

Swingup en stabilisatie van een geïnverteerde roterende slinger

Verslag FIT-stage

Sietze van Buuren

Begeleider:
Dr. H. Hasper

30 mei 2006

Samenvatting

Van een geïnverteerde roterende slinger is met behulp van regeltechniek en routine ontwikkeld, waarmee de slinger vanuit hangende positie in staande positie kan worden gebracht. Tevens stabiliseert deze routine de slinger in staande positie, zodat deze in staande positie blijft. Dit systeem is gesimuleerd in simulink met behulp van SimMechanics en gevisualiseerd met behulp van de virtual reality toolbox.

Inhoudsopgave

1	Inleiding	3
2	Theorie	4
2.1	De Furuta Pendulum	4
2.2	Lineairisatie	5
2.3	Slingertraject	6
3	Simulaties	7
3.1	Matlab	7
3.2	Simulink	9
4	Resultaten en discussie	13
5	Conclusie	15
6	Dankwoord	15
7	Appendices	16
A	Matlab	16
B	Opstelling	18

1 Inleiding

De circulaire geïnverteerde pendulum wordt ook wel aangeduid met de Furuta pendulum, vernoemd naar de Japanse wetenschapper K. Furuta. Deze pendulum bestaat uit een arm met zijn draaipunt in het midden. Aan één kant van deze arm is een slinger bevestigd, die roteert in een vlak wiens normaal samenvalt met de arm. Initieel bevindt de slinger zich in hangende positie. Het doel van het experiment is de slinger vanuit hangende positie in staande positie te brengen door middel van het bewegen van de arm en hierna de slinger in deze positie te houden.

Om de slinger omhoog te slingeren zijn verschillende strategieën mogelijk. In dit geval is gekozen voor een slingertraject dat wordt bepaald door de energie van de slinger.

Een regelsysteem voor de slinger in staande positie met incrementele uitwijkingen valt vrij gemakkelijk af te leiden door het systeem te lineairiseren m.b.v. bijvoorbeeld Jacobianen. Door dit systeem vervolgens te analyseren met Matlab kan een regelsysteem worden verkregen voor controle van de slinger in staande positie.

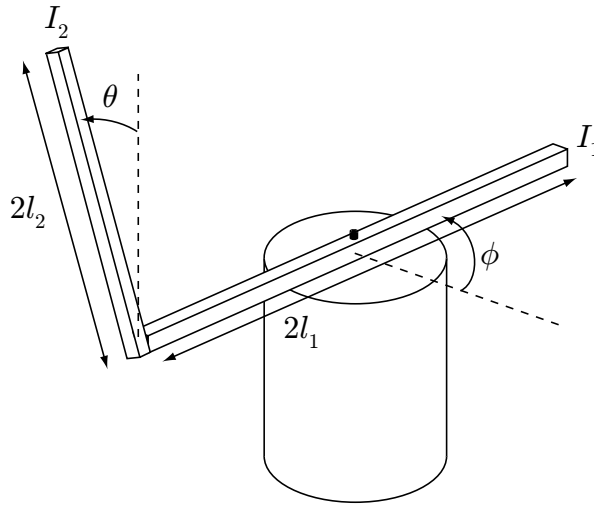
Beide bovenstaande zaken zijn gecombineerd in een Simulink model, waarbij het systeem zelf wordt gesimuleerd met behulp van SimMechanics. Dit is vervolgens nog eens gevisualiseerd met behulp van de Virtual Reality toolbox van Matlab.

Aan het begin van het project was het de bedoeling dat er ook een opstelling zou worden ontworpen en gebouwd. Dit is laatste is echter nooit gebeurd, omdat dit te kostbaar bleek. De tekening van de opstelling die is ontworpen is terug te vinden in appendix B.

2 Theorie

2.1 De Furuta Pendulum

De Furuta pendulum bestaat uit een arm (met lengte $2l_1$) die geroteerd is onder een hoek ϕ . Deze arm is bevestigd aan een motor die het systeem aandrijft. De arm en motor hebben samen een traagheidsmoment I_1 en frictiecoëfficiënt van de motor is d_1 . De slinger (van lengte $2l_2$, traagheidsmoment I_2 en massa m) is bevestigd aan deze arm en is geroteerd onder een hoek θ . De frictiecoëfficiënt van de slinger is d_2 .



Figuur 1: Schematische tekening van een Furuta pendulum.

Met behulp van Langragianen kunnen de bewegingsvergelijkingen voor de dit systeem worden bepaald. De kinetische energie van het systeem is

$$T = \frac{1}{2}(I_2 + ml_2^2)\dot{\theta}^2 + \frac{1}{2}(I_1 + ml_1^2 + (I_2 + ml_2^2) \sin^2 \theta)\dot{\phi}^2 - \dot{\theta}\dot{\phi}ml_1l_2 \cos \theta \quad (1)$$

De potentiële energie van de slinger bedraagt

$$V = mgl_2 \cos \theta \quad (2)$$

Hiermee wordt de Langrangiaan

$$L = T - V = \frac{1}{2}(I_2 + ml_2^2)\dot{\theta}^2 + \frac{1}{2}(I_1 + ml_1^2 + (I_2 + ml_2^2) \sin^2 \theta)\dot{\phi}^2 - \dot{\theta}\dot{\phi}ml_1l_2 \cos \theta - mgl_2 \cos \theta \quad (3)$$

Gebruik makend van de Lagrange vergelijking resulteert dit in de twee bewegingsvergelijkingen:

$$\ddot{\phi}(I_1 + ml_1^2 + (I_2 + ml_2^2) \sin^2 \theta) + 2\dot{\theta}\dot{\phi}(I_2 + ml_2^2) \sin \theta \cos \theta - \ddot{\theta}l_1l_2 \cos \theta + \dot{\theta}^2ml_1l_2 \sin \theta = u - d_1\dot{\phi} \quad (4)$$

$$\ddot{\theta}(I_2 + ml_2^2) - \ddot{\phi}ml_1l_2 \cos \theta - \dot{\phi}^2(I_2 + ml_2^2) \sin \theta \cos \theta - mgl_2 \sin \theta = -d_2\dot{\theta} \quad (5)$$

2.2 Lineairisatie

Om vergelijking (4) en (5) te kunnen lineairiseren, moeten deze eerst worden omgeschreven naar de vorm

$$\frac{dx}{dt} = f(x) + g(x)u = \begin{bmatrix} x_3 \\ x_4 \\ f_3(x) \\ f_4(x) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g_3(x) \\ g_4(x) \end{bmatrix} u \quad (6)$$

hierbij is $x := [\phi, \theta, \dot{\phi}, \dot{\theta}]^T$ en u het koppel van de motor. Verder staan $f_3(x)$, $f_4(x)$, $g_3(x)$ en $g_4(x)$ voor

$$f_3(x) = [k_2(-d_1x_3 - 2x_3x_4 \sin x_2 \cos x_2 - k_3x_4^2 \sin x_2) + k_3(-d_2x_4 + k_2x_3^2 \sin x_2 \cos x_2 + mgl_2 \sin x_2) \cos x_2]/K \quad (7)$$

$$f_4(x) = [(k_1 + k_2 \sin^2 x_2)(-d_1x_4 + k_2x_3^2 \sin x_2 \cos x_2 mgl_2 \sin x_2) + k_3(-d_1x_3 - 2x_3x_4 \sin x_2 \cos x_2 - k_3x_4^2 \sin x_2) \cos x_2]/K \quad (8)$$

$$g_3(x) = \frac{k_2}{K} \quad (9)$$

$$g_4(x) = \frac{k_3 \cos x_2}{K} \quad (10)$$

waarbij $k_1 = I_1 + ml_1^2$, $k_2 = I_2 + ml_2^2$, $k_3 = ml_1l_2$ en $K = k_2(k_1 + k_2 \sin^2 x_2) - k_3^2 \cos^2 x_2$.

Bovenstaande vergelijkingen kunnen worden gelineairiseerd met behulp van de methode die is beschreven in [4]. Er zal een korte uitleg worden gegeven van deze lineairisatie, voor meer details worden verwezen naar de genoemde referentie.

In deze lineairsatiemethode wordt gebruikt gemaakt van een benadering van de Taylor ontwikkeling op een vergelijking in de vorm van

$$\dot{x}_i = f_i(\mathbf{x}, \mathbf{u}) \quad (11)$$

In dit geval is dit vergelijking (6). De methode is afhankelijk van een zogenaamd operation point (x_{i_0}, u_{i_0}) . Welke wordt toegepast op de uiteindelijke notatie

$$\dot{x}_i^* = \dot{x}_i - \dot{x}_{i_0} \quad x_i^* = x_i - x_{i_0} \quad u_i^* = u_i - u_{i_0} \quad (12)$$

Voor ons systeem geldt echter dat deze operation points 0 zijn; $x_{i_0} = u_{i_0} = 0$. Met de bovenstaande uitdrukkingen wordt (11) gelineairiseerd tot

$$\dot{\mathbf{x}}^* = \mathbf{J}_x \mathbf{x}^* + \mathbf{J}_u \mathbf{u}^* \quad (13)$$

In de Jacobianen \mathbf{J}_x en \mathbf{J}_u komen de Taylor-approxomaties tevoorschijn:

$$\mathbf{J}_x = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad \mathbf{J}_u = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix} \quad (14)$$

Voor het geval dat hier wordt behandeld is $f_1 = f(x)$ en $f_2 = g(x)$. Wordt bovenstaande theorie toegepast op vergelijking (6), dan wordt de volgende state space vergelijking verkregen:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{gm^2l_1l_2}{D} & -\frac{d_1(ml_2^2+I_2)}{D} & -\frac{md_1l_1l_2}{D} \\ 0 & \frac{gml_2(ml_1^2+I_1)}{D} & -\frac{md_1l_1l_2}{D} & -\frac{d_2(ml_1^2I_1)}{D} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ \frac{ml_2^2+I_2}{D} \\ \frac{ml_1l_2}{D} \end{bmatrix} \mathbf{u} \quad (15)$$

Hierbij is $D = ml_2^2I_1 + (ml_1^2 + I_1)I_2$.

Met behulp van de LQR-functie van Matlab kan aan de hand van de bovenstaande state space vergelijking nu een feedback parameter worden bepaald waarmee het systeem voor incrementele veranderingen kan wordt gestabiliseerd.

Dit zal worden gedaan in paragraaf 3.1.

2.3 Slingertraject

Er zijn veel strategieën om een slinger omhoog te slingeren. Bij deze FIT-stage is de methode, die wordt voorgesteld door Åström en Furuta in [5] gebruikt. Hier wordt de genormaliseerd energie van de pendulum als criterium gebruikt.

De bewegingsvergelijking van een enkele pendulum is:

$$I_2\ddot{\theta} - mgl_1 \sin \theta + mal_2 \cos \theta = 0 \quad (16)$$

Hierbij is frictie van het draaimechanisme van de slinger verwaarloosd. Verder is a de versnelling van het ophangpunt van de slinger (voor ons systeem geldt: $a = \ddot{\phi}l_1$). In [5] wordt voorgesteld de energie te normaliseren met mgl_2 en het nulpunt van de energie te kiezen als de slinger zich in staande positie bevindt. De resulteert in de volgende genormaliseerde energie:

$$E = mgl_2 \left(\frac{1}{2} \left(\frac{\dot{\theta}}{\omega_0} \right)^2 + \cos \theta - 1 \right) \quad (17)$$

Hierbij is $\omega_0 \equiv \sqrt{\frac{mgl_2}{I_2}}$. De energie is zo genormaliseerd dat deze 0 als maximum heeft. Wordt (17) gedifferentieerd naar de tijd dan wordt dit (m.b.v. (16)):

$$\frac{dE}{dt} = I_2\dot{\theta}\ddot{\theta} - mgl\dot{\theta} \sin \theta = -mul\dot{\theta} \cos \theta \quad (18)$$

Aan de hand van deze vergelijking wordt voor de versnelling van het ophangpunt het volgende voorgesteld:

$$u = \text{sat}_{ng}(k(E - E_0))\text{sign}(\dot{\theta} \cos \theta) \quad (19)$$

Hierbij is k een ontwerpparameter en de functie sat_{ng} is een functie die het bereik begrenst op ng , de maximale acceleratie van het ophangpunt. Hier dus equivalent ana de maximale versnelling van de motor. E_0 is het gewenste energieniveau en bovenstaande functie zal trachten dit niveau te bereiken.

Deze zogenaamde ‘bang-bang’-strategie is gebruikt om de slinger naar de geïnverteerde toestand te slingeren.

3 Simulaties

Er zijn simulaties uitgevoerd in Matlab zelf en met behulp van Matlab toolbox Simulink. De simulaties in Matlab zijn onder andere gebruikt om een feedback parameter te bepalen. In het model in Simulink is deze feedback gecombineerd met de ‘bang-bang’-strategie die wordt beschreven in paragraaf 2.3. Zodoende is gepoogd het (regel)systeem te simuleren.

3.1 Matlab

De LQR-methode is toegepast op de, in paragraaf 2.2 afgeleide state space vergelijking. Voor de volledige code hiervan wordt verwezen naar appendix A.

Voor een systeem in de vorm van

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (20)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (21)$$

met een feedback K toegevoegd

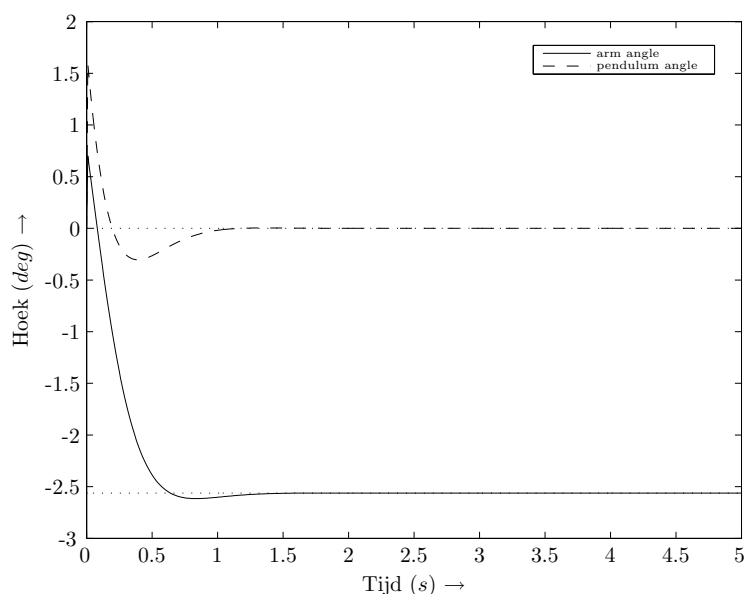
$$\mathbf{u} = \mathbf{K}\mathbf{x} \quad (22)$$

kan met behulp van de matlabfunctie `lqr` de parameter K worden bepaald. Nu verandert vergelijking 20, zodat de feedback kan worden meegenomen:

$$\dot{\mathbf{x}} = \mathbf{A}_c\mathbf{x} + \mathbf{B}\mathbf{r} \quad (23)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (24)$$

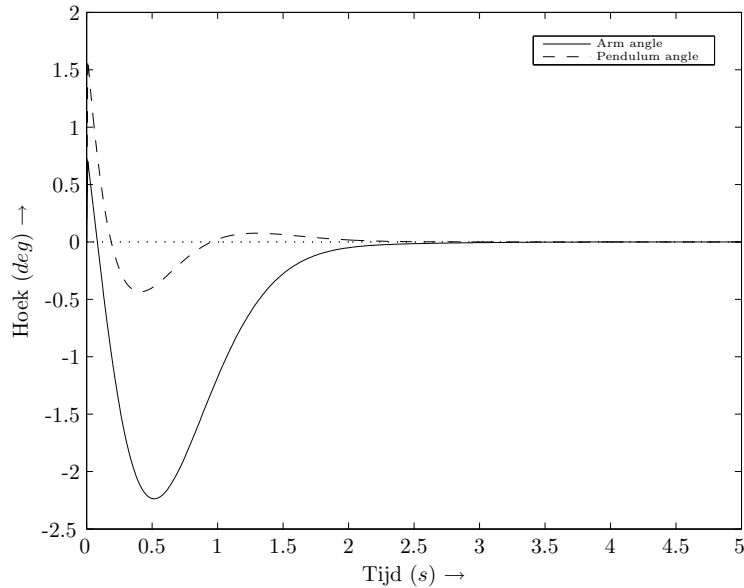
Hierin is $\mathbf{A}_c = \mathbf{A} - \mathbf{B}\mathbf{K}$. Merk op dat er nu een input variabele \mathbf{r} is. In figuur 2 is de staprespons te zien van het systeem met feedback \mathbf{K} .



Figuur 2: De step-response van het systeem met feedback K .

In figuur 2 is te zien dat de armhoek niet terug loopt naar nul, maar op een zeker waarde blijft steken. Dit is de zogenaamde steady state error. In [4] wordt op pagina's 323-327

hier over gesproken. De steady state error blijkt een waarde te hebben van $1/K_{11}$. K_{11} is de eerste waarde in de parameter \mathbf{K} . Vervolgens wordt een methode voorgesteld genaamd tracking waarmee de steady state error kan worden verholpen. Het resultaat hiervan is te zien in 3.



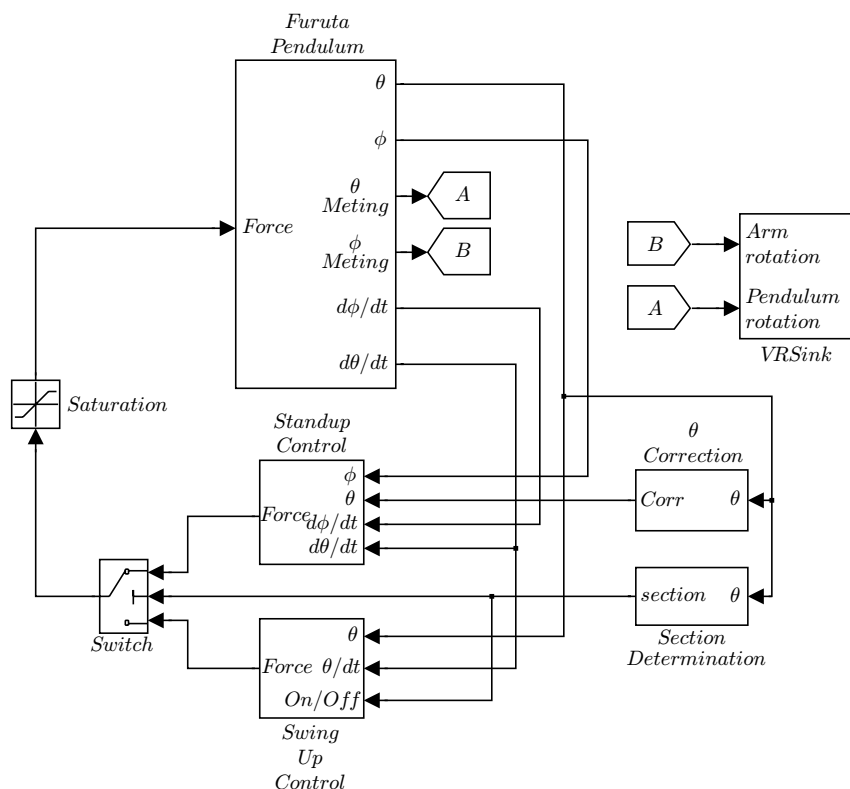
Figuur 3: De step-response van het systeem met feedback K en tracking om te compenseren voor de steady state error.

Omdat in het Simulink model ook nog andere onzekerheden zijn meegenomen, zoals bijvoorbeeld de beperkingen van de motor, zal de feedbackparameter zeker enige aanpassingen moeten ondergaan voor gebruik in de simulatie. Dit is dan ook de reden dat is besloten tracking niet mee te nemen in het simulink model.

3.2 Simulink

De bovenste laag van het systeem is te zien in figuur 4. Deze laag bestaat uit een aantal subsystemen. De kracht die wordt uitgeoefend op de motor van de Furuta pendulum (figuur 5) wordt bepaald door de ‘swingup’-routine (figuur 8) of de ‘standup’-routine (figuur 9). Deze keuze wordt gemaakt door het subsysteme in figuur 7. Verder moet de hoek θ van de slinger worden aangepast wil deze bruikbaar zijn voor het de ‘standup’-routine. Dit wordt gedaan in het subsysteme dat is afgebeeld in figuur 6.

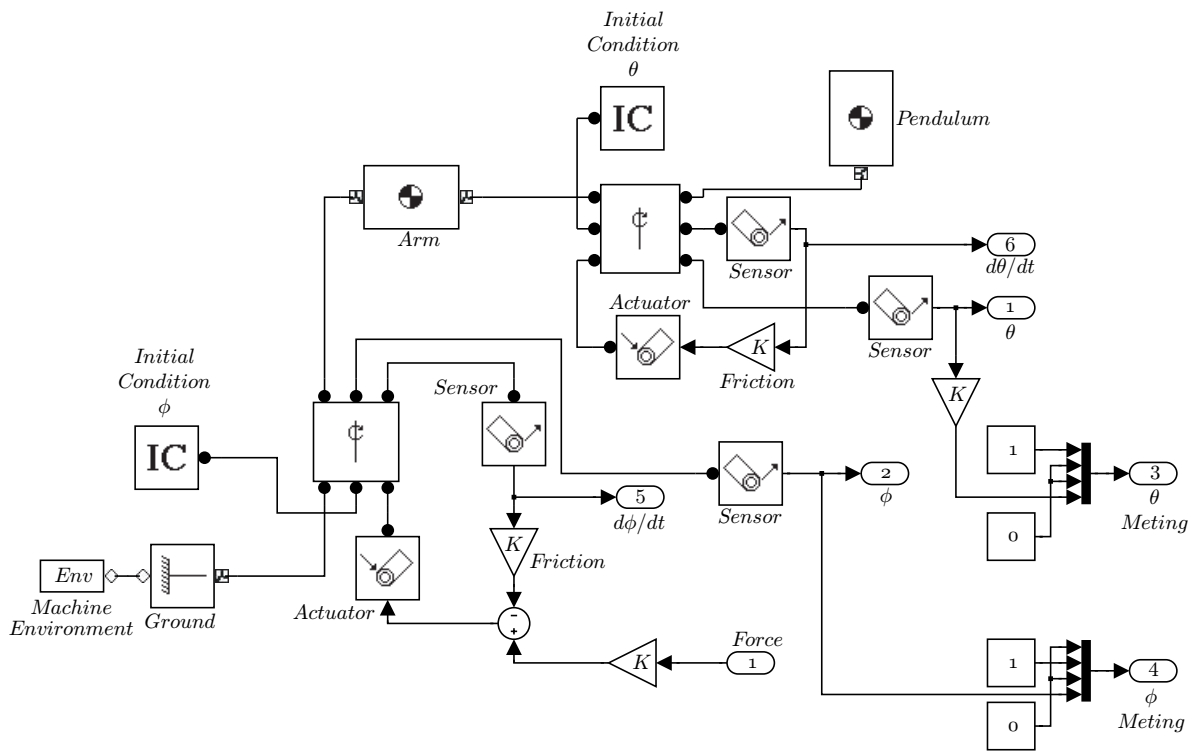
Tevens is in figuur 4 een VRSink te zien waarmee er een simpele visualisatie is gemaakt van de Furuta pendulum.



Figuur 4: Het eerste niveau van het Simulink model.

De maximale sterkte van de motor kan worden gesimuleerd met het ‘Saturation’-blok. Dit blok kan echter ook worden gebruikt om de kracht op een dusdanige wijze te beperken zodat de slinger met minimale snelheid de geïnverteerde positie bereikt of nadert. Als de snelheid namelijk te groot is zal de ‘standup’-routine niet in staat zijn om de slinger te stabiliseren en zal de slinger doorschieten. Hierna zal de ‘swingup’-routine trachten nog meer energie toe te voegen aan de slinger met als gevolg dat het alleen maar moeilijker wordt om deze te stabiliseren. Als de snelheid daarentegen minimaal is, is het voor de ‘standup’-routine gemakkelijker om de pendulum in evenwichtspositie te brengen en daar te houden. Dit komt omdat de ‘standup’-routine tot stand is gekomen aan de hand van een linearisatie en deze werkt dus alleen goed voor incrementele uitwijkingen.

Er zijn dus meerdere waarden voor de saturation waarmee de pendulum succesvol omhoog kan worden geslingerd. Voor hogere waarden zullen er vanzelfsprekend minder slingerbewegingen



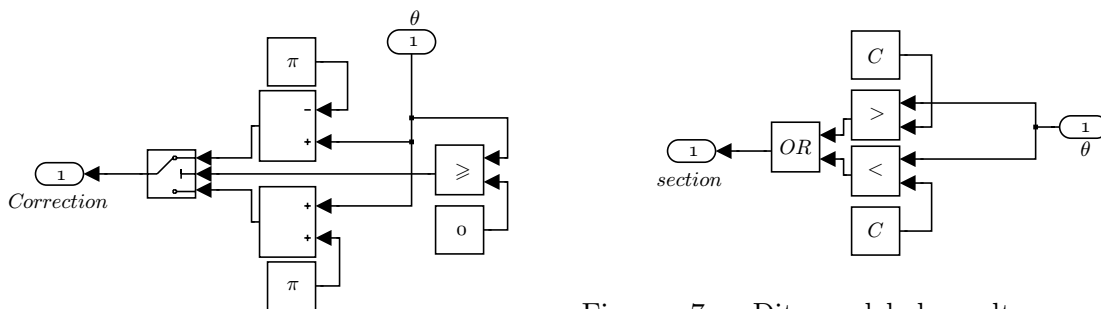
Figuur 5: Het model van de Furuta pendulum gerealiseerd m.b.v. SimMechanics.

nodig zijn, maar deze kunnen in de realiteit mogelijk niet worden gehaald door de motor.

Andere beperkingen van het systeem zoals de reactietijd en dodeltijd (van de motor of tandwielen) zijn te simuleren in Simulink. Dit is niet te zien in figuur 4, maar er is wel gekeken naar de invloed hiervan op het model.

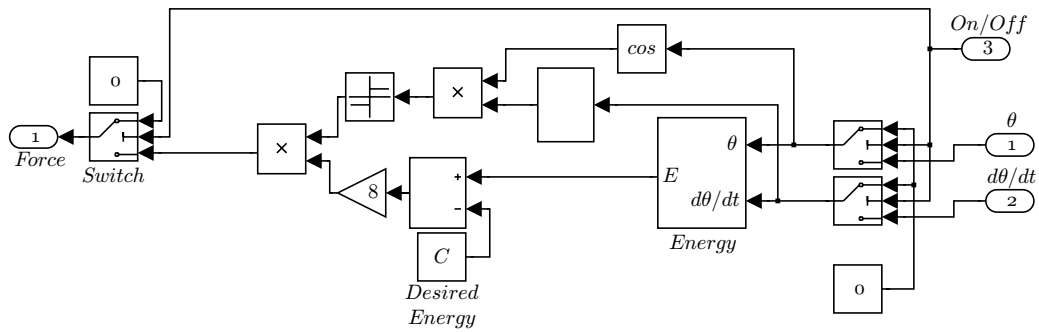
Figuur 5 toont het SimMechanics model van de Furuta pendulum. Er is voor SimMechanics gekozen en niet voor een model m.b.v. wiskundige uitdrukkingen in Simulink om het model overzichtelijk te houden. Tevens vallen aanpassingen (zoals bijvoorbeeld wijzigingen van traagheidsmomenten, wrijvingscoëfficiënten, beginposities, etc.) van het model gemakkelijker door te voeren.

Het model bestaat uit twee lichamen - een arm en een slinger (met gegeven traagheidsensors) - die via een draaipunt met elkaar zijn verbonden. De arm is op zijn beurt via een draaipunt



Figuur 6: Correctie van van pendulumhoek om bewerkingen mogelijk te maken.

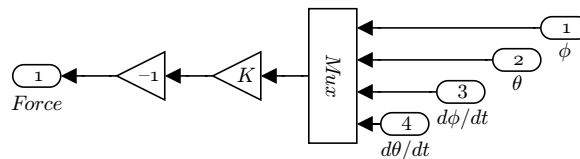
Figuur 7: Dit model bepaalt wanneer er moet worden gekozen voor de 'standup'- of de 'swingup'-routine.



Figuur 8: De 'swingup'-routine.

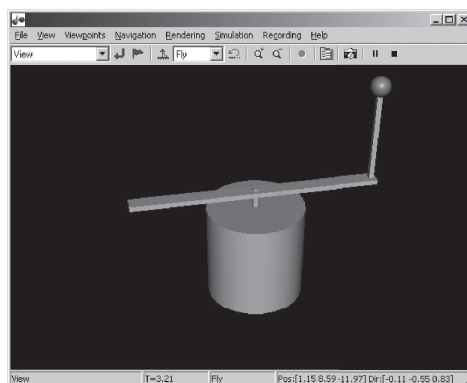
verbonden aan vast punt, in figuur 5 weergegeven door 'Ground'. Weerstand in deze draaipunten wordt gesimuleerd door de snelheid, vermenigvuldigd met de frotiecoëfficiënt, te laten uitwerken op dit draaipunt via een actuator. Via de outputs θ meting en ϕ meting worden gegevens voor de visualisatie geëxporteerd. Omdat de hoekmeting die wordt geproduceerd door SimMechanics een discontinuïteit bevat op het punt waarop de slinger de geïnverteerde positie bereikt. Is er een correctie doorgevoerd voor de variabele θ . Dit gebeurt door het subsysteem 'θ Correction', welke te zien is in figuur 6 op pagina 10.

In figuur 7 wordt bepaald in welke sectie de slinger zich bevindt. Wanneer $\theta \in [-20^\circ, 20^\circ]$,



Figuur 9: De 'standup'-routine.

waarbij het nulpunt van θ de geïnverteerde positie is, wordt de 'standup'-routine in werking gesteld. Bij alle andere hoeken wordt de 'swingup'-routine in werking gesteld. In figuur 8 is het subsysteem voor de 'swingup'-routine te zien. Dit is in wezen een realisatie in simulink van vergelijking (19). Voor de gewenste energie is een waarde van 0.1 gekozen. Tevens is te zien dat er een *On/Off* input is voor de routine. Dit is gedaan om te voorkomen dat de



Figuur 10: De visualisatie van de Furuta Pendulum m.b.v. de virtual reality toolbox

routine doorreken, terwijl diens output niet wordt gebruikt.

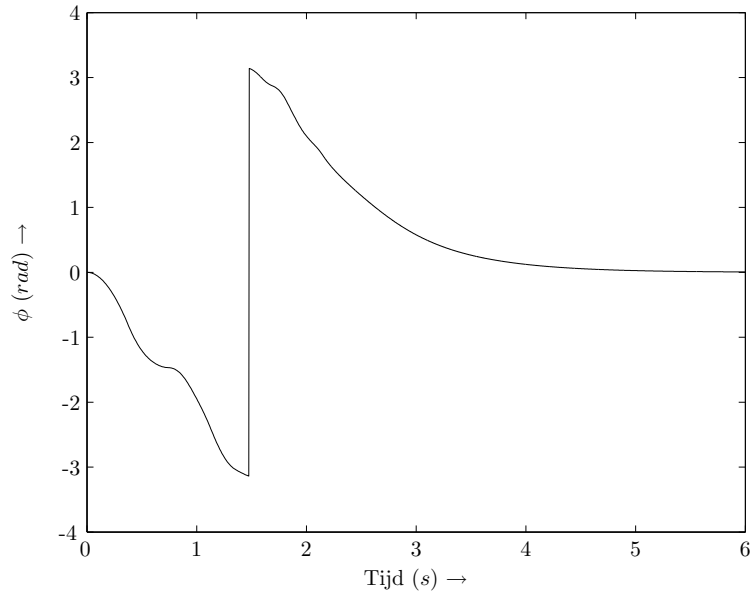
Figuur 9 laat het subsysteem van de ‘standup’-routine zien. In feite is dit niet meer van de vermenigvuldiging van in 3.1 berekende feedback parameter. Wel zijn hier via het trial & error de nodige aanpassingen op gedaan, om het ‘standup’-routine naar behoren te laten werken.

De visualisatie is gemaakt om sneller te kunnen inzien hoe veranderingen van de verschillende parameters op het systeem uitwerken. Een screenshot van de uiteindelijke visualisatie is te zien in figuur 10.

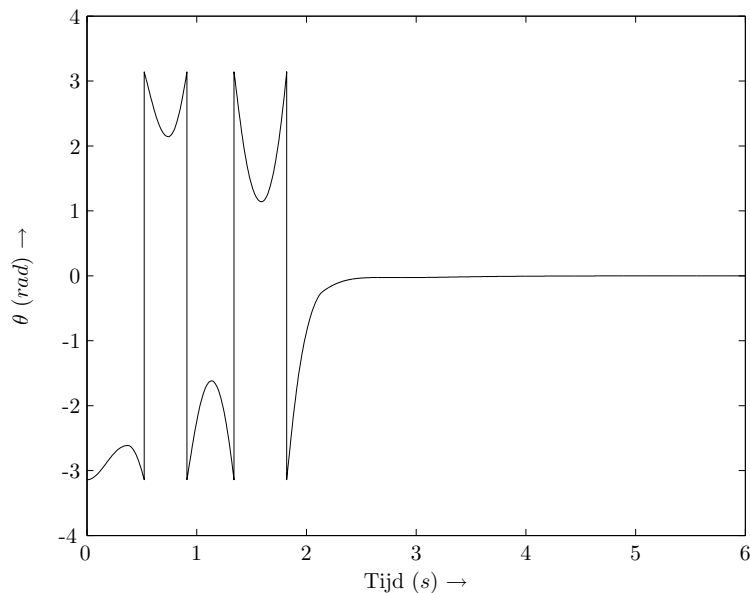
4 Resultaten en discussie

Helaas zijn er geen meetresultaten van opstelling, omdat deze nooit is gerealiseerd. Hier zullen dus alleen maar de resultaten van de opstelling worden besproken.

In figuur 11 en figuur 12 zijn respectievelijk de hoek van de arm, ϕ , en de hoek van de pendulum, θ , uitgezet tegen de tijd. Zoals blijkt uit figuur 12 heeft de slinger bij deze acceleratiesterkte vijf slingerbewegingen nodig om de slinger om hoog te brengen.



Figuur 11: De hoek ϕ uitgezet tegen de tijd.



Figuur 12: De hoek θ uitgezet tegen de tijd.

Als extra onzekerheden aan het systeem worden toegevoegd, zoals dodetijd en de responstijd van het systeem. Wordt voor deze waarde respectievelijk 1% van het de totale acceleratie en

10 *ms* gekozen, dan reageert het systeem nog naar behoren en verschillen de resultaten - afgezien van enige kleine discrepanties - niet noemenswaardig. Vergelijking met de meetgegevens van een experimenteel model is echter gewenst en zou een logische stap zijn.

5 Conclusie

Alhoewel de het simulatiemodel redelijk goed overeenstemming lijkt met de werkelijkheid, zouden diens resultaten wel moeten geverifieerd met behulp van een experimentele opstelling. Een aantal benaderingen van de realiteit zoals een eindige motoracceleratie (als gevolg van het eindige bereik van een versterker), alineaire overdracht van de tandwielen als gevolg van de backlash of de motor en de responstijd van de compensator zijn meegenomen. Echter kunnen schattingen van de invloed van deze invloeden onjuist zijn of zijn er nog andere invloeden die niet zijn meegenomen, hetgeen zou moeten worden nagegaan.

Ook zou een betere lineairisatiemethode kunnen worden gebruikt, zoals bijvoorbeeld wordt beschreven in [2] en [9]. Hierdoor zou het systeem beter te stabiliseren zijn in geïnverteerde positie.

Het slingertraject zoals dat nu in het model is toegepast, is ook verbetering vatbaar. Aan de hand [5] zou een nog efficiënter slingertraject kunnen worden gemaakt. Afhankelijk van de motorsterkte worden daar verschillende strategieën voorgesteld.

6 Dankwoord

Graag zou ik Dr. H. Hasper willen bedanken voor de assistentie bij deze FIT-stage.

7 Appendices

A Matlab

De code voor het bepalen van de feedbackparameter zoals is beschreven in paragraaf 3.1.

```
% Script_Cir_Pendulum.m
%-----
% FIT-Stage S.W. van Buuren
% Inverted Pendulum op een circulaire baan.
% Script file met test van de transferfuncties.
%-----

%Opruimen
clear all;
close all;

%Declaraties
m = .27;           %Massa Pendulum
L1 = .22;         %1/2 Length Arm
L2 = .1;          %1/2 Length Pendulum
g = 9.81;         %Gravity Constant
d1 = .00005;      %Rotor/Arm Friction (Bearing + Motor)
d2 = .000005;    %Pendulum Rolling Friction (Bearing)
I1 = 6.7e-5;      %Moment of Inertia Arm
I2 = 2.16e-7;    %Moment of Inertia Pendulum
Tau = .5;         %Input Output delay

%Definities van constanten
%Inverted position
D = m*L2^2*I1+(m*L1^2+I1)*I2;

%State Space Definition
A = [0 0 1 0;
     0 0 0 1;
     0 g*m^2*L1*L2^2/D -d1*(m*L2^2+I2)/D -m*d1*L1*L2/D;
     0 g*m*L2*(m*L1^2+I1)/D -m*d1*L1*L2/D -d2*(m*L1^2+I1)/D];
B = [0;
     0;
     (m*L2^2+I2)/D;
     m*L1*L2/D];
C = [1 0 0 0;
     0 1 0 0;
     0 0 0 0;
     0 0 0 0];
D = [0; 0; 0; 0];
sys_ss = ss(A, B, C, D);

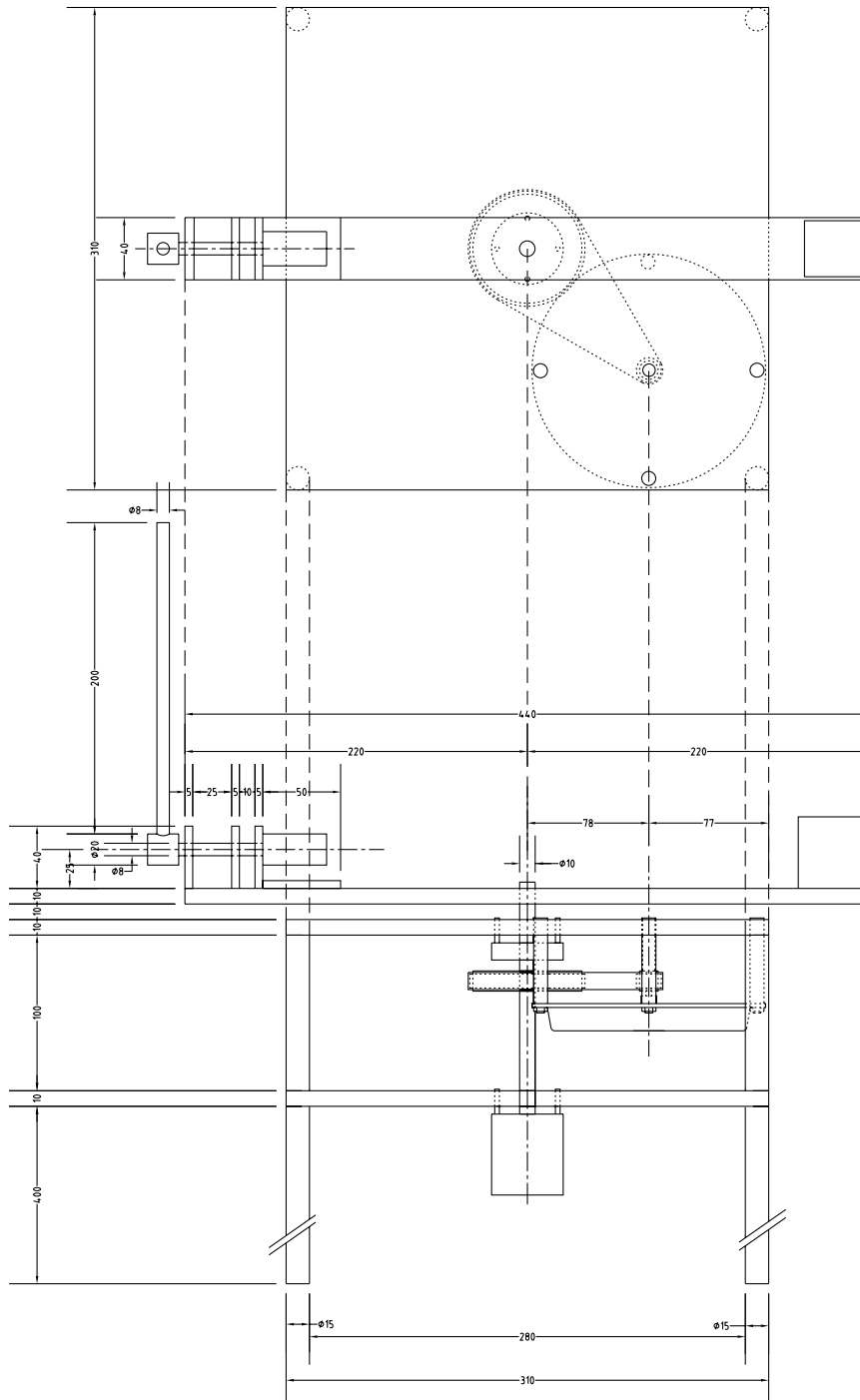
[NUM,DEN] = ss2tf(A,B,C,zeros(4,1))
```



```
%Feedback Design
%Controllability
fprintf('\nThe rank is %d.\n',rank(ctrb(A,B)));
k1 = 500;
k2 = 1000;
k3 = 200;
k4 = 500;
Q = [k1 0 0 0;
     0 k2 0 0;
     0 0 k3 0;
     0 0 0 k4];
R = 8;
%LQR-methode
[K, S, e] = lqr(sys_ss, Q, R)
An = A - B*K;
sys_lqr = ss(An, B, C, 0);
%Tracking met integral control
Ke = [-30 0 0 0];
Am = [An B*Ke;
     -C zeros(length(C),length(Ke))];
Bm = [B;
     zeros(length(B),1)];
Cm = [C zeros(length(C))];
sys_trck = ss(Am, Bm, Cm, 0);
sys_trck_delay = ss(Am, Bm, Cm, 0,'ioDelay',Tau);
```

B Opstelling

Er is een opstelling bedacht en ontworpen met behulp van autocad. Wegens te hoge kosten is deze opstelling echter nooit gemaakt.



Figuur 13: De opstelling getekend in Autocad.

Referenties

- [1] Marion, J.B. en Thornton, S.T., *Classical dynamics of particles and systems*, Harcourt, Orlando, (1995).
- [2] Sugie, R. and Fujimoto, K., Controller design for an inverted pendulum based on approximate linearization *International journal of robust and nonlinear control* **8**, 585-597 (1998).
- [3] Nise, N.S., *Control systems engineering*, Addison-Wesley, Menlo Park, 2nd edition, (1995).
- [4] Dutton, K. *et.al.*, *The art of control engineering*, Addison-Wesley, Essex, 657-664 (1997).
- [5] Åström, K.J. en Furuta, K., Swinging up a pendulum by energy control, *Proc. IFAC World Congress*, **13**, (1996).
- [6] Fraden, J., *Handbook of modern sensors*, AIP Press, Woodbury, 2nd edition, (1997).
- [7] Bentley, J.P., *Principles of Measurement systems*, Prentice Hall, Essex, 3rd edition, (1995).
- [8] Stimač, A.K., *Standup and Stabilization of the Inverted Pendulum*, Massachusetts Institute of Technology, Massachusetts, (1999).
- [9] Nair, S., Ehrich, L., A Normal Form for Energy ShapingL Application to the Furuta Pendulum, *Proc. 41st Conf. Decision and Control*, 2002.