)\} = H(s)L\{U(t) + F_w(t)\}$ waarbij $H(s)$ de overdrachtsmatrix van de xy-tafel is:

$$H(s) = \begin{bmatrix} \frac{1}{m_x s^2} & 0 \\ 0 & \frac{1}{m_y s^2} \end{bmatrix}$$

Voor het geregeld systeem uit figuur 3.2 geldt

$$L\{q(t)\} = H_{totaal}(s)L\{q_{ref}(t)\} + H_{stoor}L\{F_w(t)\}$$

Waarbij $H_{totaal}(s)$ en $H_{stoor}(s)$ worden gegeven door:

$$H_{totaal}(s) = \frac{G(s)H(s)}{1 + G(s)H(s)}; \quad H_{stoor}(s) = \frac{H(s)}{1 + G(s)H(s)}$$

Voor de regelaar $G(s)$ kunnen met het programma xytafel.m de volgende typen gekozen worden:

1. P-regelaar
2. PD-regelaar
3. PID-regelaar

In de hierop volgende paragrafen worden bovengenoemde drie typen regelaars besproken, ten behoeve van de eenvoud wordt hierbij steeds de regelaar van één slede bekeken.

3.3.1 P-regelaar

De relevante overdrachtsfuncties van een P-regelaar zijn:

$$G(s) = K_p ; \quad H_{\text{totaal}} = \frac{K_p}{ms^2 + K_p} ; \quad H_{\text{stoor}} = \frac{1}{ms^2 + K_p}$$

De noemer van $H_{\text{totaal}}(s)$, de karakteristieke polynoom van het geregelde systeem, is:

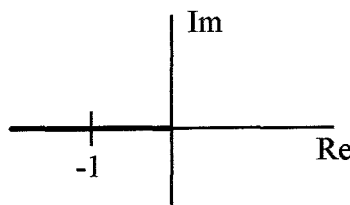
$$ms^2 + K_p = 0$$

De polen van het geregelde systeem liggen op $s = \pm j\sqrt{K_p/m}$. Uit vergelijking met de algemene vergelijking $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$ volgt dat $\zeta = 0$ en $K_p = m\omega_n^2$.

Opvoeren van de versterking $K_p \rightarrow \infty$ resulteert in $H_{\text{totaal}}(s) \rightarrow 1$ en $H_{\text{stoor}} \rightarrow 0$. Hiermee zou de doelstelling van de regeling in theorie volledig worden bereikt. Bij de toepassing van deze gedachte ontstaan echter wel stabiliteitsproblemen. Voor de rondgaande versterking geldt:

$$\overset{\circ}{H}(s) = \frac{K_p}{ms^2} \Rightarrow \overset{\circ}{H}(j\omega) = -\frac{K_p}{m\omega^2}$$

Of in een Nyquist-plot:



figuur 3.3 Overdrachtsfunctie bij P-regelaar

Deze overdrachtsfunctie gaat voor elke waarde van $K_p > 0$ door het punt -1 en is dus altijd instabiel. Om een betere regeling te verkrijgen blijkt het nuttig de regelaar ook dynamische eigenschappen te geven: in ieder geval een differentiërende actie om demping in het geregelde systeem te introduceren en eventueel nog een integrerende actie.

3.3.2 PD-regelaar

Zoals blijkt uit de voorgaande paragraaf is de fase van $\overset{\circ}{H}(j\omega)$ gelijk aan $-\pi$. Door een D-actie toe te voegen kan de fase-achterstand (met name) voor hoge frequenties teruggedrongen worden. Voor een PD-regelaar geldt de overdrachtsfunctie:

$$G(s) = K_p + K_d s$$

De totale overdrachtsfunctie van het geregelde systeem en die van de verstoring zijn nu:

$$H_{\text{totaal}} = \frac{K_d s + K_p}{ms^2 + K_d s + K_p}; \quad H_{\text{stoor}} = \frac{I}{ms^2 + K_d s + K_p}$$

De Nyquist-plot ziet er met de PD-regelaar als volgt uit:

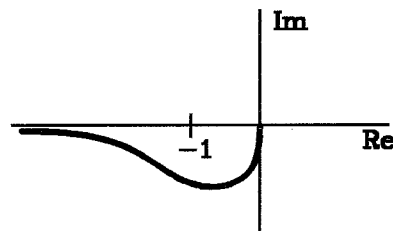


fig. 3.4 Overdrachtsfunctie bij PD-regelaar

Het toevoegen van de D-actie aan de regelkring zorgt ervoor dat het systeem stabiel wordt. De statische afwijking van de uitgang, die gedefinieerd is als:

$$e(\infty) = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s)$$

met: $E(s) = L\{q_{\text{ref}}(t)\} - L\{q(t)\}$

wordt hiermee gelijk aan nul. Op vergelijkbare wijze volgt als gevolg van de verstoring F_w een afwijking met de waarde $1/K_p$. Opvoeren van de versterking $K_p \rightarrow \infty$ zal tot het gewenste regelgedrag leiden. Hierbij geldt echter wel dat er geen modelfouten aanwezig zijn.

3.3.3 PID-regelaar

Door het toevoegen van een integrerende actie wordt de overdrachtsfunctie van de regelaar:

$$G(s) = K_p + K_d s + K_i / s$$

Dit leidt tot de volgende overdrachtsfuncties van het geregelde systeem:

$$H_{\text{totaal}} = \frac{K_d s^2 + K_p s + K_i}{ms^3 + K_d s^2 + K_p s + K_i}; \quad H_{\text{stoor}} = \frac{s}{ms^3 + K_d s^2 + K_p s + K_i}$$

Het toevoegen van de integrerende actie leidt er toe dat de statische afwijking van zowel het regelgedrag als het stoorgedrag nul gemaakt wordt. Om de statische afwijking te onderdrukken hoeft de regelaar alleen voor lage frequenties zich als integrator te gedragen.

De Nyquist-plot voor het systeem met PID-regelaar is:

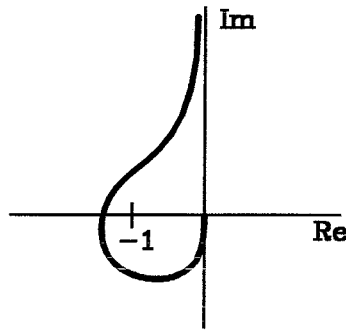


fig. 3.5 Overdrachtsfunctie bij PID-regelaar

3.3.4 Instelling van de regelaar

Aangenomen wordt dat de procesparameters zijn bepaald zijn door het aanbieden van een of meerdere stuursignalen. Dan doet zich de vraag voor welke waarden aan de factoren K_p , K_d en K_i moeten worden toegekend. Het gesloten systeem moet bij de gekozen waarden een optimaal regel- en stoorgedrag vertonen.

Uit de vergelijkingen van de P-regelaar volgt dat bij het aanbieden van een stapvormig signaal $q_{ref} = a \cdot \varepsilon(t)$ en afwezigheid van verstoringen de statische afwijking gelijk wordt aan $e(\infty) = 0$.

Als het stoorsignaal gelijk is aan de een constante wrijvingskracht, dan volgt voor de steady-state responsie van de fout $e(\infty) = F_w / K_p$.

Als de fout kleiner moet zijn dan e_{max} , dan moet K_p tenminste gelijk zijn aan $e_{max} \cdot F_w$.

Om inschakelverschijnselen weg te regelen zonder de eigenschappen van de steady-state responsie te veranderen kan een D-actie volgens paragraaf 3.3.2 worden toegevoegd.

De open-loop overdrachtsfunctie wordt hiermee:

$$G(s)H(s) = \frac{K_d s + K_p}{ms^2}$$

De karakteristieke vergelijking $ms^2 + K_d s + K_p = 0$ kan geschreven worden in de standaardvorm $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$ met ongedempte eigenfrequentie ω_n en dempingsfactor ζ . Er geldt:

$$\omega_n^2 = \frac{K_p}{m} ; \quad \zeta = \frac{K_d}{2m\omega_n} = \frac{1}{2} \frac{K_d}{\sqrt{mK_p}}$$

Voor de polen geldt:

$$s = -\frac{K_d}{2m} \pm \sqrt{\frac{K_d^2}{4m^2} - \frac{K_p}{m}}$$

Om de statische afwijking terug te dringen kan vervolgens een integrerende actie aan de regelaar worden toegevoegd. De open-loop overdrachtsfunctie is nu:

$$G(s)H(s) = \frac{K_d s^2 + K_p s + K_i}{ms^3}$$

De closed-loop overdracht heeft nu drie polen, twee ervan worden geplaatst op dezelfde plaats als de twee polen bij de beste instelling van de PD-regelaar, de derde wordt links van dit toegevoegd complexe paar geplaatst. De waarde van de derde pool heeft, volgens de vuistregel, de waarde $s = -2\zeta\omega_n$. met deze waarde heeft de extra pool ongeveer hetzelfde gedrag als de twee andere polen.

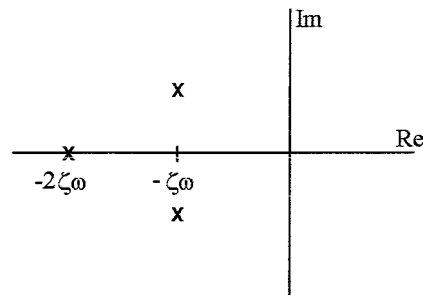


fig. 3.6 Plaats van de polen bij PID-regelaar

Uit de overdrachtsfuncties $H_{\text{totaal}}(s)$ en $H_{\text{stoor}}(s)$ volgt bij de PID-regelaar dat niet alleen de volgfout nul wordt, maar ook de steady-state responsie van het stoorsignaal nul is. Immers volgt uit de overdrachtsfuncties in paragraaf 3.3.3: $H_{\text{totaal}}(0) = 1$, dus $e_{ss} = 0$ en $H_{\text{stoor}}(0) = 0$.

Bovenstaande beschouwing is echter alleen volledig correct als de xy-tafel juist gemodelleerd is. In het gebruikte model is bijvoorbeeld niet opgenomen dat de wrijvingskracht F_w niet constant is maar van teken wisselt bij het omkeren van de bewegingsrichting. De integrerende actie kan hierdoor niet volledig effectief zijn omdat eerst een fout moet zijn opgebouwd voordat de integrerende actie effect heeft. Bij het in beweging komen en het wisselen van de bewegingsrichting van een slede wordt niet het gewenste gedrag behaald.

4. Gebruik van het programma xytafel.m

4.1 Gebruik van de keuzemenu's

Voor het programma xytafel.m is gebruik gemaakt van keuzemenu's. Deze menu's leiden de studenten door middel van vragen vanzelf door alle mogelijkheden van het programma. De studenten hoeven zich dus niet van te voren af te vragen wat de mogelijkheden zijn maar komen deze vanzelf tegen.

Het voordeel van het gebruik van keuzemenu's is dat de studenten inderdaad snel aan de slag zijn met het programma. Bovendien is de kans klein dat er vergeten wordt bepaalde informatie op te geven omdat het programma naar deze informatie vraagt.

Een probleem bij het gebruik van deze keuzemenu's is echter dat de studenten op een bepaald moment zo gewend zijn aan een bepaalde volgorde van commando's dat het beeldscherm niet meer goed bekeken wordt en eventuele foutmeldingen niet meer opgemerkt worden.

Een ander nadeel is dat doordat het programma xytafel.m een afgesloten geheel is, er tijdens het gebruik van het programma geen gebruik wordt gemaakt van losse Matlab-commando's. Dat kan makkelijk zijn voor, bijvoorbeeld, het tussentijds uitrekenen van de regelparameters. Dit is nadelig omdat de studenten hierdoor vrij willekeurige waarden voor P, I en D te kiezen. Met een nette berekening met behulp van poolplaatsing zijn wellicht (sneller) betere resultaten te bereiken dan met de "trial-and-error-methode" die nu door de studenten gebruikt wordt.

4.2 Vergelijking met casus 5 ontwerp-stroom

Studenten van de ontwerp-stroom hebben ook een casus uitgewerkt met de xy-tafel. Tussen deze casus en de casus van de onderzoek/WMT-stroom zijn drie belangrijke verschillen aan te wijzen:

- Casusomschrijving:
De onderzoek/WMT-stroom moet een cirkel beschrijven met de xy-tafel, de ontwerpers een rechte lijn in één richting (x of y). Bovendien is bij de casusomschrijving van de onderzoek/WMT-stroom sprake van een black-box model en bij de ontwerpersstroom wordt gesproken over een niet starre robot.
- Software:
De ontwerpers hebben de beschikking over losse commando's in Matlab in plaats van een menugestuurd programma. Er bestaat in het programma voor de ontwerpers ook een mogelijkheid om een andere meetmethode te gebruiken, er is de keuze uit meten met de camera en meten met de motor-hoek. Bovendien konden de ontwerpers kiezen voor wrijvingscompensatie en het gebruik van een invers model naast het ingeven van regelparameters P, I en D.
- Voorkennis:
De onderzoekers en WMT'ers moeten in het tweede jaar het vak "Regelen van multivariabele systemen" (4A560) volgen waarin toestandbeschrijving en MIMO-systemen centraal staan. De ontwerpers hebben het vak "Regeltechniek" (4A550), waarin de praktische aspecten van het regelen meer aandacht krijgen.

Hierdoor zijn er ook grote verschillen aan te wijzen bij het uitwerken van de casus door de verschillende PGO-groepen:

- Doordat in de casusomschrijving voor onderzoek en WMT sprake is van een cirkel, en het verschil tussen sturen en regelen voor deze studenten niet duidelijk is, wordt in het begin direct naar het gebruik van een harmonisch verlopend stuursignaal gegrepen.
- In de casusomschrijving van de WMT- en onderzoekstroom is sprake van een black-box model. Bij het identificeren van de xy-tafel is dit echter niet de meest voor de hand liggende methode. Dit levert problemen op voor de studenten bij het begin van de casus.
- Voor de ontwerpstroom is de casus een toepassing van de collegestof waar zij in het vak 4A550 mee bezig zijn. Voor de WMT- en ontwerpstroom is de casus vooral nuttig voor het ophalen van kennis uit het eerste jaar en het meer concreet maken van begrippen.
- Omdat de ontwerpstroom direct van Matlab-commando's gebruik maakt (in plaats van een menustructuur) is er in de beginfase meer uitleg nodig over de programmatuur.
- De ontwerpers krijgen voorbeelden aangeboden voor het aanbieden van het stuursignaal met een geschikte orde van grootte, zij hebben dan ook eerder geschikte meetwaarden.

5. Resultaten bereikt door studenten met het programma xytafel.m

5.1 Identificatie systeem

5.1.1 Te behalen resultaten

Volgens onderzoek naar de xy-tafel [2] gelden voor de xy-tafel de volgende massa's:

$massa_{balk}$	= 6.9 [kg]
$massa_{motor2}$	= 1.6 [kg]
$massa_{end-effector}$	= 2.3 [kg]

De voor het op te stellen model in rekening te brengen massa's mJ_1 en mJ_2 van motor 1 resp. motor 2 volgen uit de massa-traagheid van de motor en de massa-traagheden van de overbrengingen en van de encoders en uit de transmissieverhoudingen. Deze traagheden hebben omgerekend naar kilogrammen de volgende waarden:

$$mJ_1 = 22.1 \text{ [kg]}$$

$$mJ_2 = 1.58 \text{ [kg]}$$

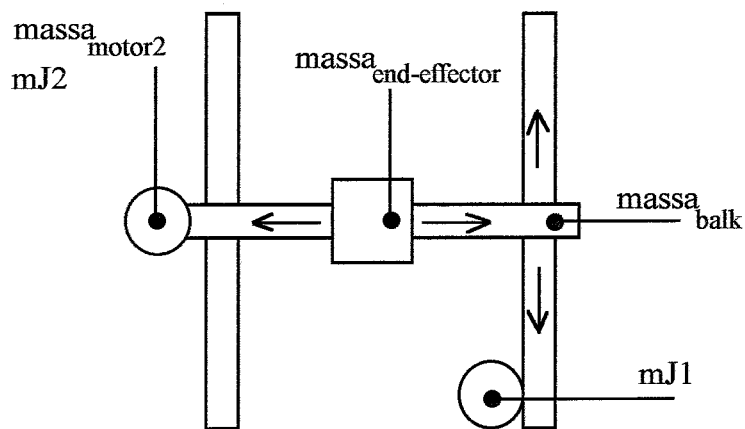


fig. 5.1 Schematische weergave massa's van de xy-tafel

De totale massa in de x-richting is hiermee:

$$M_x = massa_{balk} + massa_{motor2} + massa_{end-effector} + mJ_1 = 62.9 \text{ [kg]}$$

Voor de in y-richting te bewegen massa geldt:

$$M_y = massa_{end-effector} + mJ_2 = 3.88 \text{ [kg]}$$

In eerder onderzoek [2] is voor de coulombse wrijvingskracht in de x-richting gevonden:

$$F_{wx} \approx 50 \text{ [N]}$$

Om bovenstaande grootheden in de casus eenvoudig te bepalen moet aan de xy-tafel een kracht worden aangeboden. Voor het aanbieden van een kracht is er de keuze uit de stuursignalen zoals beschreven in paragraaf 3.2.

5.1.2 Manier van aanpak door studenten t.b.v. identificatie systeem

Na het maken van een keuze uit de verschillende typen stuursignalen moet de grootte van het stuursignaal worden vastgelegd. Dit blijkt voor een groot deel van de studenten een moeilijk probleem. Vaak wordt gestart met een experiment met een stuursignaal in de orde grootte van 1 [N]. Ook het verband tussen het uitvoeren experiment en het doel dat de studenten hiermee voor ogen hebben is niet altijd duidelijk.

Alle PGO-groepen hebben de volgende bewegingsvergelijkingen voor de xy-tafel gevonden:

$$\begin{aligned}M_x \ddot{x}(t) &= U_x(t) + F_{wx} \\ M_y \ddot{y}(t) &= U_y(t) + F_{wy}\end{aligned}$$

Geen van de groepen heeft veer- en dempingsconstanten in deze vergelijkingen opgenomen omdat zij constateren dat er bij een constante kracht er een constante versnelling van het systeem ontstaat. Er zijn hiermee nog twee onbekenden per richting over: de wrijvingskracht en de massa.

Om eerst de wrijvingskracht te bepalen kiezen drie groepen ervoor om de kracht lineair laten oplopen om te onderzoeken op welk moment de slede in beweging komt en wat dus de statische wrijvingskracht is. Een van deze drie groepen bekijkt vervolgens wat er gebeurt als de kracht lineair afneemt en wat dus de dynamische wrijvingskracht zou kunnen zijn.

Een andere groep gebruikt een harmonisch verlopende kracht om de statische en dynamische wrijvingskrachten te bepalen.

Door bovenstaande vier groepen wordt na het bepalen van de wrijvingskracht de massa bepaald door een constante kracht aan de xy-tafel aan te bieden.

Slechts een groep bepaalt de wrijvingskracht niet met een apart experiment maar zet twee maal een constante kracht op de slede waardoor er twee vergelijkingen met twee onbekenden worden gevonden.

5.1.3 Behaalde resultaten m.b.t. identificatie van het systeem

De uitkomsten die de studenten verkregen hebben uit bovengenoemde experimenten zijn opgenomen in bijlage D.

De verkregen waarden van de massa's en wrijvingskrachten zijn van de goede orde van grootte maar wel allemaal kleiner dan de waarden volgens [2].

Wat verder opvalt is de grote spreiding van de gevonden waarden van de parameters. Voor de standaarddeviatie geldt:

$$\sigma = s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

Deze is omgerekend in percentages minimaal 20% bij de schatting van de wrijvingskracht in de x-richting en maximaal 40% bij de schatting van de massa in x-richting.

Deze grote spreiding is waarschijnlijk te wijten aan het feit dat de resultaten op een onzorgvuldige wijze verkregen zijn. Niet allen zijn alle waarden bepaald uit slechts een meting, maar ook de wijze waarop de waarden bepaald worden uit de meting is erg onnauwkeurig.

5.2 Instellingen regelaar

5.2.1 Mogelijke optimale instellingen

In de casus wordt verondersteld dat een nauwkeurigheid van 0.1 [mm] op de te volgen straal van de cirkel bereikt moet kunnen worden. De samplefrequentie is 100 [Hz]. De positie van de end-effector wordt bepaald uit de opgemeten motorhoeken. Onderstaande regelparameters zijn met behulp van de karakteristieke vergelijkingen bepaald, zoals beschreven in paragraaf 3.3.4. Hierbij is $M_x = 40$ [kg] gekozen en $M_y = 3$ [kg]. De regelparameters zijn met deze massa-schattingen dus slechts een indicatie van de mogelijkheden.

	P_x	I_x	D_x	ω_x	ζ_x	s_x	P_y	I_y	D_y	ω_y	ζ_y	s_y
P	15000	-	-	19	0	$s_{1,2} = \pm 19j$	1000	-	-	18	0	$s_{1,2} = \pm 18j$
PD	9000	-	840	15	0.7	$s_{1,2} = -10.5 \pm 10.7j$	2700	-	126	30	0.7	$s_{1,2} = -21 \pm 21.4j$
PID	7578	28672	896	8	0.7	$s_{1,2} = -5.6 \pm 5.7j$ $s_3 = -11.2$	2877	24494	151	18	0.7	$s_{1,2} = -12.6 \pm 12.9j$ $s_{1,2} = -25.2$

Door steeds eerst ζ en ω_n te kiezen en vervolgens hiermee P , I en D te berekenen kunnen direct de polen geplaatst worden. In enkele experimenten kan bepaald worden bij welke ω_n en dus welke kringversterking het systeem instabiel wordt.

Met deze instellingen is een maximale nauwkeurigheid van de straal van de te lopen cirkel te bereiken van ruim 5 [mm]. Op een straal van 0.25 [m] is dit een fout van 2%. De gewenste nauwkeurigheid van 0.1 [mm] (0.04%) is blijkaar bij deze omstandigheden niet te bereiken.

Met een mogelijkheid in het programma voor het toepassen van wrijvingscompensatie zijn wellicht betere prestaties te bereiken. Ook met een hogere samplefrequentie zou voor een grotere nauwkeurigheid kunnen zorgen. Een samplefrequentie sf [Hz] heeft namelijk een tijdvertraging $t_d = 1.5/sf$ [s] tot gevolg. Bij een hogere samplefrequentie en dus een kleinere tijdvertraging kan de kringversterking verder worden opgevoerd.

5.2.2 Werkwijze en resultaten van studenten

Uit de eindverslagen van de PGO-groepen blijkt dat de door de studenten gebruikte regelparameters niet gevonden zijn door het plaatsen van de polen maar op een trial-and-error manier door te kijken hoe het systeem reageert op een verandering van een of meerdere parameters.

De uiteindelijke instellingen (zie bijlage D) komen wel in enige mate overeen met de instellingen zoals deze gevonden kunnen worden door poolplaatsing.

Opgemerkt moet worden dat de waarden van de integrerende regelactie bij een aantal groepen niet juist kan zijn wegens een fout in het programma. Deze fout is inmiddels hersteld.

De uiteindelijke nauwkeurigheid die bereikt is door de meeste PGO-groepen is 5 [mm] en komt overeen met de nauwkeurigheid die te bereiken is door gebruik te maken van poolplaatsing.

In de verslagen van de PGO-groepen is de evaluatie van de resultaten erg mager. De bereikte resultaten worden niet onderbouwd. Bovendien worden er weinig conclusies getrokken over het feit dat de gewenste nauwkeurigheid niet bereikt wordt. Aanbevelingen over de wijze waarop wel voldaan zou kunnen worden aan de eisen worden weinig gedaan.

5.3 Vergelijking met casus 5 ontwerpstroom

Bij de uitwerking van een soortgelijke casus door de ontwerpstroom zijn een aantal duidelijke verschillen en overeenkomsten aan te wijzen.

Ten eerste is de wijze waarop de ontwerp-groepen de massa's en wrijvingskrachten berekenen bij alle groepen gelijk. Er wordt steeds gebruik gemaakt van een puls gedurende een bepaalde tijd (overeenkomstig met de puls in paragraaf 3.2.1). De onderzoek/WMT-groepen hebben van verschillende methoden gebruik gemaakt.

Bij het verwerken van de metingen wordt even onzorgvuldig te werk gegaan als de onderzoek/WMT-groepen. De waarden worden, op één groep na, slechts uit een enkele meting bepaald.

Bij het instellen van de regelparameters wordt, in tegenstelling tot de onderzoek/WMT-stroom, door alle ontwerp-groepen gebruik gemaakt van het opvoeren van de bandbreedte van het systeem (zoals in paragraaf 3.3.4). Evenals de onderzoekers en WMT-ers wordt een maximale nauwkeurigheid bereikt met de PID-regelaar van ongeveer 5 mm.

Door de ontwerpers is tevens gebruik gemaakt van de mogelijkheid van het invoeren van wrijvingscompensatie, met deze mogelijkheid in het computerprogramma kon een hogere nauwkeurigheid worden bereikt. Ten slotte bestond in software voor de ontwerpstroom de mogelijkheid twee verschillende meetmethoden te gebruiken, het meten van de positie van de end-effector met behulp van de motor-hoek en met behulp van de camera. Van deze mogelijkheid hebben alle ontwerp-groepen gebruik gemaakt.

In de verslagen van de PGO-groepen van de ontwerpstroom komen een aantal onderwerpen naar voren die ontbreken in de verslagen van de onderzoek/WMT-stroom. Ten eerste is de wijze van instellen van de regelaar beter theoretisch onderbouwd. Een groot deel van de PGO-groepen maakt hierbij gebruik van Nyquist-plots van de overdracht van het geregelde systeem. Deze theorie ontbrak voor een groot deel bij de PGO-verslagen van de onderzoek/WMT-stroom. Bovendien worden door bijna alle ontwerp-groepen enkele conclusies getrokken over wat de oorzaak kan zijn van een minder goede regelprestatie. Deze conclusies konden getrokken worden na het onderzoeken van het gebruik van wrijvingscompensatie en het gebruik van verschillende meetmethoden.

Toch ontbraken ook in de PGO-verslagen van de ontwerpstroom opmerkingen over de hoge waarden van de massa's, hoewel deze opmerkingen tijdens het uitvoeren van de experimenten aan de xy-tafel wel gemaakt zijn.

Oorzaken van de verschillen tussen de twee stromen zouden kunnen zijn:

- Een verschillende casus-omschrijving
- Gebruik van andere software
- Verschillende voorkennis
- Meer uitleg aan de ontwerpers door de ervaringen met de onderzoekers en WMT'ers

De eerste drie punten zijn al uitgewerkt in paragraaf 4.2. Het laatste punt zal altijd moeilijk te vermijden zijn bij het opstarten van nieuwe casussen.

6. Conclusies en aanbevelingen

6.1 Programma xytafel.m

Het programma xytafel.m werkt naar behoren. Het is robuust en heeft de mogelijkheden die in hoofdstuk 3 beschreven zijn. Om de studenten direct en zelfstandig met de xy-tafel te laten experimenteren blijkt de menu-structuur een geschikte keuze te zijn.

Om een groep helemaal zonder begeleiding te laten werken is het aan te raden meer documentatie (op schrift) samen te stellen zodat de groep zich beter kan voorbereiden en ook meer mensen actief betrokken kunnen zijn bij de experimenten.

Binnen het programma xytafel.m moet een uitgebreidere hulpfunctie worden opgenomen om een PGO-groep snel aan het werk te krijgen. Deze hulp zal bij voorkeur moeten bestaan uit grafische afbeeldingen omdat deze beter te zien zijn voor een hele groep. De informatie kan bijvoorbeeld bestaan uit voorbeelden waardoor de studenten een idee krijgen wat een geschikte orde van grootte is van een bepaalde parameter. Met een demo-programma zouden ook snel alle mogelijkheden getoond kunnen worden.

Om de studenten meer bewust met Matlab om te laten gaan is het raadzaam om in de programmastructuur de namen van de gebruikte m-files te gebruiken. Dit zou op een zodanige wijze geprogrammeerd moeten worden dat het mogelijk wordt om zowel met een menustructuur als met losse Matlab-commando's te werken. Voorwaarde hierbij is wel dat de studenten over meer kennis van Matlab beschikken dan nu het geval is.

Om betere regelprestaties te krijgen is het raadzaam om ook een mogelijkheid voor wrijvingscompensatie in te bouwen in het programma. Met een hogere samplefrequentie zijn ook betere prestaties te bereiken omdat hiermee de tijdvertraging kleiner wordt en de kringversterking dus hoger opgevoerd kan worden.

6.2 Casusomschrijving en aansluiting met collegestof

Bij het uitvoeren van de casus was veel hulp nodig om de PGO-groepen binnen de beschikbare tijd over de juiste meetgegevens te laten beschikken. Dit was niet alleen de wijten aan het feit dat er weinig informatie over het programma beschikbaar was, maar ook omdat er in de casusomschrijving verwezen werd naar een black-box model. De xy-tafel leent zich echter beter voor het opstellen van bewegingsvergelijkingen zoals geleerd wordt bij de vakken Dynamica (4A230) en Systeemanalyse (4A320) in het eerste jaar.

Bij het oplossen van de casus wordt vooral gebruik gemaakt van kennis uit het eerste jaar (4A320, Systeemanalyse). De kennis die nu wordt opgedaan in het vak "Regelen van multivariabele systemen" (4A560) wordt niet of nauwelijks aangesproken.

De casus besteedt geen aandacht aan de plaats en invloed van de meting (in dit geval een meting van de motor-hoek). Hiermee is de casus wel gemakkelijk uit te breiden omdat op de xy-tafel ook de mogelijkheid bestaat om de positie van de end-effector direct met een camera te meten.

Voor de meeste studenten van de WMT- en onderzoekstroom is bovendien het verschil tussen sturen en regelen volstrekt onduidelijk. Hieraan zou wellicht meer aandacht kunnen worden besteed in de colleges.

Verwerking van de meetgegevens gebeurt op een erg onzorgvuldige wijze. Wellicht kan de casusomschrijving zo aangepast worden dat het duidelijk is dat dit ook een belangrijk aspect van een experiment is.

De casus over de xy-tafel is zeker nuttig omdat veel oude kennis wordt opgefrist en enkele nieuwe probleemgebieden worden aangestipt. Het zou daarom wel aan te raden zijn als de casus zich meer vooraan in het trimester zou bevinden. Op deze manier is de oude kennis die nodig is voor het vak "Regelen van multivariabele systemen" (4A560) snel opgehaald. Bovendien kunnen problemen die in het oplossen van de casus naar voren komen gebruikt worden als praktische voorbeelden in dit collegevak. Het verplaatsen van de casus naar het begin van het trimester heeft echter ook nadelen. De studenten zijn namelijk nog niet helemaal bekend met de benodigde regelconcepten, het begrip stoorgedrag etc. Het lijkt daarom raadzaam de casus te splitsen in twee delen van elk 20 uur. In het eerste deel aan het begin van het trimester kan dan aandacht besteed worden aan de identificatie van het systeem met behulp van sturing. Hierbij kunnen de problemen geïnventariseerd worden die later in het college behandeld worden zoals wrijvingscompensatie en tijdvertraging door de samplefrequentie.

Literatuur

- [1] **Banens, Jos**
Documentation on TCE modules
rapport WFW 94.050, T.U. Eindhoven, 1994

- [2] **Bossink, Saskia**
Model met vijf graden van vrijheid voor de xy-tafel
rapport WFW 95.025, T.U. Eindhoven, 1995

- [3] **Close, Charles M. & Frederick, Dean K.**
Modeling and analysis of dynamic systems, second edition
ISBN 0-395-55114-5, Houghton Mifflin Company, 1993

- [4] **Kok, J.J.**
Werktuigkundige regeltechniek I
Collegedictaat nr. 4592, T.U. Eindhoven, 1991

- [5] **Studenten WMT- en onderzoekstroom**
Eindverslagen casus 4: black-box
T.U. Eindhoven, 1996

- [6] **The Math Works**
Matlab reference guide
1992

Bijlage A

De opdrachtomschrijving van casus 4 voor de studenten WMT en onderzoek

A1 Omschrijving voor studenten

In een mode-atelier wil men met een laser patronen snijden uit lappen stof. De laser moet daarvoor evenwijdig aan een vlakke tafel van $1\text{[m]}*1\text{[m]}$ bewogen kunnen worden. Men wil hiervoor hetzelfde mechanisme gaan gebruiken als bij de xy-tafel in het laboratorium van de vakgroep WFW. Dat mechanisme wordt aangedreven met een motor voor de x-richting en een voor de loodrecht daarop staande y-richting. De aansturing van het mechanisme geschiedt door het aanbieden van geschikt gekozen spanningen aan de voedingsversterkers van die motoren.

De gewenste beweging van de laser wordt gegeven door een functie $x_d=x_d(t)$ voor de x-beweging en een functie $y_d=y_d(t)$ voor de y-beweging. Bij het starten van het mechanisme zal de laser in het algemeen niet op de juiste positie staan. Bovendien kunnen er tijdens bedrijf door allerlei oorzaken afwijkingen van de gewenste positie optreden. Het is daarom nodig voor dit mechanisme regeling te ontwerpen die zorgt dat de laser (in ieder geval na enige tijd) de gewenste beweging uitvoert en blijft uitvoeren. Die gewenste beweging is hier een cirkel met een straal van $0,25\text{[m]}$, die met een zo goed mogelijk constante snelheid van $0,10\text{[m/sec]}$ doorlopen moet worden. Tijdens het snijden mag de positie-fout van de laser maximaal 1[mm] zijn.

Bij het ontwerpen van de regeling is een model nodig. In principe kunnen bij het opstellen daarvan twee wegen gevolgd worden. Allereerst kan met geschikt gekozen experimenten worden nagegaan hoe de uitgangen (de x- en y-coördinaat van de laser) worden beïnvloed door de ingangen (de spanningen aan de voedingsversterkers). Dit levert een black box model. Daarnaast is het vaak mogelijk om met fysische basiswetten en met veronderstellingen over de eigenschappen van de onderdelen van het mechanisme een wiskundig model, een wit model, op te stellen. Vaak zijn ook dan experimenten nodig om de parameters in het model te bepalen.

In dit geval is het de bedoeling om een black box model te gebruiken. Voor de benodigde experimenten kan gebruik worden gemaakt de eerder genoemde xy-tafel in het laboratorium van de vakgroep WFW. Ontwerp een regeling en laat zien dat daarmee de specificaties met betrekking tot de beweging van de laser gehaald kunnen worden.

A2 Omschrijving voor tutoren

Bij deze casus zal het voor de studenten uit de casus-omschrijving vrij duidelijk zijn wat er in dit geval van hen verwacht wordt. Bij de uitwerking zullen zij zich terdege moeten realiseren welk type model zij nodig hebben en hoe zij dat model denken te gaan gebruiken bij het ontwerpen van een regeling. De vereiste voorkennis voor deze casus is al in het eerste jaar in de cursus Systemanalyse aangedragen. De te gebruiken literatuur bestaat dan ook uit het bij die cursus gebruikte boek van Close en Frederic.

Het te bepalen model bestaat uit de transferfunctie H , die aangeeft wat (in het s-domein) het verband is tussen enerzijds de ingangen op het beschouwde systeem en anderzijds de uitgangen van dat systeem. In dit geval is H een $(2*2)$ diagonaalmatrix. Een belangrijk aspect van deze casus is dat de studenten nagaan met welke experimenten zij de diagonaalcomponenten van H kunnen bepalen. Zodra dat gebeurd is kunnen zij met de kennis uit Systemanalyse een P(I)D-regelaar ontwerpen waarmee de specificaties

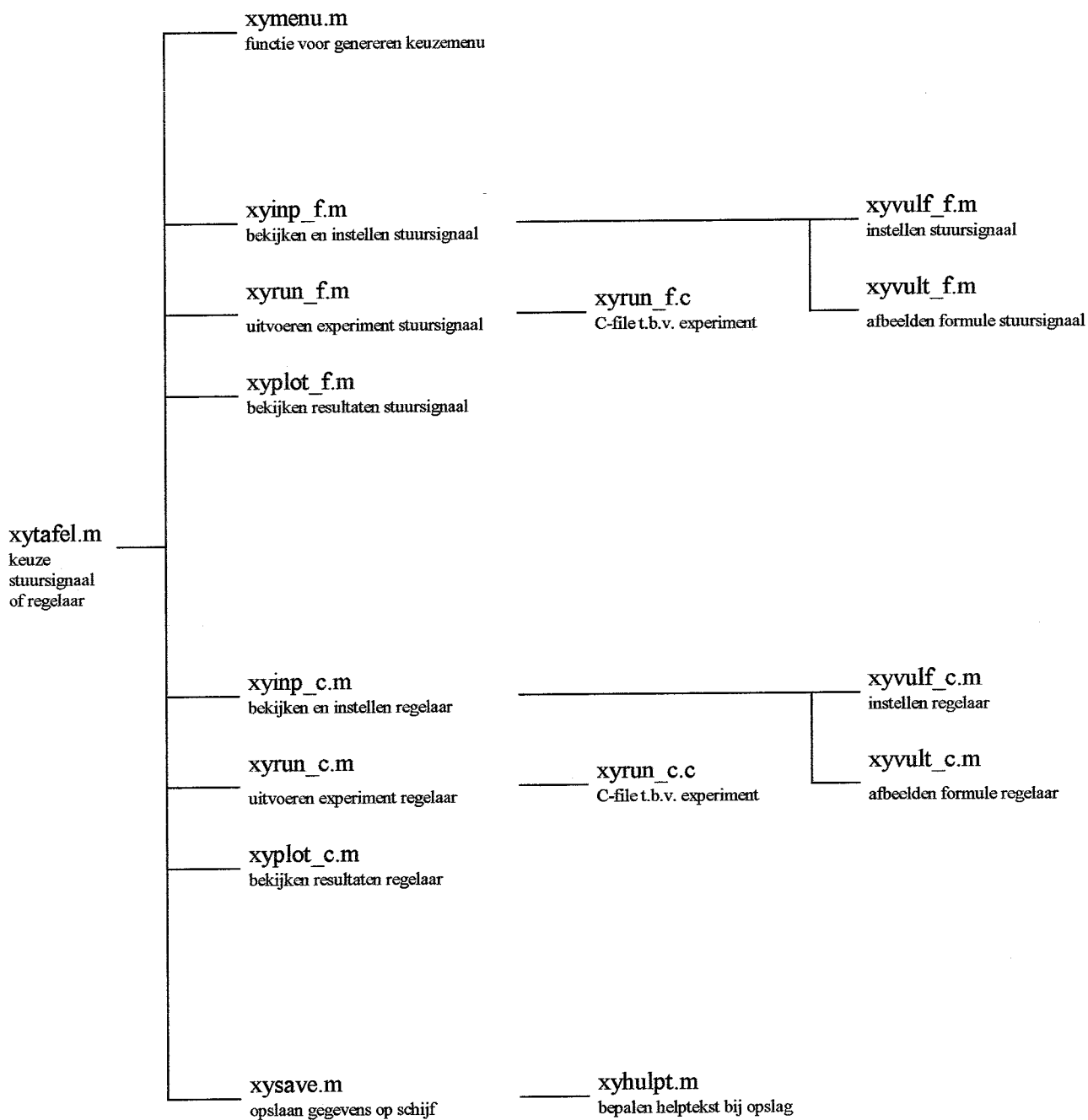
gehaald moeten kunnen worden. Tenslotte dienen zij de gekozen regelaar daadwerkelijk te beproeven op de xy-tafel in het laboratorium van WFW.

Het is niet de bedoeling dat de studenten geconfronteerd worden met allerlei bijzonderheden van de actuatoren en sensoren van de xy-tafel. Die zal zo geprepareerd worden dat zij bij de experimenten voor het bepalen van het model alleen de door hen gekozen stuurspanningen hoeven op te geven in MATLAB. De resultaten van de metingen worden opgeslagen en kunnen voor verdere bewerking (bijvoorbeeld weer met MATLAB eenvoudig naar een floppy geschreven worden. De implementatie van de gekozen regelaar zal voor de studenten niet meer inhouden dan het in een keuze-menu vastleggen van de gekozen versterkingsfactoren voor de P- en D-actie en eventueel voor de I-actie.

Bij de xy-tafel is informatie aanwezig over het gebruik ervan. Die zal niet aan iedere casus-groep worden uitgereikt.

Bijlage B

De programmastructuur met namen van de M- en C-files



Bijlage C

Alle files voor het programma xytafel.m (alfabetisch gerangschikt)

C1 xyhulpt.m

```
function[tekst]=xyhulpt(c);

% TEKST = XYHULPT(C);
%
% Hiermee wordt de hulptekst bepaald die wordt toegevoegd aan de opgeslagen
% file. In de hulptekst is beschreven welke parameters de file bevat.
%
% De functie xyhulpt wordt gebruikt in xysave.m van het programma xytafel.m

% hulptekst van stuursignaal
if c==1,

t = [
];

t1 = ['deze file bevat o.a. de volgende gegevens:
'];

t2 = t;
t3 = ['t : tijd van het experiment
'];
t4 = ['u : werkelijk aangebodeningangssignaal in x- en y-richting
'];
t5 = ['z : metingen posities in x-en y-richting
'];
t6 = ['te : eindtijd van het experiment
'];
t7 = ['sf : samplefrequentie
'];

t8 = ['ax en ay : de parameters van het stuursignaal volgens:
'];
t9 = ['U = a(1)*delta(t) + a(2)*eps(t) + a(3)*t + a(4)*cos(a(5)*t+a(6))'];

t10= ['Een snelheidsvector is uit z te bepalen met behulp van de
'];
t11= ['Matlab-functie "diff". Bijv. de snelheid in x-richting:
'];
t12= [' vx = [ 0 ; sf*diff(z(:,1)) ]
'];
t13= ['Voor meer uitleg over de functie "diff", type: help diff
'];

end;

% hulptekst van de regelaar
if c==2,

t = [
];

t1 = ['deze file bevat o.a. de volgende gegevens:
'];

t2 = ['t : tijd van het experiment
'];
t3 = ['u : berekende stuursignaal in x- en y-richting
'];
t4 = ['z : metingen posities in x-en y-richting
'];
t5 = ['zref : gewenste posities in x- en y-richting
'];
t6 = ['te : eindtijd van het experiment
'];
t7 = ['sf : samplefrequentie
'];

t8 = ['bx en by : de parameters van de regelaar volgens:
'];
t9 = ['G(s) = b(1) + b(2)*s + b(3)/s
'];

t10= ['Een snelheidsvector is uit z te bepalen met behulp van de'];
t11= ['Matlab-functie "diff". Bijv. de snelheid in x-richting:
'];
t12= [' vx = [ 0 ; sf*diff(z(:,1)) ]
'];
t13= ['Voor meer uitleg over de functie "diff", type: help diff'];

end;

tekst = [t1;
t;
t2; t3; t4; t5; t6; t7;
t;
t8; t9;
t;
t10; t11; t12; t13];

%---einde xyhulpt.m
```

C2 xyinp_c.m

```
% XYINP_C.M
%
% Hiermee worden de huidige instellingen van de regelparameters weer gegeven
% en kunnen de instellingen gewijzigd worden.
%
% Het programma wordt gebruikt in:
% * xytafel.m
% En maakt zelf gebruik van:
% * xyvult_c.m
% * xymenu.m
% * xyvulf_c.m

condir=3;

while condir>0,

%---Bepalen afbeelding formule regelaar in x-richting
tekst_cx = xyvult_c(bx);

%---Bepalen afbeelding formule regelaar in y-richting
tekst_cy = xyvult_c(by);

condir=xymenu('Regelaar wijzigen:',...
    ['Regelaar in x-richting, huidig: Gx(s) = 'tekst_cx],...
    ['Regelaar in y-richting, huidig: Gy(s) = 'tekst_cy],...
    ['Referentiesignaal in x-richting, huidig: Xref(t) = 'zrefx_tekst ' [m]],...
    ['Referentiesignaal in y-richting, huidig: Yref(t) = 'zrefy_tekst ' [m]]);
clc;
if isempty(condir),
    condir=3;
end;
if condir==1
    disp('---- Huidige instelling ----');
    disp(['    Gx(s) = 'tekst_cx]);
    bxold=bx;
    bx=xyvulf_c(bxold);
end;
if condir==2
    disp('---- Huidige instelling ----');
    disp(['    Gy(s) = 'tekst_cy]);
    byold=by;
    by=xyvulf_c(byold);
end;
if condir==3
    disp('---- Huidige instelling ----');
    disp(['    Xref(t) = 'zrefx_tekst ' [m]]);
    disp('');
    zrefx_old=zrefx_tekst;
    zrefx_tekst=input('Geef het referentiesignaal in x-richting: ','s');
    if isempty(zrefx_tekst)
        zrefx_tekst = zrefx_old;
    end;
    clc;
end;
if condir==4
    disp('---- Huidige instelling ----');
    disp(['    Yref(t) = 'zrefy_tekst ' [m]]);
    disp('');
    zrefy_old=zrefy_tekst;
    zrefy_tekst=input('Geef het referentiesignaal in y-richting: ','s');
    if isempty(zrefy_tekst)
        zrefy_tekst = zrefy_old;
    end;
    clc;
end;
end; % while condir>0

%---einde xyinp_c.m
```


C3 xyinp_f.m

```
% XYINP_F.M
%
% Hiermee worden de huidige instellingen van de stuursignalen weer gegeven
% en kunnen de instellingen gewijzigd worden.
%
% Het programma wordt gebruikt in:
% * xytafel.m
% En maakt zelf gebruik van:
% * xyvult_f.m
% * xymenu.m
% * xyvulf_f.m

fundir=3;

while fundir>0,

%---Bepalen afbeelding signaal op de ingang in x-richting
tekst_fx = xyvult_f(ax);

%---Bepalen afbeelding signaal op de ingang in y-richting
tekst_fy = xyvult_f(ay);

fundir=xymenu('Stuursignaal veranderen in:',...
              ['x-richting, huidig: Ux(t) = 'tekst_fx ' [N]],...
              ['y-richting, huidig: Uy(t) = 'tekst_fy ' [N]]);

clc;
if isempty(fundir),
    fundir=3;
end;
if fundir==1
    disp('----- Huidige instelling -----');
    disp(['    Ux(t) = 'tekst_fx ' [N]]);
    axold=ax;
    ax=xyvulf_f(axold);
end;
if fundir==2
    disp('----- Huidige instelling -----');
    disp(['    Uy(t) = 'tekst_fy ' [N]]);
    ayold=ay;
    ay=xyvulf_f(ayold);
end;

end; % while fundir>0

%---einde xyinp_f.m
```

C4 xymenu.m

```
function k = xymenu(s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15,s16)

%XYMENU Genereert een keuzemenu voor input van de gebruiker.
% K = XYMENU('Kies een kleur','Rood','Blauw','Groen') geeft op
% het scherm:
%
% ---- Kies een kleur ----
%
%      1) Rood
%      2) Blauw
%      3) Groen
%
%      0) Terug
%
%      Kies een nummer:
%
% Het door de gebruiker ingegeven nummer wordt terug gegeven.

disp(' ')
disp(['----- 's0, '-----'])
disp(' ')
for i=1:(nargin-1),
```

```

disp([' ',int2str(i),' ',eval(['s',int2str(i)])])
end;
disp("");
disp(' 0) Terug ');
disp("");
k = input('Kies een nummer: ');

%---einde xymenu.m

```

C5 xyplot_c.m

```

% XYPLOT_C.M
% Geeft op het scherm de gerealiseerde en gewenste uitgangen van de xy-tafel
% tegen de tijd. Ook wordt de fout hiertussen afgebeeld
% Dit programma wordt gebruikt in xytafel.m

e=z-zref;

clg;
subplot(211);
plot(t,zref(:,1),t,z(:,1)); title('gewenste en gerealiseerde positie in x-richting');
xlabel('[s]'); ylabel('[m]');grid;
plot(t,e(:,1)); title('fout in de x-richting');
xlabel('[s]'); ylabel('[m]');grid;
pause;

clg;
subplot(211);
plot(t,zref(:,2),t,z(:,2)); title('gewenste en gerealiseerde positie in y-richting');
xlabel('[s]'); ylabel('[m]');grid;
plot(t,e(:,2)); title('fout in de y-richting');
xlabel('[s]'); ylabel('[m]');grid;
clear e;
pause;

rref = sqrt(zref(:,1).^2+zref(:,2).^2);
r = sqrt(z(:,1).^2+z(:,2).^2);
er = r-rref;
clg;
subplot(211);
plot(t,rref,t,r); title('gewenste en gerealiseerde straal');
xlabel('[s]'); ylabel('[m]');grid;
clear rref;
clear r;
plot(t,er); title('fout in de straal');
xlabel('[s]'); ylabel('[m]');grid;
clear er;
pause;

vx = [0; sf*diff(z(:,1))];
vy = [0; sf*diff(z(:,2))];
v = sqrt(vx.^2+vy.^2);
clear vx;
clear vy;

vrefx = sf*diff(zref(:,1));
vrefy = sf*diff(zref(:,2));
vref = [0.25; sqrt(vrefx.^2+vrefy.^2)];
clear vrefx;
clear vrefy;

ev = v-vref;
clg;
subplot(211);
plot(t,vref,t,v); title('gewenste en gerealiseerde omloopsnelheid');
xlabel('[s]'); ylabel('[m/s]');grid;
clear vref;
clear v;
plot(t,ev); title('fout in de omloopsnelheid');
xlabel('[s]'); ylabel('[m/s]');grid;
clear ev;
pause;

%---einde xyplot_c.m

```

C6 xyplot_f.m

```
% XYPLOT_F.M
%
% Geeft op het scherm de in- en uitgangen van de xy-tafel tegen de tijd.
% Dit programma wordt gebruikt in xytafel.m
```

```
clg;
subplot(211);
```

```
plot(t,u(:,1)); title('stuursignaal in x-richting');
xlabel('[s]'); ylabel('[N]');grid;
```

```
plot(t,u(:,2)); title('stuursignaal in y-richting');
xlabel('[s]'); ylabel('[N]');grid;
```

```
pause;
```

```
clg;
subplot(211);
```

```
plot(t,z(:,1)); title('positie in x-richting');
xlabel('[s]'); ylabel('[m]');grid;
```

```
plot(t,[0:sf*diff(z(:,1))]); title('snelheid in x-richting');
xlabel('[s]'); ylabel('[m/s]');grid;
```

```
pause;
```

```
plot(t,z(:,2)); title('positie in y-richting');
xlabel('[s]'); ylabel('[m]');grid;
```

```
plot(t,[0:sf*diff(z(:,2))]); title('snelheid in y-richting');
xlabel('[s]'); ylabel('[m/s]');grid;
```

```
pause;
```

```
%---einde xyplot_f.m
```

C7 xyrun_c.c

```
/* xyrun_c.c
```

XY-table control program used in "xytafel.m".
Implements a PID-controller
Does not use type mat.

Uses model from file xy1.c.

File IN.MAT must contain Matlab created matrices:

```
(1*1) sf    % sf - sample frequency [Hz]
(1*1) te    % te - end time of experiment [s]

(1*3) bx    % bx - Px, Dx and Ix
(1*3) by    % by - Py, Dy and Iy

(m*2) zref  % reference trajectory data [m]; first column
              % for x1, second column for y; xyrun_c will fetch
              % next row at every sample time; any value
              % for m (data will be cyclic repeated)
```

After run of xyrun_c, file OUT.MAT will contain results from experiment
(n cycles done):

```
(1*3) bx    % copied from in.mat
(1*3) by    % copied from in.mat
```

```

        (n*2) zref % copied (and eventually cyclic repeated)
                % from in.mat

        (n*1) t    % time axis
        (n*nz) z  % collected measurements (x y)
        (n*2) u   % applied commands (ux uy)
        (1*3) wait % wait counts (min max average)
*/

#include <stdio.h>      /* Standard input/output functions */
#include <tce.h>        /* xy functions (and mat, ts, ml) */
#include <exp_xy.def>  /* definitions for XY-table */
#include <exp.h>

/* Global declarations */
/* In this implementation all the variables below MUST be */
/* declared global. */

#define nz 2
# define nmax 1000

int n, step;          /* Number of actions to do */
double sf, dt, te;    /* Sample frequency [Hz] */
double bx[3];         /* From IN.MAT */
double by[3];         /* From IN.MAT */
double z[nz];         /* Measurements */
double u[2];          /* Commands */
double P[2],          /* Proportional gains */
       D[2],          /* Differential gains */
       I[2];          /* Integral gains */
double t[1];          /* Actual time */
double zref[2];       /* Reference positions [m] */
double zold[2];       /* Old positions for speed calc */
double wait[3] = { 0, 0, 0 }; /* Wait counts */

void epilog(void) {
/*
    Sends registrations to Matlab
*/
    ml_open("out",1);
    ml_put_vec(bx,3,"bx");
    ml_put_vec(by,3,"by");
    ml_put_vec(wait,3,"wait");
    ts_save_all(0);
    ml_close();
}

void prolog(void){
/*
    Opens MATLAB-file and initializes ts
*/
    /* Get data from Matlab */
    ml_open("in",0);
    sf = ml_get_scalar("sf",100);
    te = ml_get_scalar("te",10);
    ml_get_vec(bx,3,"bx");
    ml_get_vec(by,3,"by");
    ts_load0(zref,2,"zref");
    ml_close();

    /* Establish global variables */
    dt = 1.0/sf;
    n = (int)(te*sf); /* # samples to go */
    step = (int)(n/nmax); /* steps for registration */
    if (step==0)
        step=1;

    /* Establish gains P, D and I */
    P[0]=bx[0] ; D[0]=bx[1] ; I[0]=bx[2] ;
    P[1]=by[0] ; D[1]=by[1] ; I[1]=by[2] ;

    ts_init0(z,nz,n,"z"); /* init ts registrations */
    ts_init0(u,2,n,"u");
    ts_init0(zref,2,n,"zref");
    ts_init0(t,1,n,"t");
}

```

```

void control(void) {
/*
    Control law.
    Implements a (double) PID regulator. No effort is made to
    avoid static errors (due to Coulomb friction).
    Very straight-forward speed calculation which may cause
    troubles with a high speed feedback gain or very short
    sample time.
*/
    double zint[2];      /* integrated positions */
    double v[2];        /* speeds [m/s] */

    zint[0] = 0.5*dt*(z[0]+zold[0]);      /* x */
    zint[1] = 0.5*dt*(z[1]+zold[1]);      /* y */

    v[0] = (z[0]-zold[0])/dt; zold[0] = z[0];      /* x */
    v[1] = (z[1]-zold[1])/dt; zold[1] = z[1];      /* y */

    u[0] = P[0]*(zref[0]-z[0]) - D[0]*v[0] - I[0]*zint[0];
    u[1] = P[1]*(zref[1]-z[1]) - D[1]*v[1] - I[1]*zint[1];
}

void run(void) {
/*
    Runs the experiment.
*/
    int i;

    ts_get_all();

    exp_init(z,nz,u,dt,wait);
    t[0]=0; zold[0]=z[0]; zold[1]=z[1];

    /* There we go ... */
    for ( i=0; i<(n-1); i++ ) {
        exp_get();      /* get measurements z */
        if ((i%step)==0)
            ts_put_all(); /* put time histories */
        ts_get_all(); /* fetch next reference data */
        control(); /* compute commands */
        exp_put(); /* put commands u */
        t[0] += dt;
    }

    exp_fini(); /* Finish xy_operations */
                /* MUST BE DONE NOW !!! */
}

int main(void) {

    prolog();
    run();
    epilog();

    return 0;
}

/* end of xyrun_c.c */

```

C8 xyrun_c.m

```
% XYRUN_C.M
%
% De regelparameters worden aangeboden aan de xytafel, in de file out.mat
% worden alle meetgegevens opgeslagen.
% Deze file wordt weer ingeladen in Matlab
%
% xyrun_c.m wordt gebruikt in xytafel.m.

BX=[BX; bx'];
BY=[BY; by'];
save test_c BX BY;

if sf*te > 2000;
    te = 2000/sf;
end;

t=0:(1/sf):te;

zref = [(eval(zrefx_tekst))' (eval(zrefy_tekst'))];

%---Opslaan belangrijke gegevens voor xy-tafel
save in bx by te sf zref;

%---Aanbieden gekozen regelparameters aan xy-tafel (echt of simulatie)
save alles;
save fn;
clear;
load fn;
if exp==1,
    ! xyrun_c    % echt experiment
else
    ! xyrun_cs   % simulatie
end;

%---Meetgegevens weer inladen in Matlab
load alles;
load out

%---einde xyrun_c.m
```

C9 xyrun_f.c

```
/* xyrun_f.c
```

Program used in "xytafel.m" to obtain the responses of the xy-table,
using known commands uc given by Matlab-file
Uses model from file sim_ex.c

1. Reads from MATLAB Data file IN.MAT:

name	size	meaning
sf	1*1	sample frequency [Hz]
te	1*1	endtime of experiment [s]
ax & ay	6*1	command values for actuator(s)

2. Runs the experiment during n cycles using sample frequency
sf.

3. Creates MATLAB Result file OUT.MAT:

name	size	meaning
t	n*1	time base
z	n*nz	measurements
u	n*nu	commands as applied
wait	3*1	wait count info vector

Remark: nz is defined for the specific
experiment.

Remark: when running a real experiment, wait will hold

```

        minimum wait count
        maximum wait count
        average wait count
    in function exp_get; see exp.doc for details;
    when running a simulator, wait will hold
        Total clock time used
        # integrator steps with progression
        # integrator steps without progression
    see sim.doc for details.
*/

#include <stdio.h>
#include <tce.h>
#include <exp_xy.def>
#include <exp.h>
#include <math.h>

# define nu 2
# define nz 2
# define nmax 1000

int  n, step;          /* Number of actions to do          */
double sf, te;        /* From IN.MAT                      */
double t [1],        /* Actual time                       */
        u [nu],      /* Commands (as applied)            */
        z [nz],      /* Measurements                      */
        wait[3],     /* Wait counts                       */
        zold[2],     /* Old positions for speed calculation */
        ax[6],       /* Parameters for steering           */
        ay[6];

void  epilog(void) {
    ml_open("out",1);
    ts_save_all(1);
    ml_put_vec(wait,3,"wait");
    ml_close();
}

void  prolog(void) {
    ml_open("in",0);
    sf = ml_get_scalar("sf",100);
    te = ml_get_scalar("te",10);
    ml_get_vec(ax,6,"ax");
    ml_get_vec(ay,6,"ay");
    ml_close();

    n = (int)(te*sf);
    ts_init0(t, 1,n,"t"); /* Initialize time series for t */
    ts_init0(z,nz,n,"z"); /* Initialize time series for z */
    ts_init0(u,nu,n,"u"); /* Initialize time series for u */
}

void control(void)
{
    double a1[2], upuls[2], umax[2];
    int i;
    umax[0]=246;
    umax[1]=41;
    for(i=0 ; i<2 ; i++)
    {
        a1[0] = ax[0];
        a1[1] = ay[0];
        if (t[0] < (a1[i]/umax[i]))
            upuls[i] = umax[i];
        else
            upuls[i]=0;
    }
    u[0]= ax[0]*upuls[0] + ax[1] + ax[2]*t[0] + ax[3]*cos(ax[4]*t[0]+ax[5]);
    u[1]= ay[0]*upuls[1] + ay[1] + ay[2]*t[0] + ay[3]*cos(ay[4]*t[0]+ay[5]);
}

```

```

#define error(msg) { exp_fini(); epilog(); tce_error(msg); }

void watch()
{
    double v[2];          /* speeds [m/s] */
    v[0]=(z[0]-zold[0])*sf; zold[0] = z[0];
    v[1]=(z[1]-zold[1])*sf; zold[1] = z[1];

    if ( (z[1]>0.2) && (v[1]>1.0) )
        error("Not allowed position-speed combination #1");

    if ( (z[1]<-0.2) && (v[1]<-1.0) )
        error("Not allowed position-speed combination #2");
}

void run(void) {
/*
    Timing for data in the time series, in MATLAB notation, is:
    - z(k,:) contains the measurements at time t = (k-1)*dT, where
      dT = 1/sf, so the first sample is at time t = 0.
    - u(k,:) are the commands issued at time t = (k-1)*dT; with
      Zero Order Hold characteristic they apply between time
      t = (k-1)*dT and k*dT. So, measurements z(k,:) do not
      depend on commands u(k,:).
*/
    double dt;
    int i;

    dt = 1.0/sf;
    t[0] = 0;
    step = int(n/nmax);
    if (step == 0)
        step=1;
    exp_init(z,nz,u,dt,wait);

    zold[0] = z[0];          /* initial values zold*/
    zold[1] = z[1];

    /* There we go.... */
    for (i=0; i<(n-1); i++)
    {
        ts_get_all();
        control();
        exp_put();
        if ((i%step)==0)
            ts_put_all();

        exp_get();
        watch();
        t[0] += dt;
    }
    exp_fini();
}

void main(void) {
    prolog();
    run();
    epilog();
}

/* end of xyrun_fc */

```


C10 xyrun_f.m

```
% XYRUN_F.M
%
% Hiermee wordt het werkelijke stuursignaal berekend.
% Deze signalen worden aangeboden aan de xytafel, in de file out.mat worden
% alle meetgegevens opgeslagen. Deze file wordt weer ingeladen in Matlab
%
% xyrun_f.m wordt gebruikt in xytafel.m.

AX=[AX; ax']; % dit is om te kijken wat er met het programma wordt gedaan
AY=[AY; ay'];
save test_f AX AY;

if sf*te > 8000 % anders wordt de vector te lang voor matlab
    te = 8000/sf;
end;

%---Opslaan belangrijke gegevens voor xy-tafel
save in ax ay te sf;

%---Aanbieden gekozen stuursignaal aan xy-tafel (echt of simulatie)
save alles;
save fn exp;
clear;
load fn;
if exp==1,
    ! xyrun_f % echt experiment
else
    ! xyrun_fs % simulatie
end;

%---Meetgegevens weer inladen in Matlab
load alles;
load out

%---einde xyrun_f.m
```

C11 xysave.m

```
% XYSAVE.M
%
% Vraagt naar de filename waaronder de gegevens moeten worden bewaard.
% In deze file komen te staan:
% * ingangssignalen in x- en y-richting; ux en uy
% * uitgangssignalen in x- en y-richting; outx en outy
% * de tijd; t
% * de samplefrequentie; sf
% Het programma xysave.m wordt gebruikt in xytafel.m

% ---Geef de filename
filename = input('Geef de filename: ', 's');

% ---Haal lege plaatsen weg
space = find(filename == ' ');
if isempty(space) == 0,
    filename = filename(1:space(1)-1);
end;
clear space;

% ---Bepaal de extensie als deze nog niet ingegeven is
if isempty(find(filename == '.')),
    filename = [filename, '.mat'];
end;

% Zorg dat file op a-drive wordt opgeslagen als geen drive is gespecificeerd
if isempty(find(filename == '\')),
    filename = ['a:\' filename];
end;

% Aanmaken van de hulptekst die aan de file wordt toegevoegd
```

```

hulp=xyhulpt(begin);

% ---Kijken of filename al bestaat en bewaren onder ingegeven naam
if exist(filename) ==2,
    err = 1;
    disp("");
    disp(' PAS OP: deze filename bestaat al!!!');
else
    err = 0;
    disp("");
    disp([' ==> de gegevens worden bewaard als: ' filename]);
    save alles;
    save fn filename hulp;
    clear;
    load fn;
    load in;
    load out ;
    eval(['save ', filename]);
    load alles;
end;

if err == 1,
    if input(' ==> oude file vervangen? (j/n) ','s') == 'j',
        err = 0;
        disp("");
        disp([' ==> de gegevens worden bewaard als: ' filename]);
        save alles;
        save fn filename hulp;
        clear;
        load fn;
        load in;
        load out ;
        eval(['save ', filename]);
        load alles;
    end;
end;

disp("");
disp(' >>> druk op een willekeurige toets om verder te gaan <<<');
pause;

clc;
%--einde xysave.m

```

C12 xytafel.m

```

% XYTAFEL.M
%
% Programma voor het aansturen van de XY-tafel.
% In dit programma kunnen twee zaken gekozen worden:
% * Het aanbieden van een zelf gekozen stuursignaal aan de beide motoren
% * Het aanbieden van een regelaar om een bepaald trajectorie te doorlopen
%
% Het programma xytafel.m maakt gebruik van de volgende files:
% * xymenu.m
% * xy*_f.m
% * xy*_c.m
% * xysave.m

%---Initialiseren van het programma

clear;
clc;
clg;

load test_f;
load test_c;

a = [ 0.0; 0.0; 0.0; 0.0; 0.0; 0.0 ];
b = [ 0.0; 0.0; 0.0 ];
ax = a; ay = a;
bx = b; by = b;
te = 2.0;
sf = 100;

```

```

zrefx_tekst = '0';
zrefy_tekst = '0';
begin = 3;
exp = 0; % als exp=1 dan echt experiment, anders simulatie

expe = input('uitvoeren experiment (e) of simulatie (s)? ', 's');
if expe == 'e',
    exp = 1;
else
    exp = 0;
end;
clc;

%---Zolang er niet wordt gekozen voor het beëindigen van het programma
%---is er de keuze uit de twee mogelijkheden

while begin>0

begin=xymenu('Begin','Aanbieden Stuursignaal','Aanbieden Regelaar');

if isempty(begin)
    begin = 3;
end;

clc;

if begin == 1
func = 6;

%---Zolang er niet gekozen wordt om dit deel van het programma te beëindigen,
%---is er de keuze uit 5 mogelijkheden

while func > 0,

func = xymenu('Stuursignaal',...
    'Bekijken en instellen parameters',...
    'Instellen eindtijd experiment',...
    'Uitvoeren experiment',...
    'Bekijken resultaten',...
    'Opslaan gegevens op schijf');

if isempty(func),
    func = 6;
end;

clc;
if func==1,
    xyinp_f;
end;
if func==2,
    told=te;
    te=input(['Geef de eindtijd < num2str(te) ' [s]> : ']);
    if isempty(te),
        te=told;
    end;
    clc;
end;
if func==3,
    xyrun_f;
end;
if func==4,
    xyplot_f;
end;
if func==5,
    xysave;
end;
end; %while func>0
end;

if begin==2
contr=6;

%---Zolang er niet gekozen wordt om dit deel van het programma te beëindigen,
%---is er de keuze uit 5 mogelijkheden

while contr>0,

```

```

contr = xymenu('Regelaar',....
    'Bekijken en instellen parameters',....
    'Instellen eindtijd experiment',....
    'Uitvoeren experiment',....
    'Bekijken resultaten',....
    'Opslaan gegevens op schijf');
if isempty(contr),
    contr=6;
end;
clc;

if contr==1,
    xyinp_c;
end;
if contr==2,
    told=te;
    te=input(['Geef de eindtijd < num2str(te) '[s]> : ']);
    if isempty(te),
        te=told;
    end;
    clc;
end;
if contr==3,
    xyrun_c;
end;
if contr==4,
    xyplot_c;
end;
if contr==5,
    xysave;
end;

end; %while contr>0;
end;

if begin==0
    uit=input('Programma xytafel verlaten? (j/n): ', 's');
    if uit == 'j'; begin = 0;
    else if uit == 'n'; begin = 0;
        else begin=3;
            end;
    end;
end;
end;
end
%---einde xytafel.m

```

C13 xyvulf_c.m

```

function[b]=xyvulf_c(bold);

% B = XYVULF_C(BOLD)
%
% Hiermee wordt de vector B bepaald, deze bestaat uit de parameters die de
% regelaar in x- of y-richting bepalen. De oude regelaarparameters worden
% gegeven als default waarde
%
% De regelaar ziet er uiteindelijk als volgt uit:
%  $G(s) = b(1) + b(2)*s + b(3)/s$ 
%
% Het programma maakt gebruik van xymenu.m,
% en wordt aangeroepen door xyinp_c.m.

b1=0; b2=0; b3=0;

%---Er kan gekozen worden uit 3 typen regelaars

contyp=xymenu('Type Regelaar','P','PD','PID');

if contyp==1,
    disp("");
    disp('P-regelaar:  $G(s) = Kp$ ');
    b1=input(['geef de waarde van Kp < num2str(bold(1)) '> : ']);
end;

```

```

if contyp==2
    disp("");
    disp('PD-regelaar: G(s) = Kp + Kd*s');
    b1=input(['geef de waarde van Kp < num2str(bold(1)) > : ']);
    b2=input(['geef de waarde van Kd < num2str(bold(2)) > : ']);
end;

if contyp==3
    disp("");
    disp('PID-regelaar: G(s) = Kp + Kd*s + Ki/s');
    b1=input(['geef de waarde van Kp < num2str(bold(1)) > : ']);
    b2=input(['geef de waarde van Kd < num2str(bold(2)) > : ']);
    b3=input(['geef de waarde van Ki < num2str(bold(3)) > : ']);
end;

%---Als op de enter-toets wordt gedrukt dan wordt de bijbehorende
%---variabele op nul gesteld:

if isempty(b1), b1=bold(1); end;
if isempty(b2), b2=bold(2); end;
if isempty(b3), b3=bold(3); end;

b=[b1; b2; b3];

clc;

%---einde xyvulf_c.m

```

C14 xyvulf_f.m

```

function[a]=xyvulf_f(aold);

% A = XYVULF_F(AOLD)
%
% Hiermee wordt de vector A bepaald, deze bestaat uit de parameters die het
% stuursignaal in x- of y-richting bepalen. Het oude signaal wordt
% gegeven als default waarde
%
% Het stuursignaal ziet er uiteindelijk als volgt uit:
% u = a(1)*d(t) + a(2)*eps(t) + a(3)*t + a(4)*cos(a(5)*t+a(6))
%
% Het programma maakt gebruik van xymenu.m,
% en wordt aangeroepen door xyinp_f.m.

a1=0; a2=0; a3=0; a4=0; a5=0; a6=0;

%---Er kan gekozen worden uit 4 typen stuursignalen en een combinatie daarvan

funtyp=xymenu('Type Stuursignaal','Puls','Stap','Lineair',...
    'Harmonisch','Combinatie bovenstaande signalen');

if funtyp==1,
    disp("");
    disp('Puls-sigitaal: a*delta(t)');
    disp("");
    disp('delta(t) is een puls met oppervlakte 1, de hoogte is de maximale kracht');
    disp('die door de motor kan worden geleverd ( x-richting: 246[N], y-richting:41 [N] ) ');
    disp('a is dus het aantal keer dat deze puls achter elkaar wordt gezet');
    disp("");
    a1=input(['geef de waarde van a < num2str(aold(1)) > : ']);
end;

if funtyp==2
    disp("");
    disp('Stap-sigitaal: a*eps(t)');
    disp("");
    disp('a is het aantal Newton dat er op het tijdstip t=0 wordt opgezet');
    disp("");
    a2=input(['geef de waarde van a < num2str(aold(2)) > : ']);
end;

```

```

if funtyp==3
    disp("");
    disp('Lineair signaal: a*t');
    disp("");
    a3=input(['geef de waarde van a < num2str(aold(3)) > : ']);
end;

if funtyp==4
    disp("");
    disp('Harmonisch signaal: a*cos(w*t+phi)');
    a4=input(['geef de waarde van a < num2str(aold(4)) > : ']);
    a5=input(['geef de waarde van w < num2str(aold(5)) > : ']);
    a6=input(['geef de waarde van phi < num2str(aold(6)) > : ']);
end;

if funtyp==5
    disp("");
    disp('Gecombineerd signaal:');
    disp('a1*delta(t) + a2*eps(t) + a3*t + a4*cos(a5*t + a6)');
    a1=input(['geef de waarde van a1 < num2str(aold(1)) > : ']);
    a2=input(['geef de waarde van a2 < num2str(aold(2)) > : ']);
    a3=input(['geef de waarde van a3 < num2str(aold(3)) > : ']);
    a4=input(['geef de waarde van a4 < num2str(aold(4)) > : ']);
    a5=input(['geef de waarde van a5 < num2str(aold(5)) > : ']);
    a6=input(['geef de waarde van a6 < num2str(aold(6)) > : ']);
end;

%--Als op de enter-toets wordt gedrukt dan wordt de bijbehorende
%--variabele op de oude waarde gesteld:

if isempty(a1), a1=aold(1); end;
if isempty(a2), a2=aold(2); end;
if isempty(a3), a3=aold(3); end;
if isempty(a4), a4=aold(4); end;
if isempty(a5), a5=aold(5); end;
if isempty(a6), a6=aold(6); end;

a=[a1; a2; a3; a4; a5; a6];

clc;

%--einde xyvult_f.m

```

C15 xyvult_c.m

```

function[tekst]=xyvult_c(b)

% TEKST = XYVULT_C(B)
%
% Met xyvult_c.m wordt in xyinputc.m de ingevoerde formule als tekst
% geschreven, zodat deze afgebeeld kan worden op het scherm.
% Als input van deze functie gelden de parameters B van hetingangssignaal.
% De functie geeft een string terug.
%
% De string heeft de volgende vorm:
% b1 + b2*s + b3/s
%
% Als een parameter de waarde nul heeft dan wordt dat deel van de
% vergelijking niet afgebeeld.
%
% De functie wordt gebruikt in xyinp_c.m

% --Bepalen of de parameter wel of niet wordt afgebeeld:

if b(1)==0, b1=[];
else b1=num2str(b(1));
end;

if b(2)==0, b2=[]; s=[];
else b2=[' + ' num2str(b(2))]; s='*s';
end;

if b(3)==0, b3=[]; ss=[];

```

```

else      b3=[' + num2str(b(3))]; ss='/'s';
end;

if b(1)==0 & b(2)==0 & b(3)==0 s='0'; end;

% ---Het maken van de string: b1 + b2*s + b3/s
tekst = [b1 b2 s b3 ss];

% ---einde xyvult_c.m

```

C16 xyvult_f.m

```

function[tekst]=xyvult_f(a)

% TEKST = XYVULT_F(A)
%
% Met xyvult_f.m wordt in xyinput.m de ingevoerde formule als tekst
% geschreven, zodat deze afgebeeld kan worden op het scherm.
% Als input van deze functie gelden de parameters A van het stuursignaal.
% De functie geeft een string terug.
%
% De string heeft de volgende vorm:
% a1*delta(t) + a2*eps(t) + a3*t + a4*cos(a5*t+a6)
%
% Als een parameter de waarde nul heeft dan wordt dat deel van de
% vergelijking niet afgebeeld.
%
% De functie wordt gebruikt in xyinp_f.m

% ---Bepalen of de parameter wel of niet wordt afgebeeld:

if a(1)==0, a1=[]; dt=[];
else      a1=num2str(a(1)); dt='*delta(t)';
end;

if a(2)==0, a2=[]; eps=[];
else      a2=[' + ' num2str(a(2))]; eps='*eps(t)';
end;

if a(3)==0, a3=[]; tt=[];
else      a3=[' + ' num2str(a(3))]; tt='*t';
end;

if a(4)==0, a4=[]; cos1=[]; cos2=[];
else      a4=[' + ' num2str(a(4))]; cos1='*cos('; cos2=')';
end;

if a(5)==0, a5=[]; wt=[];
else if a(4)~=0, a5=num2str(a(5)); wt='*t';end;
end;

if a(6)==0, a6=[];
else if a(4)~=0, a6=[' + ' num2str(a(6))]; end;
end;

if a(1)==0 & a(2)==0 & a(3)==0 & a(4)==0, dt='0'; end;

% ---Het maken van de string:
% ---a1*delta(t) + a2*eps(t) + a3*t + a4*cos(a5*t+a6)

tekst = [a1 dt a2 eps a3 tt a4 cos1 a5 wt a6 cos2];

% ---einde xyvult_f.m

```

Bijlage D

Overzicht resultaten van gebruikers

D1 Resultaten identificatie systeem

groep	M_x [kg]	F_{wx} [N]	M_v [kg]	F_{wv} [N]
onderzoek 1	60	40	2.9	6.5
onderzoek 2	40	30	1.11	9
onderzoek 3	16.7	50	3.57	8.57
WMT 1	48.148	35	3.81	5
WMT 2	59	45	2.97	5.3
\bar{x}	45	40	2.9	6.9
σ	18	7.9	1.1	1.8

D2 Resultaten instellingen PD-regelaar

groep	P_x	D_x	P_v	D_v
onderzoek 1	10000	200	3000	150
onderzoek 2	40000	300	30000	1000
onderzoek 3	-	-	-	-
WMT 1	-	-	-	-
WMT 2	8000	600	2000	150

D3 Resultaten instellingen PID-regelaar

groep	P_x	I_x	D_x	P_v	I_v	D_v
onderzoek 1	10000	90	500	3000	-	150
onderzoek 2	-	-	-	-	-	-
onderzoek 3	25000	50000	650	4500	5000	150
WMT 1	10000	10000	600	2000	2000	150
WMT 2	8000	125	600	2000	-	150