

Project Mechatronisch Ontwerpen

Self – Balancing Robot

Afstudeerrichting: *Ma-ENE*

Groep samenstelling:

Booten Matthias

Hillen Kris

Indeherberg Yannick

Datum: 01/06/2017



Inhoud

1	Inleiding.....	3
1.1	Vooraf bepaalde doelstellingen	4
2	Opbouw.....	5
2.1	Microcontroller.....	5
2.2	Accelerometer	6
2.3	Gyroscoop.....	6
2.4	Motor driver	7
2.5	Motor.....	8
2.6	Bluetooth Module	9
2.7	Batterij.....	9
3	Aansluitschema	10
4	Sensorwerking testen met LabView.....	11
4.1	Metingen van de gyroscoop	11
4.2	Metingen van de accelerometer	13
5	Matlab Programma	14
5.1	Bepalen van de transfert functie van het systeem.	14
6	Regelaar en filter	23
6.1	Inleiding.....	23
6.2	Algemene blokdiagram.....	23
6.3	Gekozen regelaar.....	24
7	Arduino programma.....	25
8	Interface met Arduino Bluetooth Controller	29
9	Conclusies.....	30
9.1	Motor.....	30
9.2	Motordriver	31
9.3	Matlab	31
9.4	Labview	31
9.5	Constructie	31
9.6	Taakverdeling en planning	33
9.7	Kostenplaatje	33
10	Bronvermelding.....	34
10.1	Bronvermelding figuren	34
10.2	Literaire bronnen	35

11	Datasheets	36
----	------------------	----

Afbeeldingen lijst

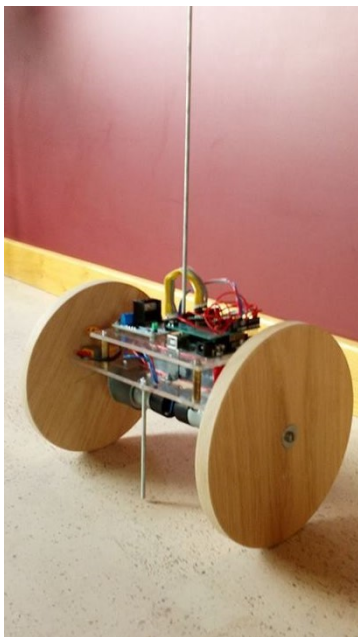
Figuur 1-1	Self-balancing robot	3
Figuur 2-1	ARDUINO UNO & benaming aansluitpinnen	5
Figuur 2-2	Gyroscoop.....	6
Figuur 2-3	montage bordje L298N	
Figuur 2-4	L298N motor driver.....	
Figuur 2-5	motor.....	8
Figuur 2-6	Bluetooth module	9
Figuur 3-1	Aansluitschema	10
Figuur 4-1	Interface labview programma gyroscoopmeting	
Figuur 4-2	Interface Labview programma accelerometer meting	13
Figuur 5-1	Open lus impuls respons	15
Figuur 5-2	Root-Locus Editor.....	17
Figuur 5-3	Wortellijn diagram.....	17
Figuur 5-4	Marginale stabiliteit bij PD-regelaar	19
Figuur 5-5	Stap respons & Bode plot voor PD-regelaar	21
Figuur 5-6	Gesimuleerde data voor geoptimaliseerde PD-regelaar	21
Figuur 5-7	Stap respons en Bode plot voor geoptimaliseerde PID-regelaar.....	22
Figuur 2-8-1	ARDUINO UNO & benaming aansluitpinnen	34
Figuur 8-2	Gyroscoop.....	34
Figuur 8-3	motor.....	34

Tabellen lijst

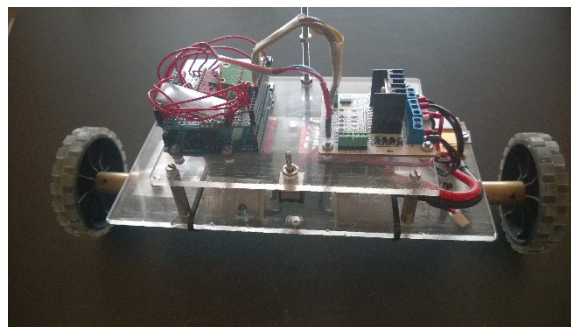
Tabel 1	Motor data tabel.....	8
Tabel 2	Resultaten PD-regelaar	18
Tabel 3	Kostenplaatje.....	33

1 Inleiding

Als onderwerp werd er gekozen voor een self-balancing robot. De robot in kwestie bestaat uit een frame gemonteerd op twee door DC-motoren aangedreven wielen. De motor balanceert zichzelf door de wielen te laten draaien in de valrichting. Hierdoor komt het frame terug in de evenwichtspositie.



Figuur 1-2 Self-balancing robot met grote wielen



Figuur 1-1 Self-balancing robot met kleinere wielen

1.1 Vooraf bepaalde doelstellingen

Volgende doelstellingen werden door de studenten zelf opgelegd voor hun project.

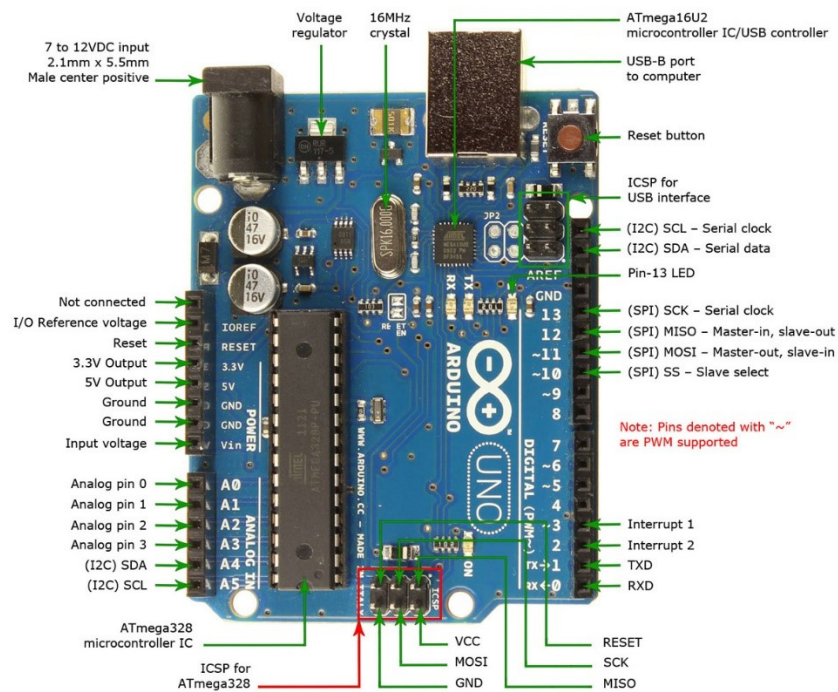
- Het bouwen van een zelf balancerende robot op twee wielen.
- Balancerings waartoe volgende aspecten behoren.
 - Meeting
 - Gyroscop
 - Meet graden per seconden (hoeksnelheid)
 - Wordt gebruikt om te kijken of de robot vooruit of achteruit beweegt.
 - Accelerometer
 - Meet hoekversnelling
 - Wordt gebruikt om de positie van de robot te bepalen.
 - Controller
 - Als achterliggende controller wordt beoogd een microcontroller te gebruiken.
 - Aandrijving
 - Een motor per wiel
 - Een motor driver
- Eventuele uitbreidingen
- Een mooi afgewerkt geheel

2 Opbouw

2.1 Microcontroller

Arduino UNO

Het brein achter de self-balancing robot is de microcontroller. Hiervoor werd er gekozen voor de Arduino UNO. De taak van deze microcontroller is het inlezen van verschillende gegevens die voortkomen uit de accelerometer en de gyroscoop alsook het correct aansturen van de motordriver en bijgevolg de DC motoren.

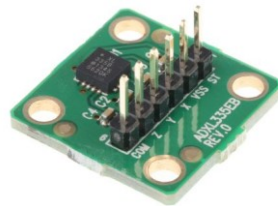


Figuur 2-1 ARDUINO UNO & benaming aansluitpinnen

2.2 Accelerometer

ADXL 335EB

Een accelerometer is een meetelement dat in staat is de versnelling te meten [m/s^2]. De gebruikte accelerometer is in staat te meten volgens drie assen (X, Y, Z). De gemeten versnelling wordt weergegeven als een spanningssignaal [V]. Het meetbereik omvat $\pm 3g$ of 3 keer de aardversnelling in positieve en negatieve zin. Wanneer de robot stilstaat en zelf geen versnelling veroorzaakt is deze nog wel onderhevig aan de aardversnelling ($g = 9,81 \text{ m/s}^2$). Hierdoor zal bij een robot die zich in rust bevindt wel nog een aardversnelling gemeten worden. Uitgangssignalen spanningssignalen die hoger of lager liggen dan het signaal overeenkomstig met dat van de aardversnelling.

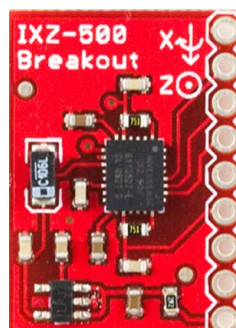


Figuur 2-2 Accelerometer

2.3 Gyroscoop

IXZ-500

De gyroscoop is een meetelement dat in staat is om de hoeksnelheid (symbool ω) in graden per seconde te meten [$^\circ/\text{s}$] om de x-as en de z-as. Bij rotatie om een van deze assen zal er dus een uitgangssignaal volgen. Voor de balancing robot wordt er gebruik gemaakt van het uitgangssignaal van de x-as. Hierdoor is het programma in de microcontroller in staat te bepalen dat de robot zich in de voorwaartse of achterwaartse hellende positie bevindt. Dit signaal is nodig voor correcties in balanceren maar ook voor het behouden van een correcte evenwichtspositie bij voorwaartse of achterwaartse bewegingen. Het uitgangssignaal van de gyroscoop werd in kaart gebracht met behulp van een LabView programma meer uitleg hierover volgt hierover in paragraaf [4].

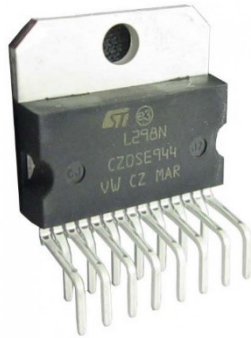


Figuur 2-3 Gyroscoop

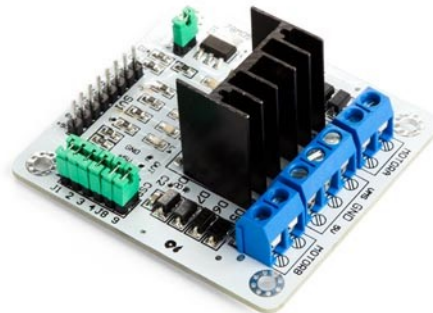
2.4 Motor driver

L298N DUAL BRIDGE DC STEPPER CONTROLLER BOARD

De motor driver berust op de werking van de L298N I.C.(*). Deze driver is in staat 2 DC motoren tegelijkertijd aan te sturen of 1 stappenmotor. Voor de toepassing van de balancing robot werd er gebruikgemaakt van de mogelijkheid om 2 DC motoren aan te sturen. Dit hield dat de driver in stond voor de snelheid en draairichting van de respectievelijke motoren.



Figuur 2-5
L298N motor
driver



Figuur 2-4
montage bordje L298N

De motor driver werd tegelijkertijd aangeschaft met een handig montage bordje voorzien van koelvinnen waartegen de motor driver I.C. (*) tegen bevestigd wordt en gepaste aansluit pinnen(**). Voorts werd er gebruik gemaakt van de resterende +5V en GND uitgangen op de motordriver om de Arduino UNO mee te bekrachtigen.

*I.C. → integrated component

**Extra informatie omtrent aansluitpinnen van de motordriver zie link datasheet in bijlagen.

2.5 Motor

RB350018-2A723R 12 V/DC

Technische specificaties	
Tandwielreductie	18:1
Categorie	Krachtige transmissiemotor
Max. draaimoment	2.06 Nm
Voedingsspanning	12 V/DC
Nominale spanning	12 V
Aslengte	15 mm
Onbelast toerental bij nom. spanning	333 omw/min
Max. belastingsstroom	0,7 A
Belast toerental	317 omw/min
As-Ø	6 mm
Nullast-stroom	0,32 A
Lengte	101.6 mm
Diameter	36 mm

Tabel 1 Motor data tabel

Als motor werd er gekozen voor DC-motoren met een aanwezige reductor overbrenging. De volledige datasheet is te vinden in de bijlagen. Zowel toerental als leverbaar koppel werd in acht genomen bij de keuze van deze motor. Hier wordt dieper op ingegaan in paragraaf [9-1]



Figuur 2-6 motor

Berekening benodigd koppel

Voor het bepalen van het maximum koppel van de motor werd onderstaande formule gebruikt. Deze formule maakt gebruik van het gewicht van de robot en de maximale herstel hoek van waaruit de robot zijn positie dient te herstellen.

$$T_{max} = m \cdot g \cdot l \cdot \sin \theta_{max} = 1,5 \cdot 9,81 \cdot 0,05 \cdot \sin(5^\circ) = 0,064 \text{ Nm of } T_{max}=6,4\text{Ncm [1]}$$

De motor levert met 2,06Nm voldoende koppel om de robot te balanceren.

2.6 Bluetooth Module

HC-06

De bluetooth module staat in voor het onderhouden van bluetooth communicatie met de balancing robot. Hieronder valt het signaal van in bedrijfsstelling maar ook bewegingscommando's.



Figuur 2-7 Bluetooth module

2.7 Batterij

LiPo accupack 11.1 V 2400 mAh 20 C Conrad energy Stick XT60

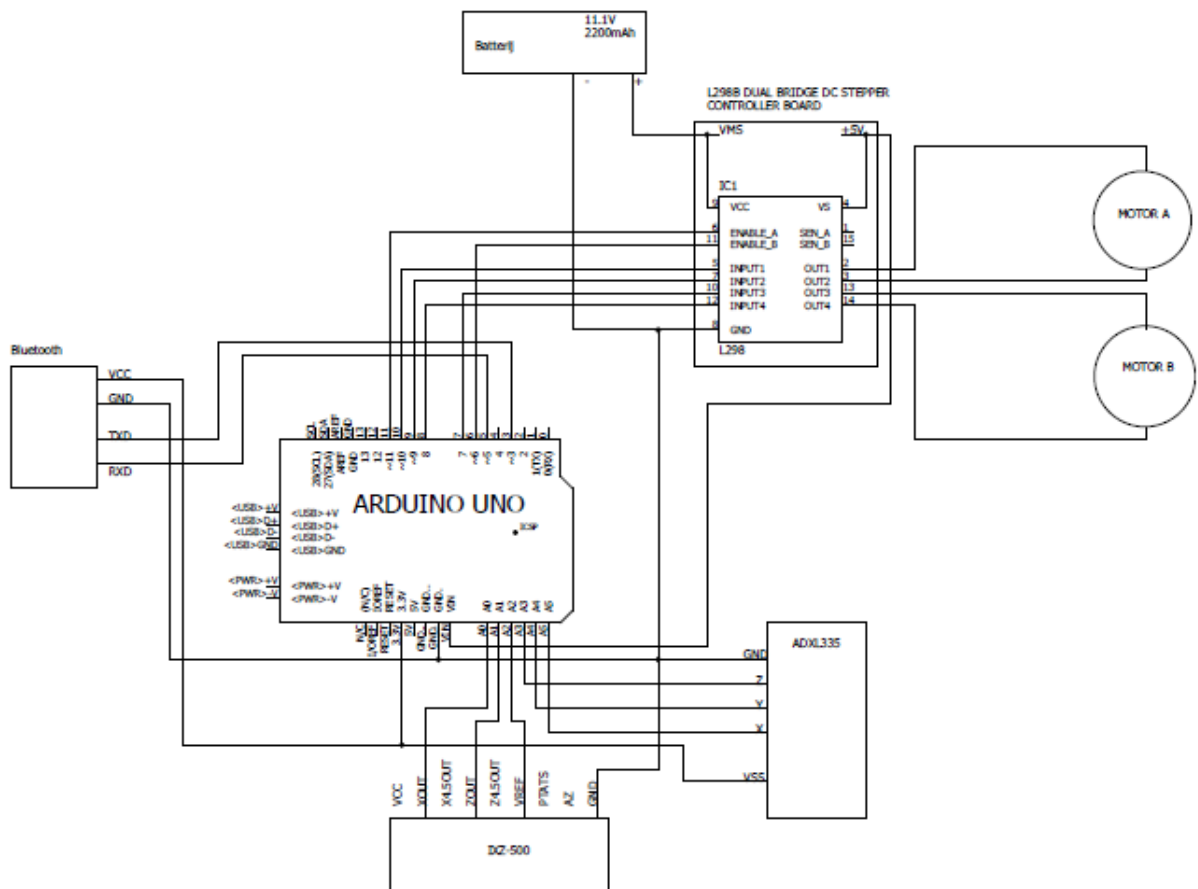
Aangezien LiPo batterijen beschikken over een goede energie opslag met beperkte afmetingen en prijs werd er gekozen voor een LiPo batterij van 2400mAh.



Figuur 2-8 Batterij

3 Aansluitschema

Hieronder wordt het aansluitschema weergegeven van de voorgenoemde componenten.

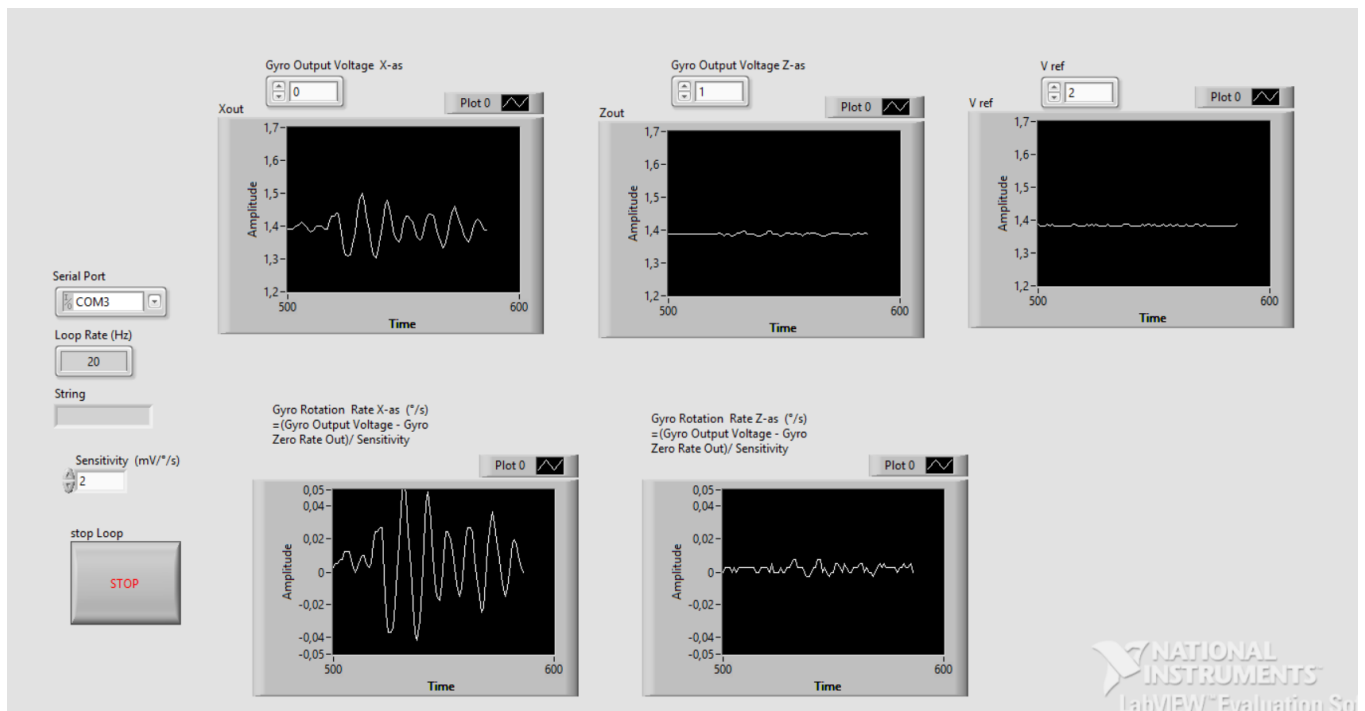


Figuur 3-1 Aansluitschema

4 Sensorwerking testen met LabView

Om de werking van de gyroscoop en de accelerometer te bestuderen en een beter begrip te krijgen van hun uitgangssignalen werd onderstaand Labview programma aangemaakt.

4.1 Metingen van de gyroscoop



Figuur 4-1 Interface labview programma gyroscoopmeting

Serial port

De serial port komt overeen gekozen de port waarmee men met de Arduino connecteert. Voor dit project werd er gekozen gebruik te maken van COM3.

Loop Rate (Hz)

De looprate geeft aan met welke frequentie het programma in LabView wordt doorlopen tijdens het inlezen van de analoge signalen. Dit is vooral ter controle van de goede werking van het programma.

Sensitivity

De sensitivity of gevoeligheid is de waarde om het spanningssignaal van de gyroscoop om te rekenen naar een bruikbare waarde. In dit geval om de spanning om te rekenen naar het aantal verdraaide graden per seconde.

Gyro Output Voltage X-as

Het eerste scherm geeft het uitgangsvoltage dat overeenkomt met een verdraaiing om de Z-as. Des te groter de snelheid van de verdraaiing van de robot om de X-as in $^{\circ}/s$, des te groter de uitwijking van het referentie signaal dat 1,4V bedraagt.

Gyro Output Voltage Z-as

Dit scherm vertoont het uitgangsvoltage dat overeenkomt met een verdraaiing om de Z-as. Het uitgangsvoltage stemt overeen met de grote van de hoeksnelheid, hoe groter de waargenomen hoeksnelheid om de Z-as in °/s, des te groter de uitwijking t.o.v. het referentie signaal dat 1,4V bedraagt.

Vref

Het Vref scherm vertoont de referentie spanning waarmee de waargenomen signalen worden vergeleken.

Gyro rotation rate X-as (°/s)

De uitgangsspanning die wordt weergegeven op dit scherm is het geijkte signaal van de rotatie om de X-as. Dit wordt bekomen volgens onderstaande formule.

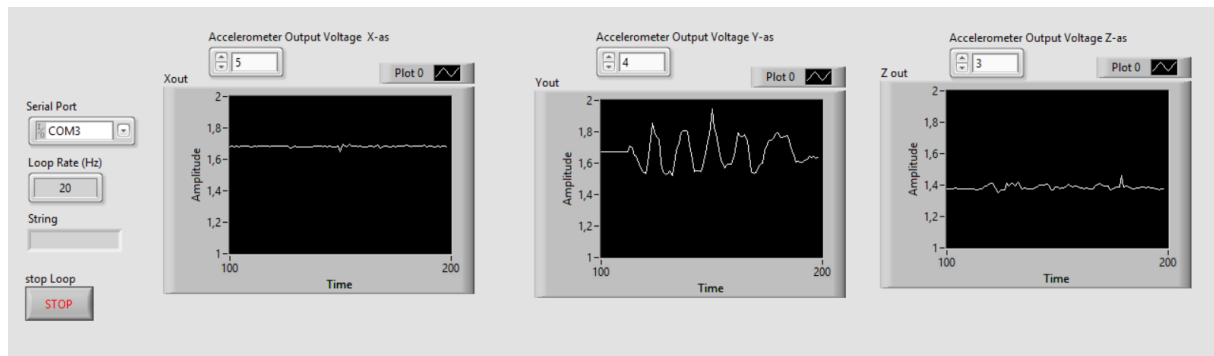
$$\text{Gyro Rotation Rate X - as} = (\text{Gyro Output Voltage X - as}) - (V_{ref}) \quad [V]$$

Gyro rotation rate Z-as (°/s)

De uitgangsspanning die wordt weergegeven op dit scherm is het geijkte signaal van de rotatie om de Z-as. Dit wordt bekomen volgens onderstaande formule.

$$\text{Gyro Rotation Rate Z - as} = (\text{Gyro Output Voltage Z - as}) - (V_{ref}) \quad [V]$$

4.2 Metingen van de accelerometer



Figuur 4-2 Interface Labview programma accelerometer meting

Serial port , Loop Rate (Hz), Sensitivity

Bovenstaande parameters kennen de zelfde functie als hun tegenhangers in het

Aanpasbare waarden erboven [5][4][3]

Geeft weer via welke input van de Arduino er wordt ingelezen.

Accelerometer Output Voltage X-as & Output Voltage Z-as

Deze assen werden ook gesimuleerd om de werking ervan te testen. Deze signalen zijn echter niet vereist voor de werking van de robot.

Accelerometer Output Voltage Y-as

De y-as wordt gebruikt om de hoekverandering per seconde uit te drukken om de y-as. Dit is de as waarrond de robot moet balanceren. Het signaal vertegenwoordigt de verandering van de hoek. (mv/°/s)

Bij een meting van 1,65V mogen we de robot als waterpas beschouwen. De gemeten acceleratie is in dit geval nog enkel ten gevolge van de aardversnelling.

5 Matlab Programma

5.1 Bepalen van de transfert functie van het systeem.

Voor een correcte bepaling van de transfert functie werd de opbouw van de robot opgesplitst in twee delen. Het eerste deel omvat de basis en motors van de robot. Het tweede deel omvat de staaf maar ook alles wat boven de basis uit komt en als deel van de inverse pendulum kan worden beschouwd. Dit houdt bijkomend dus ook nog de batterij en alles wat er op het bovenste plaatje van de robot gemonteerd is in.

Gebruikt Matlab programma:

```
M = 0.5; %Massa van de blokvormige basis (2motoren en
plankje) (ruw geschat zonder wielen)
m = 0.75; %Massa van pendulum (de staaf + batterij en alles
bovenste plankje)
b = 0.1; %Geschatte wrijvingscoefficient van de wielen
I = 0.006; %Traagheidsmoment pendulum
g = 9.81; %Aardversnelling
l = 0.3; %Geschatte afstand tot zwaartepunt (stang tot
center vd basis)
q = (M+m)*(I+m*l^2)-(m*l)^2;
s = tf('s');

P_basis = (((I+m*l^2)/q)*s^2 - (m*g*l/q))/(s^4 + (b*(I + m*l^2))*s^3/q -
((M + m)*m*g*l)*s^2/q - b*m*g*l*s/q);

P_pend = (m*l*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q -
b*m*g*l/q);

sys_tf = [P_basis ; P_pend]; %Dit vormt een matrix met 1 kolom en 2 rijen

inputs = {'u'};
outputs = {'x'; 'phi'}; %Transfert functie 1 per rij in de gemaakte matrix

set(sys_tf, 'InputName', inputs)
set(sys_tf, 'OutputName', outputs)

%Vervolgens kan de open-loop respons van een impuls op het systeem worden
bekeken.
%Met onderstaande code wordt er nagegaan hoe het systeem reageert wanneer
%er plots een kracht (een impuls) op het systeem wordt uitgeoefend.
%Dit wordt gedaan aan de hand van het MATLAB commando impulse.
%Hierbij wordt er een plot gegeven die de reactie van het systeem weergeeft
%per tijdseenheid.

t=0:0.01:1;
impulse(sys_tf,t);
title('Open-Loop Impulse Response')

sys_tf
```

Resultaten:

Hierbij zijn we enkel geïnteresseerd in de transfertfunctie bijhorend aan de hoek phi van het systeem. Hierbij bekomen we een systeem van derde orde.

```
>> MatlabTransferFunctionBalancingRobot
```

```
sys_tf =
```

From input "u" to output...

$0.0001251 \text{ s}^2 - 0.003756$

x: -----

$7.019\text{e-}05 \text{ s}^4 + 1.251\text{e-}05 \text{ s}^3 - 0.004695 \text{ s}^2 - 0.0003756 \text{ s}$

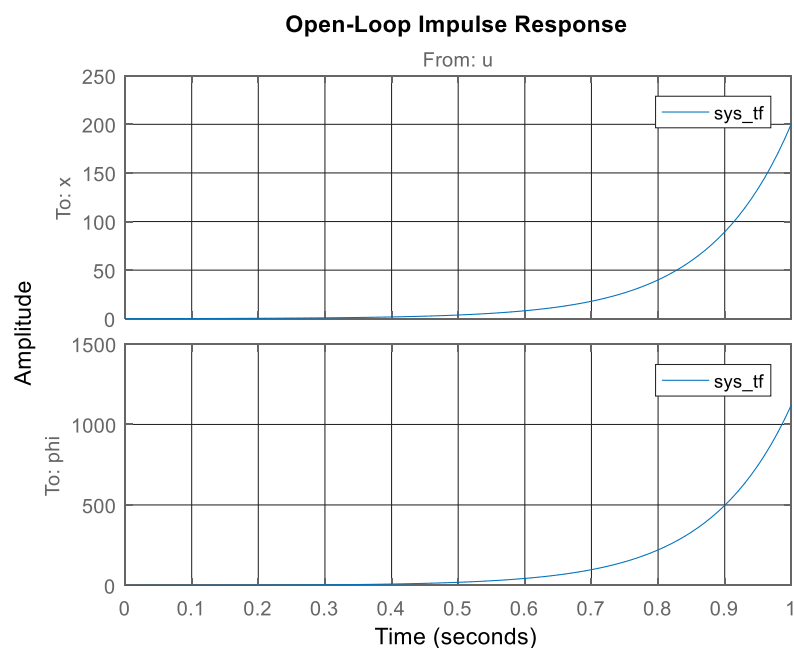
0.0003829 s

phi: -----

$7.019\text{e-}05 \text{ s}^3 + 1.251\text{e-}05 \text{ s}^2 - 0.004695 \text{ s} - 0.0003756$

Continuous-time transfer function.

Bijkomend werd er in de Matlab code een plot voorzien om de open-loop impuls respons van het systeem weer te geven.



Figuur 5-1 Open lus impuls respons

Rltool

Om een beter beeld te bekomen van de stabiliteit wordt het wortelijndiagram getekend met behulp van Rltool in Matlab.

Programma:

```
M = 0.5; %Massa van de blokvormige basis (2motoren en
plankje) (ruw geschat zonder wielen)//NOG EENS WEGEN
m = 0.75; %Massa van pendulum (de staaf + batterij en alles
bovenste plankje) //NOG EENS WEGEN
b = 0.1; %Geschatte wrijvingscoefficient van de wielen
I = 0.006; %Traagheidsmoment pendulum
g = 9.81; %Aardversnelling
l = 0.3; %Geschatte afstand tot zwaartepunt (stang tot
center vd basis)
q = (M+m)*(I+m*l^2)-(m*l)^2;
s = tf('s');

P_basis = (((I+m*l^2)/q)*s^2 - (m*g*l/q))/(s^4 + (b*(I + m*l^2))*s^3/q -
((M + m)*m*g*l)*s^2/q - b*m*g*l*s/q);

P_pend = (m*l*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q -
b*m*g*l/q);

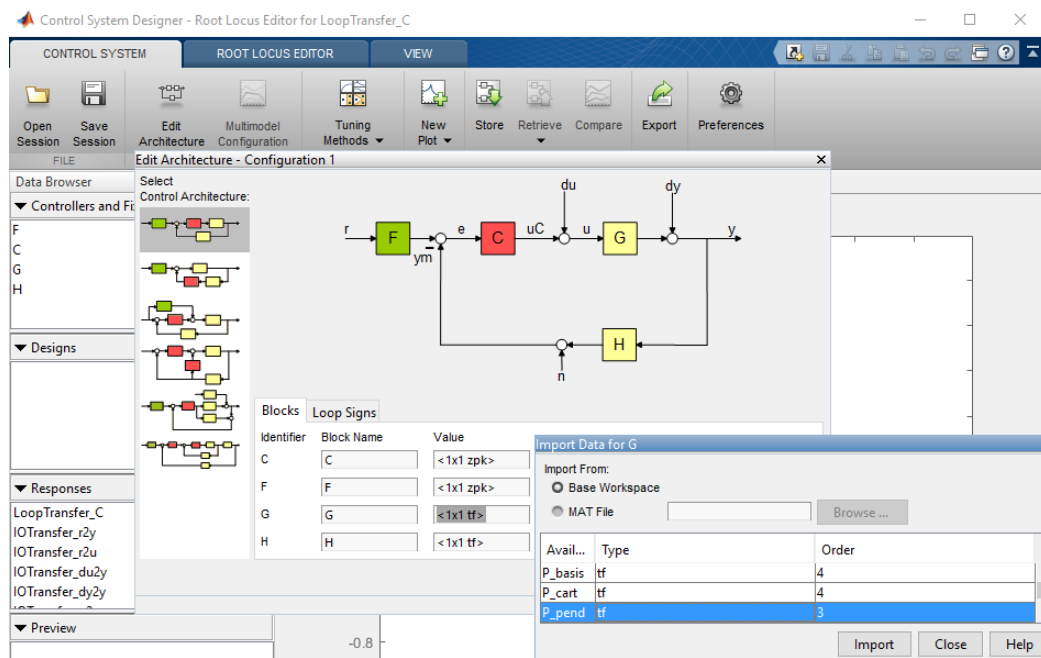
sys_tf = [P_basis ; P_pend]; %Dit vormt een matrix met 1 kolom en 2 rijen

inputs = {'u'};
outputs = {'x'; 'phi'}; %Transfert functie 1 per rij in de gemaakte matrix

set(sys_tf, 'InputName', inputs)
set(sys_tf, 'OutputName', outputs)

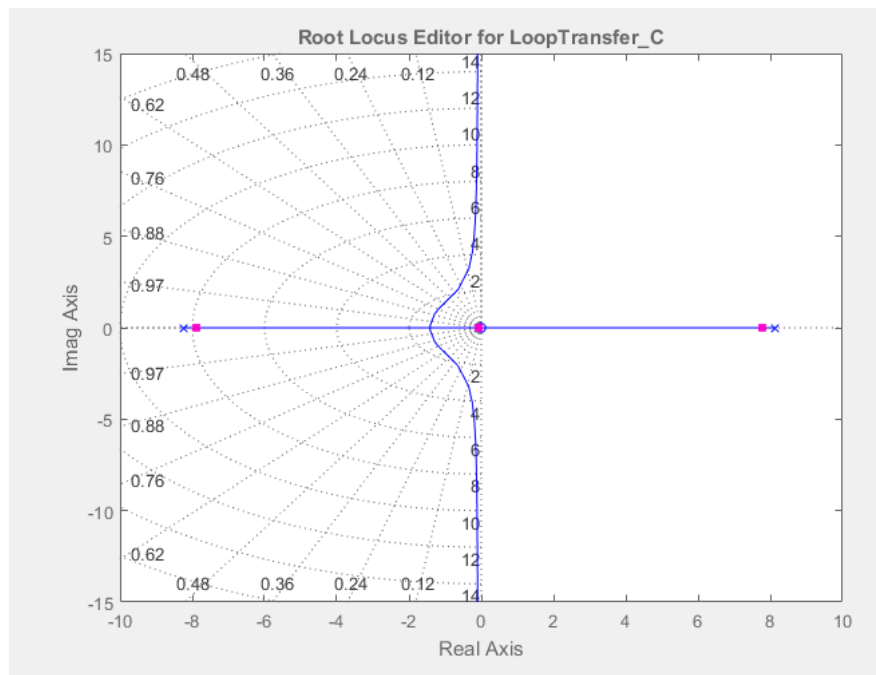
sys_tf

rltool; %Toevoegen van rltool voor een root locus diagram (wortellijnen
diagram)
```



Figuur 5-2 Root-Locus Editor

Bij het doorlopen van het Matlab programma wordt het Root Locus diagram getekend. Oorspronkelijk is deze leeg. Door bij Edit Architecture de transfer functie van de pendulum in te laden bij het blokje G uit het blokdiagram wordt het wortellijndiagram in figuur 5-3 getekend.



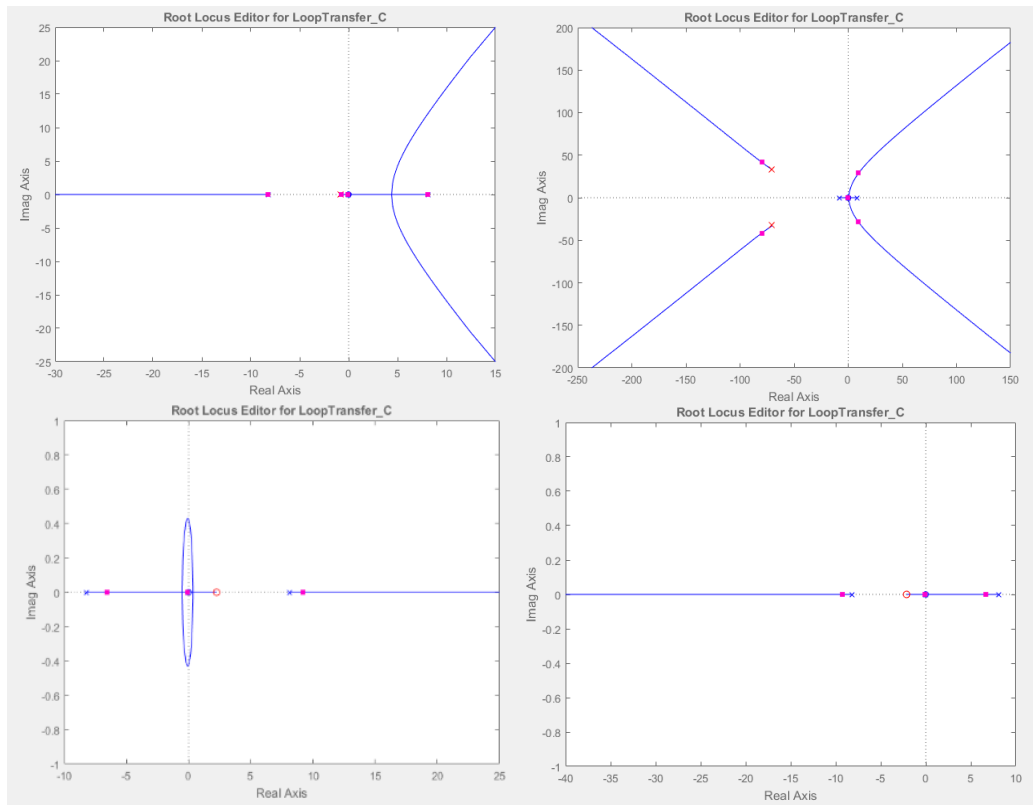
Figuur 5-3 Wortellijn diagram

Conclusie

Doordat er zich een pool in het rechtervlak bevindt kan er worden geconcludeerd dat dit systeem instabiel is.

Kijken naar de bruikbaarheid van een PD-regelaar

Door het toevoegen van een PD-regelaar komt er een nulpunt bij op het wortellijn diagram. Er wordt hieronder gekeken hoe aanpassingen hoe dit de stabiliteit beïnvloed met als doel het bereiken van stabiliteit.



Figuur 5-4 Resultaten wortellijndiagram PD-regelaar

Figuur	Nulpunt/pool	L-R half vlak	Bereikt stabiliteit
Links boven	reële pool	L	Nee
Rechts boven	complexe pool	L	Nee
Links onder	reëel nulpunt	R	Nee
Rechts onder	reëel nulpunt	L	Nee

Tabel 2 Resultaten PD-regelaar

*Toevoegen van een complex nulpunt droeg op zowel het linker als rechter half vlak niet voldoende bij tot het bereiken van stabiliteit.

*Toevoegen van zowel een reële als een complexe pool op het rechter half vlak droeg niet voldoende bij tot het bereiken van stabiliteit.

Onder welke omstandigheden is een PD-regelaar bruikbaar?

Matlab programma

```
M = 0.5; %Massa van de blokvormige basis (2motoren en
plankje) (ruw geschat zonder wielen)//NOG EENS WEGEN
m = 0.75; %Massa van pendulum (de staaf + batterij en alles
bovenste plankje) //NOG EENS WEGEN
b = 0.1; %Geschatte wrijvingscoefficient van de wielen
I = 0.006; %Traagheidsmoment pendulum
g = 9.81; %Aardversnelling
l = 0.3; %Geschatte afstand tot zwaartepunt (stang tot
center vd basis)
q = (M+m)*(I+m*l^2) - (m*l)^2;
s = tf('s');

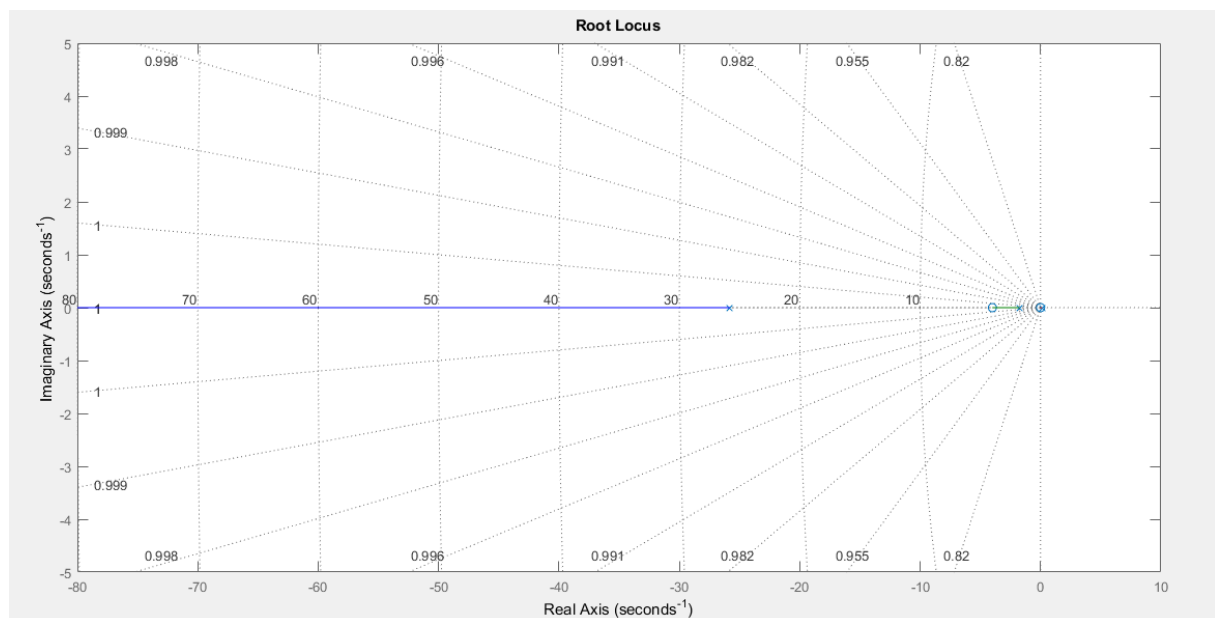
P_pend = (m*l*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q -
b*m*g*l/q);

%%
Kp = 20; %Versterking P-actie
Ki = 0; %Versterking I-actie
Kd = 5; %Versterking D-actie

H=1; %Eenheidsterugkoppeling

C = pid(Kp,Ki,Kd); %regelaar/controller
T = feedback(C*P_pend,H);

rlocus(T);
```



Figuur 5-5 Marginale stabiliteit bij PD-regelaar

In figuur 5-5 vallen een nulpunt en een pool samen in de oorsprong, het wortellijn diagram vertoont marginale stabiliteit. Voorgaande vaststellingen wijzen erop dat het systeem op een andere wijze stabiliteit zal moeten bereiken.

Bepalen van de geschikte regelaar

Rekening houdend met voorgaande resultaten wordt er bij het bepalen van een geschikte regelaar ook gekeken naar het transiënt gedrag en responsie tijd. Er wordt voor deze toepassing verwacht dat de regelaar in staat is op een snelle en evenwichtige manier te reageren. Agressieve of wispelturige reacties worden niet aanschouwd als voordelen voor de regelaar van de balancing robot.

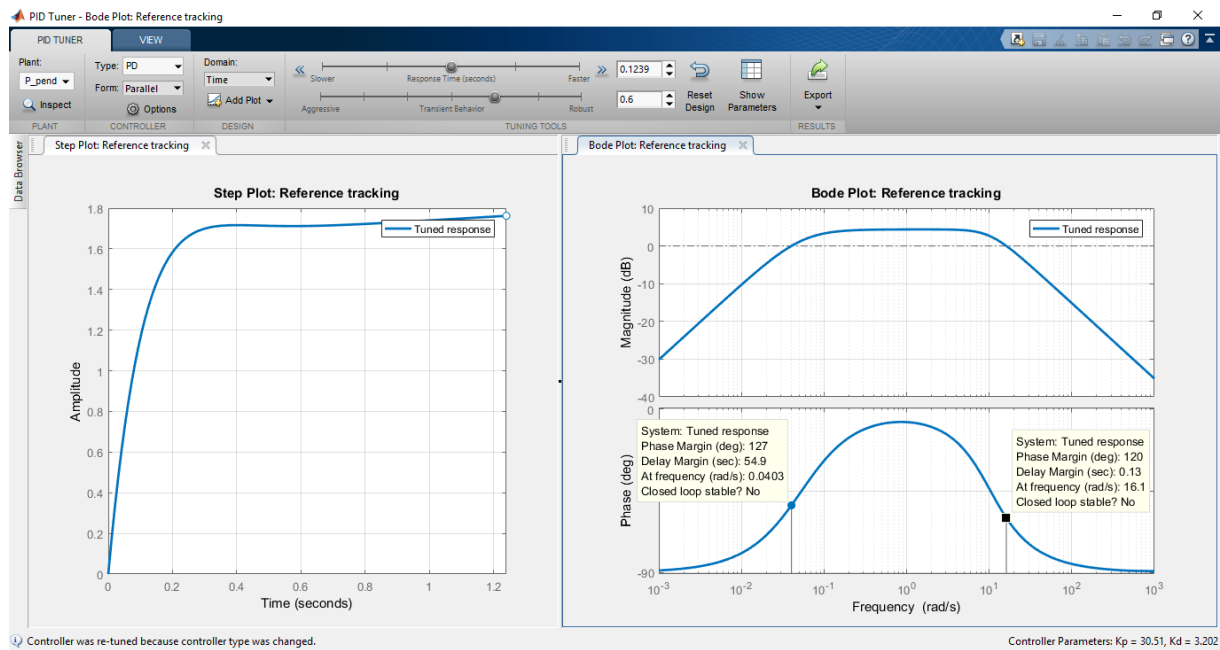
Om de meest geschikte regelaar voor de bekomen transfer functie te vinden wordt er gebruik gemaakt van de PID tuner in Matlab.

De geoptimaliseerde regelaar parameters voor PD en PID regelaar worden naast elkaar gehouden om beiden te vergelijken.

Door middel van de pidtool wordt er gezocht naar de meest optimale regelaar. Dit wordt gedaan door voor elk type regelaar de optimale waarden te berekenen en vervolgens te kijken of het geheel stabiel is in gesloten lus (closed loop stable).

Hiervoor wordt onderstaande code toegevoegd:

```
%%  
Kp = 1; %Versterking P-actie  
Ki = 1; %Versterking I-actie  
Kd = 1; %Versterking D-actie  
  
H=1; %Eenheidsterugkoppeling  
  
C = pid(Kp,Ki,Kd); %regelaar/controller  
T = feedback(C*P_pend,H);  
pidtool(P_pend);
```

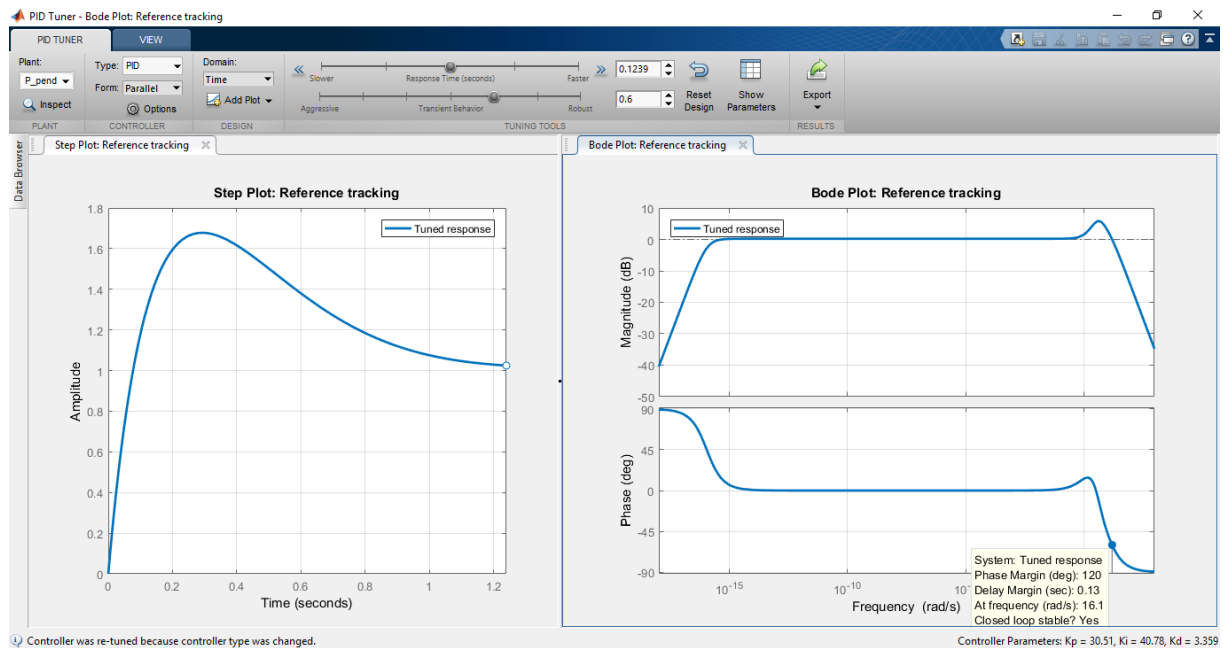


Figuur 5-6 Stap respons & Bode plot voor PD-regelaar

Uit figuur 5-6 valt er af te leiden dat het systeem met een PD-regelaar niet stabiel is in gesloten lus. Dit kan men zien in het bode diagram aan: “closed loop stable? No”.

Controller Parameters	
	Tuned
Kp	11.4026
Ki	n/a
Kd	46.9807
Tf	n/a
Performance and Robustness	
	Tuned
Rise time	NaN seconds
Settling time	NaN seconds
Overshoot	NaN %
Peak	Inf
Gain margin	Inf dB @ NaN rad/s
Phase margin	60 deg @ 0.161 rad/s
Closed-loop stability	Unstable

Figuur 5-7 Gesimuleerde data voor geoptimaliseerde PD-regelaar



Figuur 5-8 Step respons en Bode plot voor geoptimaliseerde PID-regelaar

Uit figuur 5-8 kan er worden afgeleid dat het system met behulp van een PID-regelaar wel stabiel is in gesloten lus.

Hieronder volgen de resultaten die bekomen werden bij een PD-regelaar.

Onderstaande parameters bevestigen de mogelijkheid gebruik te maken van de PID-regelaar in gesloten lus voor de waarden van de opgegeven simulatie.

Controller Parameters	
	Tuned
K_p	30.5134
K_i	40.7823
K_d	3.3587
T_f	n/a
Performance and Robustness	
	Tuned
Rise time	0.0627 seconds
Settling time	1.11 seconds
Overshoot	63.7 %
Peak	1.68
Gain margin	-6.56 dB @ 3.36 rad/s
Phase margin	60 deg @ 16.1 rad/s
Closed-loop stability	Stable

Figuur 5-9 Gesimuleerde data voor een geoptimaliseerde PID-regelaar

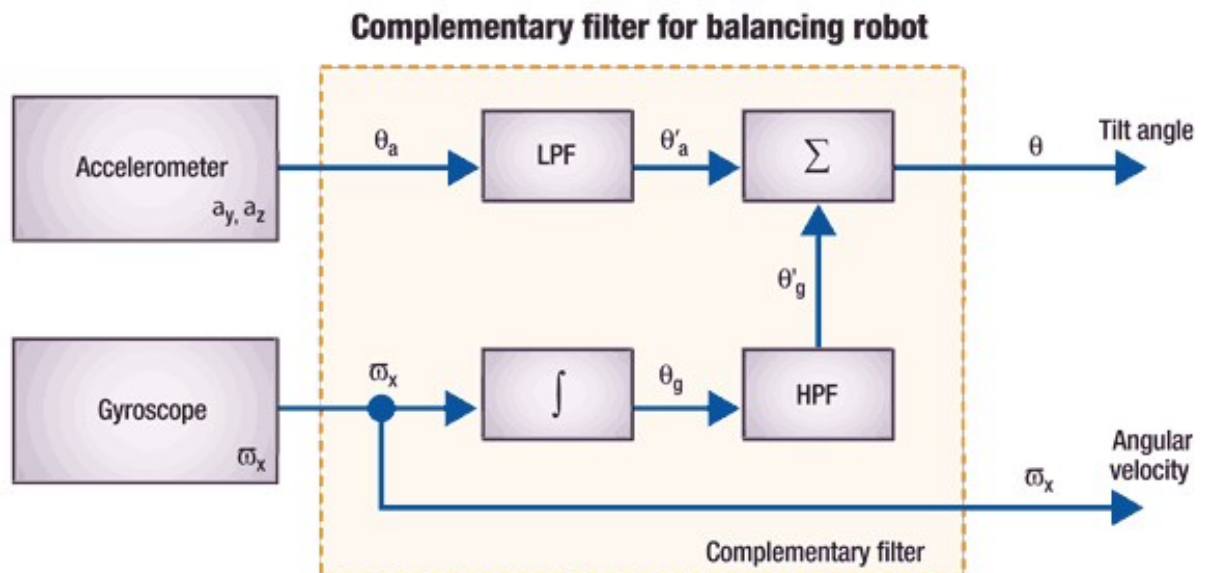
6 Regelaar en filter

6.1 Inleiding

Aangezien het systeem inherent instabiel zal er een regelaar moeten worden gekozen die bijdraagt tot een correcte balancering van de robot. Verder kan er ook worden gekeken naar een filter die het ruis op de signalen van de gyroscoop en accelerometer onderdrukt.

6.2 Algemene blokdiagram

Tijdens het programmeren werden verschillende mogelijkheden geprobeerd er werd er vooral gefocust op het vinden en optimaliseren van een goed werkende PD-regelaar. Onderstaand blokdiagram geeft hier een beeld van weer.



Figuur 6-1 blokschema

6.3 Gekozen regelaar

Er werd gekozen voor een regelaar met P-actie en D-actie. Een I-actie werd niet gebruikt omdat er al van nature een integrerende werking in het systeem aanwezig is. Het gemeten signaal van de gyroscoop in θ/s is al een geïntegreerd signaal.

P-actie

De P-actie wordt altijd gebruikt om het aanwezige foutsignaal u te versterken of te verzwakken. Dit schept de mogelijkheid het systeem sneller te maken maar ook de standfout te verkleinen en de statische nauwkeurigheid te vergroten.

De versterking van een P-actie wordt verbonden aan K_r , hierbij moeten volgende eigenschappen in acht genomen worden:

- Mogelijkheid van een instabiel systeem bij gebruik van een te grote K_r .
- te hevige systemen bij grote K_r waarden
- de standfout kan verkleind worden maar zal nooit helemaal nul zijn
- goede ruisonderdrukking bij grote K_r waarden

D-actie

De D-actie kent een differentiërende werking, dit komt volgens de Laplace theorie overeen met vermenigvuldigen met p . De differentiërende werking schept de mogelijkheid bij te dragen aan een stabiliserende werking. Dit op zich kan een “wispelturig” systeem “rustiger” doen werken. Hierbij moet wel rekening worden gehouden dat t_d niet te klein worden gekozen anders zal dit stabiliserende effect niet te merken zijn. Wanneer er anderzijds wordt gekozen voor een te grote t_d zal de regelaar zich ‘onrustig’ gaan gedragen.

7 Arduino programma

Hieronder volgt het gebruikte arduino programma.

```
#include <SoftwareSerial.h>
SoftwareSerial BT(3, 5);

//Motor A
int ENA = 11; //Snelheidscontrole pwm signaal motor A
int IN1 = 10; //Richting IN1 = 1; IN2 =0 motor A ene richting
int IN2 = 9; //Richting IN1 = 0; IN2 =1 motor A andere richting

//Motor B
int IN3 = 7; //Richting IN3 = 1; IN4 =0 motor B ene richting
int IN4 = 8; //Richting IN3 = 0; IN4 =1 motor B andere richting
int ENB = 6; //Snelheidscontrole pwm signaal motor B

//sensoren
int ACL = A5; //ingang van de accelerometer
int GYR = A0; //ingang van de gyroscoop
int GYRREF = A2; //referentiespanning gyroscoop

//regeling
int Kp = 3; //proportioneel
int Kd = 3; //d-actie
int setacl = 400; //setpoint accelerometer
int v; //snelheid motoren (0-255)

//bluetooth
char a; // stores incoming character from other device
int af = 0; //afwijking voor setpoint voor voorwaarts en achterwaarts rijden
```

```
//cyclustijd
unsigned long tijd = millis();
unsigned long dt; //cyclustijd
void setup()
{
  BT.begin(9600);
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
}
void loop()
{

  if (BT.available()){ // if text arrived in from BT serial...
    a=(BT.read());
    if (a=='0') //select
    {
      af=0;          //zet de afwijking op setacl terug op 0
      BT.println("af = ");
      BT.println(af);
    }
    if (a=='1') //vooruit ^
    {
      af = af + 1;      //laat de afwijking op set acl 1 stijgen
      setacl = setacl + af;
      BT.println(af);
      BT.println(setacl);
    }
    if (a=='2') //achteruit V
    {
      af = af - 1;      //laat de afwijking op set acl 1 dalen
```

```
    setacl = setacl + af;
    BT.println(af);
    BT.println(setacl);
}
if (a=='3') //links <
{
    BT.println("links");
}
if (a=='4') //rechts >
{
    BT.println("rechts");
}
if (a=='5') //[]
{
    Kd = Kd - 1;      //Kd met 1 dalen
    BT.println("Kd = ");
    BT.println(Kd);
}
if (a=='6') //X
{
    Kp = Kp - 1;      //Kp met 1 dalen
    BT.println("Kp = ");
    BT.println(Kp);
}
if (a=='7') //driehoekje
{
    Kp = Kp + 1;      //Kp met 1 stijgen
    BT.println("Kp = ");
    BT.println(Kp);
}
if (a=='8') //O
{
    Kd = Kd + 1;      //Kd met 1 stijgen
    BT.println("Kd = ");
}
```

```
    BT.println(Kd);
}
if (a=='9') //start
{
    BT.println("start");
}
}
int acl = analogRead(ACL)- setacl;
int gyr = analogRead(GYR) - analogRead(GYRREF);
v = acl * Kp + gyr * Kd;    //output met p-factor op hoek en d factor op hoeksnelheid
//// v= acl *Kp; //Gewone P deed het goed //(2)
//v=(0,8 *(gyr*dt) +0,2*(acl)) * Kp; //(3)
if(v>0){
    digitalWrite(IN1, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN4, HIGH);
}
if(v<0){
    digitalWrite(IN1, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN4, LOW);
}
if (abs(v)<10){    //minimum spanning van de motor, anders draait hij toch niet
    analogWrite(ENA, 10);
    analogWrite(ENB, 10);
}
else {
    analogWrite(ENA, abs(v));
    analogWrite(ENB, abs(v));
}
dt = (millis() - tijd);
}
```

8 Interface met Arduino Bluetooth Controller

Voor gemakkelijke bediening werd er gekeken naar een bluetooth app die in staat was gemakkelijk te installeren op bluetooth devices zoals gsm's en tablets. De app is in staat met behulp van een bluetooth apparaat zoals een gsm of tablet verbinding te maken met de bluetooth connector van de robot. Hiermee kunnen er verscheidene commando's worden verstuurd naar de robot. Deze zullen verwerkt en uitgevoerd worden door het programma van de Arduino.



Figure 8-1 Arduino Bluetooth controller interface scherm

Link om app te downloaden:

<https://play.google.com/store/apps/details?id=com.giumig.apps.bluetoothserialmonitor&hl=nl>

9 Conclusies

9.1 Motor

Aanvankelijk werd er gekozen voor twee RH158.12.250 DC-motoren die al voor handen waren in Acro. Deze beschikten over veel koppel maar het lage toerental perkte de recovery angle enorm in. Het lage toerental zorgde dat de motor niet snel genoeg reageerde om zichzelf te balanceren. Hierdoor werd er onder andere gekeken naar oplossingen zoals wielen met grotere diameter. Achteraf gezien werd er beter eerder naar betere motoren gekeken met hoger rendement gewerkt te hebben als groep.

RH158.12.250

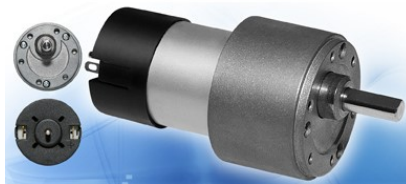


Figure 9-1 RH158.12.250

Vervolgens werd de H29-3HA125-RoH6 geprobeerd. Dit was ook een DC-motor die voor handen was in Acro maar met meer snelheid. Helaas was het houdkoppel van deze motor te laag waardoor deze niet in staat was de robot te balanceren.

H29-3HA125-RoH6



Figure 9-2 H29-3HA125-RoH6

De 3^e motor beschikte over het nodige toerental en het koppel vereist om de motor te balanceren. Deze motoren hebben aldus geen negatieve impact op de recovery angle. Sterker nog deze motor heeft voldoende koppel om de robot op te heffen uit liggende positie en vervolgens verder te balanceren.

RB350018-2A723R 12 V/DC



Figuur 9-3 RB350018-2A723R 12 V/DC

9.2 Motordriver

Over het algemeen was de motordriver zeer geschikt voor de toepassing. Het enige nadeel dat ondervonden werd, waren losse aansluitblokjes. Dit kon zorgen voor slecht contact en bijgevolg een gebrekkige respons van de motors. Dit werd verholpen door deze aansluitblokjes zelf vaster te bevestigen.

9.3 Matlab

Uiteindelijk blijft de matlab simulatie net dat, een simulatie. Hiermee wordt bedoeld dat deze ingesteld wordt aan de hand van parameters die kunnen afwijken ten opzichte van de realiteit en mogelijk zelfs geen rekening houdt met parameters die er bij de werkelijke robot wel aanwezig zijn. Hierdoor is volgende stelling belangrijk, het gezond verstand moet primeren en een simulatie alleen geeft geen uitsluitsel tegenover de werkelijkheid. Met PD-regelaar was het mogelijk de robot te balanceren met behulp van wielen met grotere diameter en ondergrond die een grotere rolweerstand bood. Dit getuigt echter nog niet van een stabiele werking van het regelsysteem. Daar de PD-regelaar nog kan worden geoptimaliseerd kan de uitkomst van Matlab waarin een PD-regelaar een minder goede optie zou zijn als een PID-regelaar niet ondersteund worden. Wel wijst het testen van de robot met het gebruikte programma erop dat een goed afgestelde PD-regelaar in staat zou moeten zijn de robot voor langere termijn te balanceren met een grote recovery angle.

9.4 Labview

Het Labview programma liet toe de sensorwaarden vlot in te lezen. Dit gaf een goed beeld van hun werking en vergemakkelijkte de instelling van de regelparameters in het programma.

9.5 Constructie

Over het algemeen liet de modulaire opbouw van de robot toe gemakkelijk onderdelen te veranderen en te verplaatsen. De robot is geconstrueerd met het doel zo overzichtelijk en eenvoudig mogelijk te zijn.

9.5.1 Gewicht

Er werd ondervonden dat hoe lichter de robot, des te makkelijker deze gebalanceerd kan worden. Dit wordt vooral duidelijker wanneer men kijkt naar paragraaf 2.5 waarin de formule van het koppel gebruikt wordt. Hierin heeft een grotere massa een nadelig effect op de recovery angle.

9.5.2 Grotere wielen

Grotere wielen zijn een constructieve keuze. Deze hebben het voordeel meer moment te realiseren op de robot maar tegelijkertijd helaas ook massa toe te voegen aan het geheel. Het grotere traagheidsmoment droeg wel sterk bij tot het bekomen van een betere balancering.

9.5.3 Arduino Uno Shield

De Arduino Uno Shield creëerde de mogelijkheid de sensoren en bluetooth module op compacte en overzichtelijke wijze te monteren. Achteraf bekeken was het wel praktischer geweest om de sensoren met headsockets op de shield te bevestigen in plaats van deze vast te solderen.

9.5.4 Dipswitch

Bij veelvuldig testen werd er gekeken naar een oplossing om de robot van zijn stroombron te onderbreken. Het hoofddoel hiervan was om de batterij optimaal te benutten en terwijl als groep met het hoogst mogelijke rendement te kunnen werken (immers zou veelvuldig opladen nadelig zijn voor het tijdschema van dit project).

9.6 Kwaliteit van de constructie

Hoewel er op de meeste plaatsen handig gebruik wordt gemaakt van kabelhulzen, krimpkousen en zo kort mogelijke kabels was het op enkele plaatsen nog mogelijk geweest kortere kabels te gebruiken. Dit is vooral aan de motorkabels te merken die door de andere motoren in een later stadium van de bouw op maat zijn gemaakt. Wel werd er getracht zo ver mogelijk vooruit te kijken door middel van een modulaire opbouw bij de constructie om ruimte te laten voor veranderingen. Deze modulaire opbouw loonde vooral in de latere stadia waarin er soms kleine snelle aanpassingen vereist waren voor het testen van de robot.

9.7 Taakverdeling en planning

Voor de taakverdeling en de planning wordt er verwezen naar de eerder ingeleverde afstemmingen met de opdrachtgever.

9.8 Kostenplaatje

Hierbij moet worden vermeld dat er bij de constructie gebruik gemaakt werd van materialen die vanuit Acro al ter beschikking waren (accelerometer, gyroscoop & Arduino UNO), ook hadden de studenten bepaalde constructiematerialen (plexiglas, hout en wielbussen) zelf al voor handen. Voorgenoemde componenten worden aldus niet meegerekend.

Naam	Aantal	Kostprijs (Incl. Btw)
Arduino UNO shield	1x	€14,52
Batterij 2400MAH 20C ECO-SERIES LIPPO accu	1x	€24,99
Bluetooth Connector	1x	€14,90
Dual Bridge DC stepper controller board L298N	1x	€9,90
Reely Accu-Adapter XT60-T-stekker	1x	€2,99
	Totaal:	€67,3

Tabel 3 Kostenplaatje

10 Bronvermelding

10.1 Bronvermelding figuren

Figuur 2-1 ARDUINO UNO & benaming aansluitpinnen

<https://www.robomart.com/arduino-uno-online-india>

Figuur 2-2 Accelerometer

<http://uk.rs-online.com/web/p/analogue-development-kits/7946739/>

Figuur 2-3 Gyroscoop

<https://www.sparkfun.com/products/retired/9410>

Figuur 2-4 Montage bordje voor de LN298N

<https://www.gotron.be/l298n-dubbele-h-brug-driver-voor-dc-of-stappenmotor.html>

Figuur 2-5 L298N motor driver

<http://www.youngengineer.in/ics/l298-dual-full-bridge-driver-ic-pack-of-2>

Figuur 2-6 motor

<http://www.conrad.be/ce/nl/product/233131/Modelcraft-Krachtige-transmissiemotor-RB350018-2A723R-12-VDC?ref=list>

Figuur 10-8 Batterij

<http://www.conrad.be/ce/nl/product/1344141/LiPo-accupack-111-V-2400-mAh-20-C-Conrad-energy-Stick-XT60?ref=searchDetail>

Figuur 6-1 blokschema

https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2015/dc686_nn233_hz263/final_project_webpage_v2/dc686_nn233_hz263/

Figuur 9-1 RH158.12.250

http://www.micromotors.eu/gear_motor_rh158.html

Figuur 9-3 RB350018-2A723R 12 V/DC

<http://www.conrad.be/ce/nl/product/233131/Modelcraft-Krachtige-transmissiemotor-RB350018-2A723R-12-VDC?ref=list>

10.2 Literaire bronnen

Formule maximaal koppel

[1] " $T_{max} = m \cdot g \cdot l \cdot \sin \theta_{max}$ " Modeling, Simulation and Fabrication of a Balancing Robot –by: Ye Ding, Joshua Gafford, Mia Kunio –From: Harvard University, Massachusetts Institute of Technology

Modelering verdelen in cart en pendulum

Two-Wheeled Self-Balancing Robot

Design and control based on the concept of an inverted pendulum

-by: Hellman Hanna, Sunnerman Henrik –From: KTH Stockholm Zweden

11 Datasheets

Arduino Uno

<https://www.farnell.com/datasheets/1682209.pdf>

Accelerometer

<http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL335.pdf>

Gyroscoop

<https://www.cdiweb.com/datasheets/invensense/PS-IXZ-0500B-00-03.pdf>

Motordriver

<https://www.gotron.be/l298n-dubbele-h-brug-driver-voor-dc-of-stappenmotor.html>

datasheet L298N (van motordriver)

<http://www.alldatasheet.com/datasheet-pdf/pdf/22440/STMICROELECTRONICS/L298N.html>

Motor

http://www.produktinfo.conrad.com/datenblaetter/225000-249999/233131-da-01-en-HOCHLEISTUNGSGETRIEBEMOTOR_12V_DC_1_18.pdf