

**Balancerende Robot**

Mentor: Mr. Habraken

Gerealiseerd door Karel De Smet

Derde graad TSO – Elektriciteit-Elektronica

Schooljaar 2015-2016

****

**Balancerende Robot**

Mentor: Mr. Habraken

Gerealiseerd door Karel De Smet

Derde graad TSO – Elektriciteit-Elektronica

Schooljaar 2015-2016

# Voorwoord

De geïntegreerde proef is een jaarproject en is een reflectie van de manier waarop en hoe je te werk gaat. Dit project was voor mij een boeiende uitdaging waar ik met plezier, en soms ook wel frustratie aan gewerkt heb.

Ik wil graag een aantal mensen bedanken voor hun hulp. In de eerste plaats mijn mentor, Dave Habraken, voor zijn hulp en ondersteuning gedurende het hele project. Daarnaast ook Jos Lodewyckx, de mentorcoördinator, en mijn medeleerlingen. Ook mijn ouders, voor de mentale ondersteuning en zeker niet te vergeten mijn vriendin Tosia voor haar niet aflatende steun.

Karel

inhoudstafel

[Voorwoord 4](#_Toc449551274)

[1 Inleiding en opdrachtomschrijving 6](#_Toc449551275)

[1.1 Inleiding 6](#_Toc449551276)

[1.2 Opdrachtomschrijving 6](#_Toc449551277)

[2 Planning 7](#_Toc449551278)

[3 Het concept 8](#_Toc449551279)

[3.1 IMU 8](#_Toc449551280)

[3.2 Ultrasone sensoren 8](#_Toc449551281)

[3.3 IMU en ultrasone sensoren 9](#_Toc449551282)

[3.4 eventuele uitbreidingen 10](#_Toc449551283)

[3.5 Besluit 11](#_Toc449551284)

[4 Realisering van het ontwerp/concept 12](#_Toc449551285)

[4.1 Blokschema 12](#_Toc449551286)

[4.1.1 De Arduino UNO 12](#_Toc449551287)

[4.1.2 Motoren 13](#_Toc449551288)

[4.1.3 De IMU met gyroscoop en accelerometer 13](#_Toc449551289)

[4.1.4 Ultrasone sensoren 13](#_Toc449551290)

[4.1.5 Bluetooth® 13](#_Toc449551291)

[4.1.6 Het besturingsprogramma 13](#_Toc449551292)

[4.1.7 6V batterij 14](#_Toc449551293)

[5 Keuze van de componenten 15](#_Toc449551294)

[5.1 de IMU 15](#_Toc449551295)

[5.1.1 De werking van de accelerometer 15](#_Toc449551296)

[5.1.2 I2C, het verzenden van de data 17](#_Toc449551297)

[5.1.3 Technische aspecten van de IMU 17](#_Toc449551298)

[5.2 Motoren 18](#_Toc449551299)

[5.2.1 Servomotoren 18](#_Toc449551300)

[5.2.2 Stappenmotoren 19](#_Toc449551301)

[5.2.3 28BYJ-48 met de ULN2003 driver 20](#_Toc449551302)

[5.2.4 TS3079N535 20](#_Toc449551303)

[5.2.5 Besluit 21](#_Toc449551304)

[5.2.6 Elektrisch Schema 22](#_Toc449551305)

[6 De behuizing 23](#_Toc449551306)

[6.1 Schets 23](#_Toc449551307)

[6.2 Realisatie 23](#_Toc449551308)

[7 Werking 24](#_Toc449551309)

[7.1 GY-521 en de MPU6050 24](#_Toc449551310)

[7.1.1 De gyroscoop gebruiken 24](#_Toc449551311)

[7.1.2 De accelerometer gebruiken. 25](#_Toc449551312)

[7.2 Het programma en de flowchart 26](#_Toc449551313)

[7.2.1 Flowchart: Hoofdprogramma 26](#_Toc449551314)

[7.2.2 Uitleg functie initialiseren 28](#_Toc449551315)

[7.2.3 Uitleg functiemotor aansturen: 29](#_Toc449551316)

[7.2.4 Regelaars 29](#_Toc449551317)

[8 Kostprijs 35](#_Toc449551318)

[9 besluit 36](#_Toc449551319)

[10 bijlagen 37](#_Toc449551320)

[10.1 bronnenlijst 37](#_Toc449551321)

[10.2 Belangrijke datasheets 38](#_Toc449551322)

# Inleiding en opdrachtomschrijving

## Inleiding

Uitdagende software schrijven heeft me altijd al geboeid. Om mijn GIP te creëren heb ik mijn elektronische kennis van de afgelopen 2 jaar en mijn softwarekennis  
gebundeld om zo tot een boeiend resultaat te komen.

## Opdrachtomschrijving

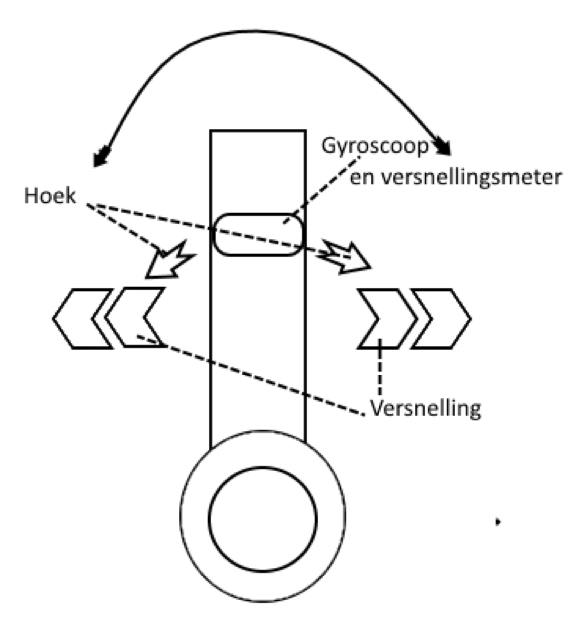
Mijn GIP is een robotje dat op 2 wielen kan blijven staan. De 2 wielen zijn verbonden aan 2 motoren die de wielen doen bewegen en ervoor zorgen dat het kan blijven staan. Om te weten hoe de motoren moeten reageren om te kunnen blijven rechtstaan, moeten er metingen worden gemaakt om de positie te bepalen. Deze metingen komen oorspronkelijk van de IMU (Inertial measurement unit). De IMU is een elektronische component die specifieke krachten, hoeken en soms magnetische velden meet en rapporteert door gebruik van snelheidsmeters (accelerometer), gyroscopen en soms magnetometers. De gekregen waarden van de IMU zijn nog niet bruikbaar om de positie te bepalen. Ze moeten nog omgerekend worden door middel van de complementaire filter. Deze filter kan door middel van alle assen van zowel de gyroscoop als de accelerometer een roll, pitch en yaw waarde creëren. De filter zorgt ook voor het minimaliseren van alle soorten storingen. Om het proces van het robotje zo efficiënt mogelijk te laten lopen, wordt gebruik gemaakt van een PID-regelaar. De PID-regelaar kijkt naar de versnelling en de hoek van het robotje en regelt de robot door de foutmarge in functie van de tijd te meten in een proportionele, een integrerende en een afgeleide functie. De programma’s en alle componenten zijn geconnecteerd aan de Arduino UNO.

# Planning

|  |  |  |
| --- | --- | --- |
| **EINDDATUM** | **PLANNING** | **Aangepaste datum** |
| 11-09 | Opdrachtomschrijving |  |
| 18-09 | Vergelijking van 2 mogelijke oplossingen |  |
| 22-09 | Blokschema |  |
| 28-09 | Presentatie + persoonlijke planning |  |
| 2-10 | Verbeterde versies vergelijking 2 concepten en blokschema |  |
| 8-11 | Bouwstuk af |  |
| 12-10 | Inleveren elektrische/elektronische schema’s |  |
| 19-10 | Functionele analyse en flowchart |  |
| 22-11 | Alles uitzoeken hoe de gyroscoop werkt |  |
|  | EXAMENPERIODE |  |
| 28-12 | Flowchart verbeteren |  |
| 02-01 | Elektrisch schema verbeteren |  |
| 05-01 | Lay-out en teksten nakijken van scriptie |  |
| 10-01 | Flowchart omzetten in programma | 12-01 |
| 14-01 | Programma van sensoren laten werken | 16-01 |
| 17-01 | Programma van de stappenmotoren laten werken | 21-01 |
| 24-01 | Programma in totaal laten werken + wielen aansluiten |  |
| 09-02 | Nieuwe stappenmotoren gekocht en een nieuwe constructie gemaakt. |  |
| 17-02 | Stappenmotoren laten draaien en in het programma maken |  |
| 19-02 | Scriptie verbeterd |  |
| 24-02 | Scriptie verbeterd |  |
| 27-02 | Scriptie verbeterd |  |
| 29-02 | Scriptie verbeterd |  |
| 01-03 | Scriptie verbeterd |  |
| 13-03 | Scriptie verbeterd |  |
|  | Examenperiode |  |
| 24-03 | PID-regelaar programmeren |  |
| 25-03 | PID-regelaar programmeren |  |
| 26-03 | PID-regelaar programmeren |  |
| 27-03 | PID-regelaar programmeren |  |
| 30-03 | PID-regelaar in scriptie toevoegen |  |
| 31-03 | PID-regelaar in scriptie toevoegen |  |
| 01-04 | PID-regelaar in scriptie toevoegen |  |
| 02-04 | Nieuwe servomotoren gekocht |  |
| 19-04 | Inleiding, opdrachtomschrijving en voorwoord verbeteren |  |
| 27-04 | Verbetering in grammatica en Nederlandse taal |  |
| 30-04 | Opendeurdag |  |

# Het concept

## IMU

Het idee om een IMU te gebruiken is het meest gebruikte idee op het internet voor een balancerende robot. Het gebruik van 2 wielen zorgt ervoor dat je de robot alleen moet corrigeren in 1 as. Dat maakt het gemakkelijker om de robot stabiel te houden. De IMU stuurt de signalen van de gyroscoop en de accelerometer naar de Arduino UNO. Daar worden de signalen omgezet naar bruikbare informatie voor de motoren. Op die manier worden de bewegingen van de robot gecorrigeerd

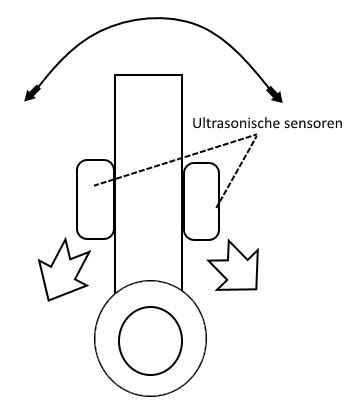
VOORDELEN:

* veel voorbeelden online te vinden
* gemakkelijk controleerbaar

NADELEN:

* de waarden van de IMU moeten omgezet en gefilterd worden wat een relatief lange rekentijd vraagt

## Ultrasone sensoren

In dit idee wordt de IMU vervangen door ultrasone sensoren. Er wordt 1 sensor per kant geplaatst. Ze zijn steeds naar de grond gericht en ze zorgen ervoor dat de Arduino UNO de nodige informatie krijgt om de hoogte tussen de grond en de sensor van de beide kanten te berekenen. Zo kan de Arduino UNO de nodige signalen naar de motoren sturen om mooi te kunnen balanceren.

VOORDELEN:

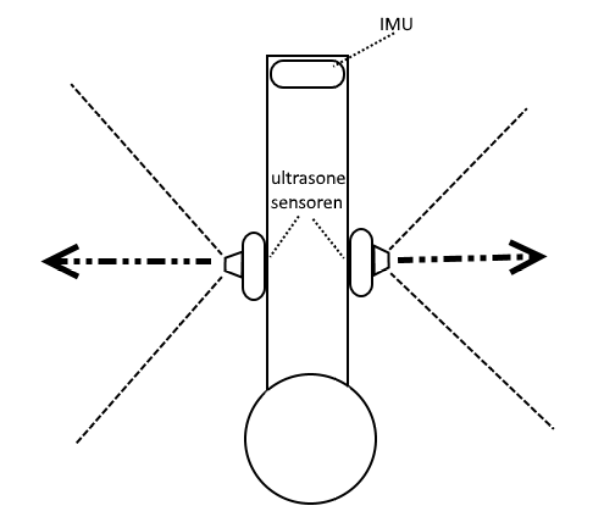
* ultrasone sensoren zijn goedkoper
* het is mogelijk om een automatische noodstop te maken

NADELEN:

* het is moeilijker om het onevenwicht te meten
* weinig voorbeelden online
* kan niet gebruikt worden op een helling

## IMU en ultrasone sensoren

Dit idee gebruikt een combinatie van beide vorige ideeën. De IMU wordt gebruikt om de huidig hoek te bepalen zodat de robot recht blijft staan. De ultrasone sensoren worden hier gebruikt als veiligheidssensor, niet als extra meting. Dankzij de 2 ultrasone sensoren, 1 aan elke kant, weet de robot dat er zich een object in zijn buurt bevindt en kan hij daarmee rekening houden. Dit kan gebruikt worden in de volgende voorbeelden:

* Als de robot rechtdoor rijdt en een muur tegenkomt: dankzij de sensoren weet de robot dat hij moet stoppen om een botsing te vermijden.
* Als er iemand een stoot geeft om te testen of de robot wel degelijk zelf balanceert: dankzij deze sensoren weet de robot langs welke kant het duwtje komt en kan het in de andere richting rijden om te balanceren en te voorkomen dat hij omvalt.

VOORDELEN:

* veiliger
* meer kennis van de omgeving

NADELEN:

* vertraging in het programma: de tijd dat het signaal terugkeert

## eventuele uitbreidingen

Pas nadat alles is afgewerkt zal er een uitbreiding kunnen gemaakt worden.

Een besturingssysteem: met de pc zal je waarden kunnen doorsturen naar de robot via Bluetooth. Deze waarden bepalen wat de robot doet en hoe hij reageert. Dit kan ook een eenvoudige afstandsbediening zijn waarmee je de robot bestuurt.

## Besluit

Het juiste concept kiezen is niet eenvoudig. Uiteindelijk, met dank aan mijn mentor, zijn we tot een oplossing gekomen. Er wordt in fases gewerkt.

Dit wil zeggen dat ik begin met het realiseren van het eerste idee en zo bouw ik op naar het idee met de bluetoothverbinding.

Dit zijn de stappen die ik zal nemen:

Fase 1:

In fase 1 wordt de basis van de robot gemaakt. Hiervoor gebruik ik de motoren en de gyroscoop. Zo probeer ik de robot te laten balanceren.

Fase 2:

In fase 2 voeg ik de ultrasone sensoren toe aan de basis zodat de robot veiliger kan rondrijden.

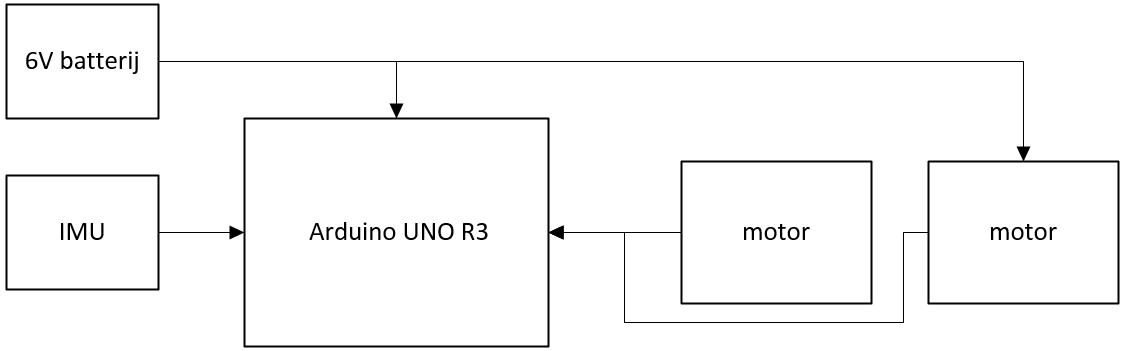
Fase 3:

In fase 3 wordt een besturingsprogramma gemaakt zodat de robot kan bestuurd kan worden.

Na het uitvoeren van de eerste fase zou de robot reeds moeten kunnen balanceren.

# Realisering van het ontwerp/concept

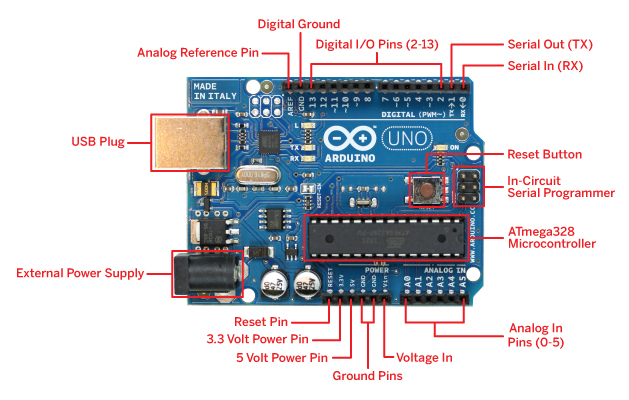
## Blokschema



### De Arduino UNO

De Arduino UNO is een open source elektronisch platform dat te gebruiken is voor zowel hardware als software. Het is het brein van de robot. Alle sensoren zijn eraan aangesloten. Een programma op de Arduino UNO gaat het mogelijk maken dat alle sensoren tegelijkertijd samen kunnen werken en de robot zal laten balanceren.

De Arduino UNO heeft 14 digitale input/output pinnen, 6 analoge pinnen, een 3,3V pin, een 5V pin, een voltage ingang pin en 3 ground pinnen.



### Motoren

De motoren doen al het werk en zorgen ervoor dat de zelf balancerende robot recht blijft te staan. De motoren gaan sneller draaien naarmate de afwijkende hoek verder weg ligt dan 90°. De motoren moeten krachtig genoeg zijn om de hele constructie mee te nemen.

### De IMU met gyroscoop en accelerometer

De gyroscoop en de accelerometer zitten samen in 1 module, de IMU.

#### De gyroscoop:

De gyroscoop geeft de nodige informatie door aan de Arduino UNO om te blijven balanceren. Hij geeft de hoeksnelheid in aantal graden per seconde. In elk van de 3 assen geeft de gyroscoop het aantal graden per seconde weer. Dit wordt omgerekend naar de positie van de robot.

#### De accelerometer:

De accelerometer geeft de versnelling weer. Voor dit project hebben we de versnelling nodig om te vermijden dat de robot in onevenwicht komt en om te bepalen met welke versnelling het onevenwicht ontstaat. Dit is cruciaal want de robot heeft een bepaalde massa en hoe sneller er een onevenwicht ontstaat, des te meer de stappenmotor moet corrigeren.

### Ultrasone sensoren

De ultrasone sensoren worden gebruikt om de afstand te berekenen tussen de robot en de grond. Dit is enkel nodig op een helling. Terwijl de gyroscoop de positie meet van de robot ten opzichte van de grond gaan ultrasone sensoren zich bezighouden met het berekenen van de hoeken van de helling. Er zijn 2 sensoren aan elke kant die elkaar controleren.

### Bluetooth®

De Bluetoothconnectie zorgt voor de communicatie tussen het besturingsprogramma en de Arduino UNO. Hierdoor kan de robot draadloos bestuurd worden.

### Het besturingsprogramma

Het besturingsprogramma staat op een Android app of een computer en zorgt ervoor dat de robot op afstand kan bestuurd worden via de Bluetoothconnectie.

### 6V batterij

De voeding van de motoren en de Arduino UNO wordt door de 4 1,5V batterijen gevoed. De 2 motoren zullen van 5V tot 12V werken en de Arduino UNO zal van 6V tot 18V werken.

# Keuze van de componenten

Voor alle componenten ben ik nagegaan welke keuze de beste is voor mijn project. Daarvoor heb ik meerdere bronnen geraadpleegd.

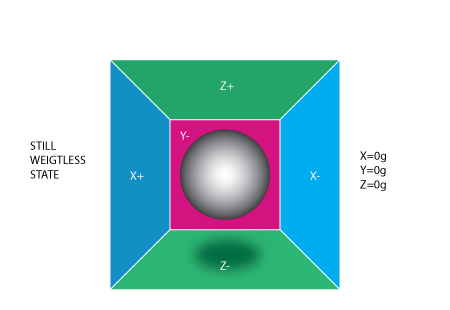
## de IMU

IMU is een afkorting en staat voor ‘Inertial measurement unit’. De IMU hoort ook tot een van de MEMS-sensoren (‘Micro-Electro-Mechanical Systems’).

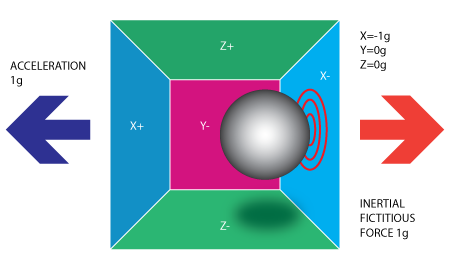
Mijn IMU, de GY-521, is een zeer bekende en goedkope IMU. Dankzij de MPU-6050 chip heeft hij een 6DOF (6 graden van vrijheid), dat betekent dat 3 assen voor de gyroscoop zijn en de andere 3 voor de accelerometer.

### De werking van de accelerometer

Een accelerometer zou je kunnen voorstellen als een doos met een bal in. Als zich op deze doos geen zwaartekracht bevindt, zweeft de bal in het midden zoals op de illustratie. Y+ is verwijderd zodat je in de doos kan zien.

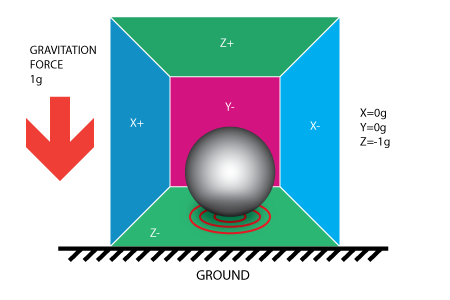


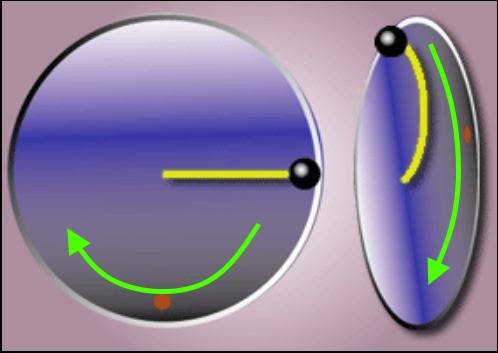
Wanneer er zwaartekracht zou optreden met bv. X= -1g, dit is equivalent met x= -9.81m/s², kunnen we hieruit afleiden dat er een versnelling is van x = 1g (9.81m/s²) (zoals op de onderstaande foto geïllustreerd).



In een voorbeeld zonder zwaartekracht is enkel de inertiale kracht een parameter die de acceleratie aanpast. Als we dit voorbeeld op aarde zouden zetten, zou de bal helemaal anders reageren. De parameters die er dan mee zouden werken zijn de zwaartekracht, de inertiaalkracht, de centrifugale kracht en de corioliskracht.

Aspecten die de metingen van de IMU minder accuraat maken:

* **De zwaartekracht** of gravitatie is een aantrekkende kracht die twee massa's op elkaar uitoefenen. Op aarde dit heeft een waarde van 9.812m/s² (formule: waar m = massa, g = 9,812m/s²)
* **De inertiaalkracht** of de schijnkracht of traagheidskracht is een kracht die wordt waargenomen als een beweging vanuit een coördinatenstelsel wordt bekeken, dat zelf aan een versnelling onderhevig is.
* **De centrifugale kracht** of middelpuntvliedende kracht is een reactiekracht uitgeoefend op een lichaam die dit doet afbuigen. De kracht kan worden voorgesteld als een vector met een richting van het rotatiecentrum naar het voorwerp.
* **De drift** is de offset die zowel voorkomt in de gyroscoop als in de accelerometer. De waarde van de offset is bij elke sensor anders. Dit komt door de fysieke onnauwkeurigheid van elke sensor. De specifieke oorzaak van deze offset is steeds anders. Er zijn veel parameters dat hierin meespelen, zoals: een interne vervorming, een interne verschuiving, … .
* **De corioliskracht** is de verklaring voor de afbuiging van de baan van een voorwerp dat beweegt binnen een roterend systeem, in dit geval de aarde.



### I2C, het verzenden van de data

I2C is een synchrone, seriële bus. Deze wordt gebruikt voor datacommunicatie tussen de Arduino UNO en de IMU. I2C werkt op 2 buslijnen, SDA (seriële data) en SCL (seriële klok). Over de SDA wordt data verzonden met het bijpassende kloksignaal dat via de SCL-lijn komt. Beide lijnen zijn bidirectioneel.

[](https://en.wikipedia.org/wiki/File:I2C_data_transfer.svg)

De werking van I2C dataoverdracht:

1. S (start) is het signaal dat aan de SDA wordt gegeven om omlaag getrokken te worden zodat de dataoverdracht kan beginnen. De SCL blijft laag.
2. In het blauwe vakje wordt de SCL laag gehouden terwijl de SDA de eerste databit gelijk zet. De data wordt ontvangen op het groene vakje terwijl SCL continu hoog gehouden wordt.
3. P (stop) is het signaal dat aangeeft dat de dataoverdracht gedaan is. Hierbij wordt SDA omhoog getrokken tot in zijn ruststand. De ruststand is wanneer I2C niet actief is. SDA en SCL staan dan allebei hoog.

### Technische aspecten van de IMU

De communicatie van de IMU werkt via een I2C bus.

Er is zeer veel documentatie over de MPU6050 die op de gy521 zit.

De GY-521 bezit 8 pinnen:

Vcc: +3,3V aanbevolen maar kan tot 5V werken.

GND: de ground

SCL: seriële klok

SDA: seriële data

XDA: Aux data

XCL: Aux Klok

ADO: Analog/Digital Output

INT: Interrupt pin

## Motoren

### Servomotoren

De werking van een servomotor is zeer eenvoudig. Dit maakt hen ook zeer betrouwbaar. Er zijn verschillende fysieke delen in een servomotor: een gelijkstroommotor, tandwielen en een printplaatje met een positiesensor.

Omdat een kleine gelijkstroommotor een groot toerental heeft maar weinig koppel, kan dit toerental worden gereduceerd en het koppel worden versterkt.

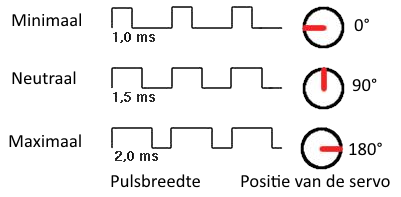
In het printplaatje is een positiesensor geïntegreerd. Dankzij een feedback loopgeef je een bepaalde hoek. De feedbackloop is eenvoudig principe. De positie van de motor wordt constant gemeten. Terwijl de motor draait bepaalt de positiesensor of de motor “verder” of “terug” gaat, of “stopt”.

Er zijn 2 verschillende servomotoren:

* Positionele rotatie: de meeste populaire van de 2 draait rond zijn as in ongeveer 2 helften van 180°. Als de gegeven hoek onder -90° of boven 90° is, zal dit niet gaan omdat er fysieke stoppunten geplaatst zijn. Om de motor te besturen wordt een hoek opgegeven, en de tijd waarin deze hoek genomen moet worden.
* Continue rotatie: gelijkaardig aan de positionele rotatie, maar de motor kan voor onbepaalde tijd in beide richtingen kan draaien. Om de motor te besturen wordt eerder een stuursignaal gebruikt dan een hoeksignaal om de richting en de snelheid te interpreteren.

Om een servomotor te kunnen besturen, wordt gebruik gemaakt van het PWM-principe: pulse width modulation of puls breedte modulatie.

Bij de positionele rotatie is de besturing van de servomotor als volgt:

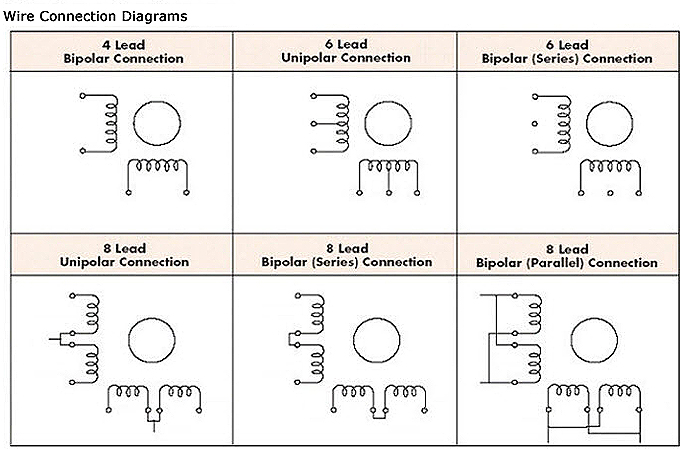


Hoe groter de pulsbreedte, des te groter de hoek zal zijn. Als je constant een waarde doorgeeft, bv. 2,0ms, zal de hoek constant blijven, nl. 180°. De kracht van de motor blijft ook constant omdat de motor de hele tijd wordt aangestuurd.

### Stappenmotoren

Stappenmotoren werken zeer precies. Men kan een stappenmotor exact aansturen door de spoelen aan te sturen. Zo kan je aansturen om precieze hoeken te draaien.

Stappenmotoren komen in verschillende soorten voor:



De verschillen tussen de **bipolaire** en de **unipolaire** stappenmotoren zijn:

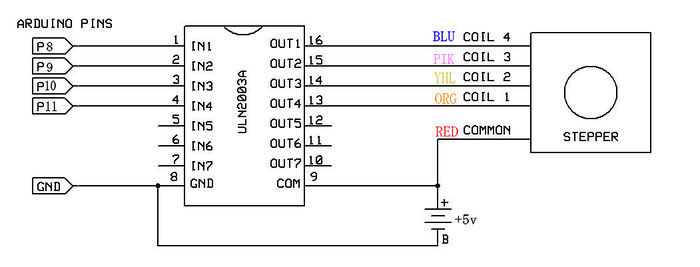
|  |  |
| --- | --- |
| **UNIPOLAIR** | **BIPOLAIR** |
| Deze soort stappenmotoren hebben **twee windingen per fase**, één voor elke richting van het magnetisch veld.  In deze opstelling kan een magnetische pool worden omgekeerd zonder eerst de stroomrichting te keren.  De commutatieschakeling kan zeer eenvoudig worden gemaakt (bv. één enkele transistor) per winding. | Bipolaire motoren hebben **een enkele winding per fase**.  Er zijn twee kabels per fase, er zijn geen gemeenschappelijke.  Omdat windingen beter worden benut, zijn ze krachtiger dan bij een unipolaire motor met hetzelfde gewicht. Dit komt door de fysieke ruimte die ingenomen wordt door de wikkelingen. |

Een unipolaire motor heeft tweemaal de hoeveelheid draad in dezelfde ruimte, maar gebruikt op elk moment slechts de helft, en is dus 50% minder efficiënt (of ongeveer 70% van het beschikbare koppel).

### 28BYJ-48 met de ULN2003 driver

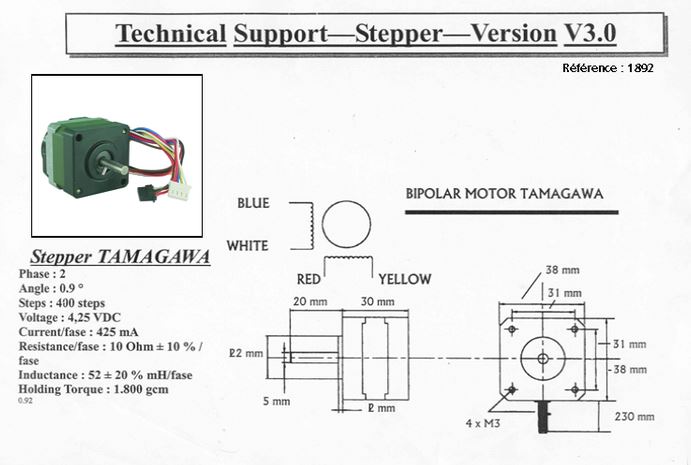
28BYJ-48

Deze stappenmotor is een unipolaire stappenmotor met 5 geleiders (2 voor 2 spoelen en 1 voor de spanning). De motor is zeer compact en kan 5,625° per stap maken. Met andere woorden, in 64 stappen is de motor 360° graden gedraaid. Een groot nadeel is de snelheid. Na het testen is ondervonden dat de maximale snelheid niet genoeg was om de robot uit een situatie met een grote snelheid en een grote hoek te halen.

ULN2003

De drivers zijn nodig om de stroom te scheiden van de Arduino UNO en van de batterij. Aan de IN5 tot IN7 kan men ook weerstanden plaatsen en die gebruiken als stappen van de stappenmotor zodat de stappen nog preciezer zijn.

### TS3079N535

Deze stappenmotor is een 2-fasen bipolaire stappenmotor. Het voordeel van deze stappenmotor is het gewicht. Dankzij het gewicht kan de robot beter balanceren doordat het zwaartepunt sterk gedaald is. Er wordt geen drivers gebruikt omdat dit niet nodig is. Alles kan rechtstreeks worden verbonden met de Arduino UNO. De typerende spanning is volgens de datasheet 4,25 VDC. Dit is ideaal voor het aansluiten op de Arduino UNO omdat de Arduino UNO een uitgangsspanning heeft van iets minder dan 5V. Het aantal graden per stap is ook zeer nauwkeurig, nl. 0,9°. Dit maakt dat zelfs zeer kleine bewegingen kunnen worden bijgestuurd. Om 1 volledige toer te maken moeten 400 tappen worden gemaakt. Dit zal nooit het geval zijn in mijn toepassing.

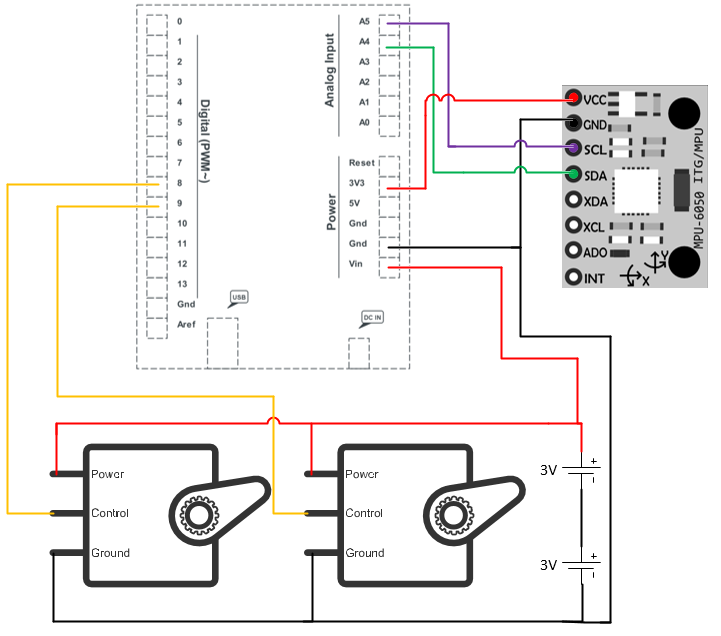
.

### Besluit

Alle bovenstaande motoren zijn uitgetest.

* 28BYJ-48:
  + voordelen
    - kleine motor, niet veel plaats nodig
    - precieze werking
    - niet zwaar
  + nadelen
    - 6 draden per de driver én 5 draden per de motor
    - niet krachtig bij grote snelheden
    - niet snel
* TS3079N535
  + voordelen
    - zware motor, goed voor tegengewicht
    - precieze werking
    - snel
    - werkt op 4.25 VDC
  + nadelen
    - 4 draden per motor
    - groot, neemt veel plaats in
    - niet krachtig bij grote snelheden
* ULM300
  + voordelen
    - 3 draden per motor
    - zeer krachtig
    - minder lijnen code
    - kan grote snelheden halen
  + nadelen
    - minder precies
    - niet zwaar

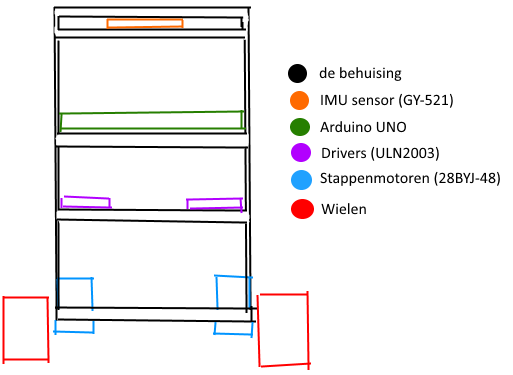
### Elektrisch Schema



# De behuizing

## Schets

De balancerende robot moet een praktisch behuizing hebben. Op mijn eerste schets is te zien dat alle componenten in lagen zijn gezet en samen met de behuizing een toren vormen.



## Realisatie

* **1ste constructie:** Alle horizontale plaatjes zoals in de schets werden gerealiseerd in hout (behalve het bovenste plaatje). Om alle plaatjes aan elkaar vast te maken werd in iedere hoek een van elk plaatje een gat geboord met een diameter van 6mm. Door alle plaatjes boven elkaar te zetten en 4x 15cm lange schroefdraad van 6mm straal door de plaatjes te schuiven, werd de schets gerealiseerd. Om alle plaatjes op hun plaats te houden werden nog bij elk plaatje 8 moertjes geplaatst. Om de motoren eraan vast te maken, werd er een L-constructie gemaakt in aluminium.

De nadelen die hier werden ondervonden waren dat alle moertjes steeds loskwamen en het aluminium zeer beweeglijk was.

* **2de constructie:** voor de tweede constructie werden Meccano staafjes gebruikt. Deze staafjes zijn gemaakt van staal dus zeer stevig. Onder de Arduino UNO liggen geen Meccano plaatjes, maar een plaatje van kunststof zodat er geen kortsluiting mogelijk is.

Ik heb **constructie 2** gekozen omdat het veel steviger is. Wanneer de robot in onevenwicht is, komen de moertjes niet los. Het ziet er ook professioneler uit.

# Werking

## GY-521 en de MPU6050

De MPU is een MEMS en bevat een 6DOF (6 degrees of freedom):

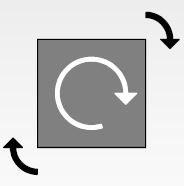
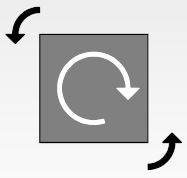
-3 assen (x, y, z) van de gyroscoop

-3 assen (x, y, z) van de accelerometer

### De gyroscoop gebruiken

Wanneer de gyroscoop stilstaat, is het referentiepunt 0.

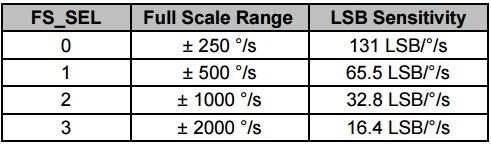
De gyroscoop staat stil, en geeft een waarde van **0**.



|  |  |
| --- | --- |
| De gyroscoop draait in **tegenwijzerzin**. En geeft een **negatieve** waarde. | De gyroscoop draait in **wijzerzin**. Geeft een **positieve** waarde. |

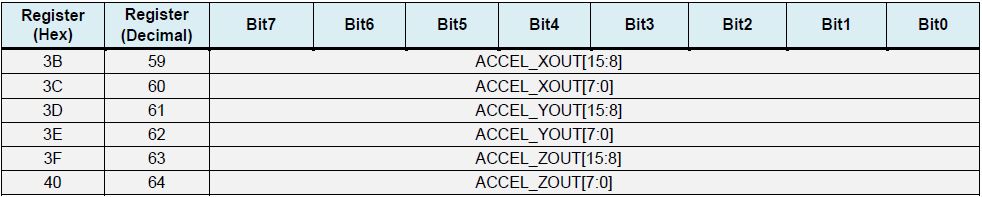
Er zijn 4 verschillende mogelijkheden. Elke mogelijkheid heeft een verschillende maximale grootte en gevoeligheid.

De gevoeligheid, of LSB Sensitivity, heeft als eenheid LSB/°/s of LSB/g (g=9,81m/s2). Dit betekent dat per °/s of g een bepaald aantal laagste bitwaarden verandert, LSB (Least Significant Bits). De Full Scale Range is de maximale waarde, het register wordt opgevuld met 1’tjes.Het FS\_SELregister geeft de 4 verschillende gebruiksmogelijkheden weer.

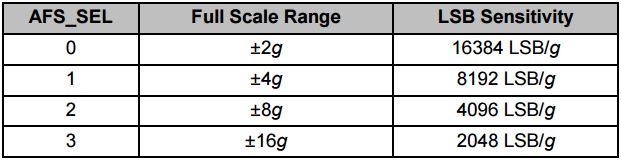


### De accelerometer gebruiken.

De accelerometer meet de versnelling op exact dezelfde manier als de gyroscoop. De metingen die Accelerometer maakt komen in de onderstaande register.



De gegevens die op de registers staan zijn binair. Het omzetten naar decimaal wordt gedaan aan de aan hand van de AFS\_SEL (Accelerometer Full scale select). De Full Scale Range geeft de limiet weer diede accelerometer kan meten. De LSB (Least Significant Bit) sensitivity geeft de nauwkeurigheid weer.



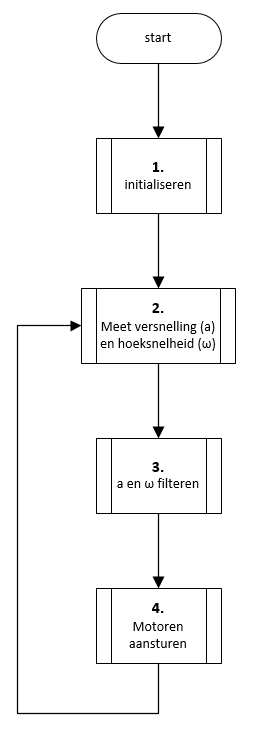
Het uitrekenen van de g waarden gebeurt als volgt

|  |  |
| --- | --- |
| Bij AFS\_SEL = 0 | Bij AFS\_SEL = 3 |
| Limiet is ±2g | Limiet is ±16g |
| Bij ACCEL\_XOUT:  0001001110101110 (5038) | Bij ACCEL\_XOUT:  0001001110101110 (5038) |
| Uitgerekende waarde:  5038 LSB/ 16384 LSB/g:  0,30749g | Uitgerekende waarde:  5038 LSB/ 2048 LSB/g:  2,4599609375g |

## Het programma en de flowchart

### Flowchart: Hoofdprogramma

**De flowchart:**



(1.) initaliseren:

Deze functie initialiseert alle componenten en stelt alles goed in om te kunnen starten. (Deze functie is onderverdeeld in meerdere functies, zie hiervoor verder.)

(2.) Meet versnelling (a) en hoeksnelheid (ω):  
Deze functie gebruikt de IMU om de waarden van de gyroscoop en de accelerometer op te meten. De accelerometer werkt met een eenheid m/s², dit is de valsnelheid. De gyroscoop werkt met een de eenheid °/s en dit is de hoek uitgedrukt per seconden.  
De functie gebruikt een I²C-communicatie om deze registers op te vragen.

(3.) a en ω filteren:

Om de ruwe waarden te filteren naar bruikbare waarden, wordt gebruik gemaakt van de complementaire filter (zie hiervoor verder)

(4.) Motoren aansturen:

Deze functie stuurt de motoren aan in functie van de versnelling en de hoeksnelheid die eerder werden berekend (deze functie is onderverdeeld in meerdere functies, zie hiervoor verder)

#### [https://developer.valvesoftware.com/w/images/7/7e/Roll_pitch_yaw.gif](https://www.google.be/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjfx-CqxqzMAhWlBcAKHUPxCX8QjRwIBw&url=http://theboredengineers.com/2012/05/the-quadcopter-basics/&v6u=https://s-v6exp1-v4.metric.gstatic.com/gen_204?ip%3D81.164.125.111%26ts%3D1461682149344586%26auth%3Db7cutnzhfkzrfxfupk35yttvswhd7lz2%26rndm%3D0.48397385320562686&v6s=2&v6t=106028&psig=AFQjCNEMXfwuw3f0Z1rEotiQ9eoQJDCXhw&ust=1461768549298289) a en ω filteren:

De accelerometer kan een accurate pitch opmeten als de robot in rust staat of slechts heel weinig beweegt. Als de robot beweegt, kn de accelerometer de snelheid niet meer accuraat volgen. De gyroscoop geeft de hoeksnelheid. Na het integreren van de data van de gyroscoop met een bepaalde tijdsduur kan de pitch berekend worden . De pitch zal na een tijd niet meer juist zijn door het driften van de gyroscoop.

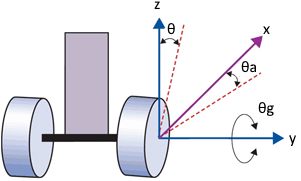
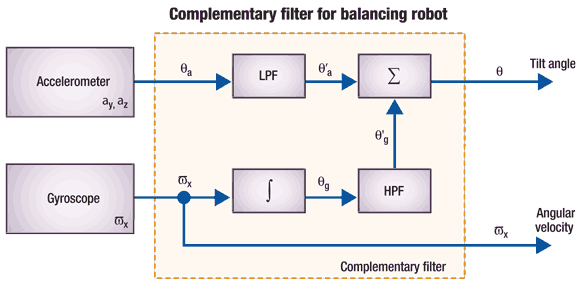
De complementaire filter wordt gebruikt om de data van de accelerometer en de data van de gyroscoop samen te voegen en een pitch te berekenen. Door het juiste gebruik van beide componenten worden bijna alle storingen weggewerkt.

In deze filter wordt gebruik gemaakt van een laag-doorlaat filter (Low-Pass filter of LPF) voor de accelerometer en een hoog-doorlaat filter (High-Pass filter of HPF) voor de gyroscoop.

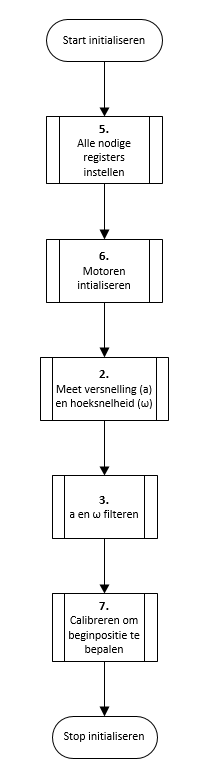
Voor θa, de static tilt hoek, te meten heb 2 of 3 assen nodig van de accelerometer, de ay en de az. Om de θg te berekenen heb je 1 of 2 assen nodig van de gyroscoop, de ωX.

Na het berekenen van θg en θa gaat de filter deze waarden gebruiken om een pitch (θ) te bekomen.

Het volledig proces van de filter is te zien in het volgend diagram.

Om θa te bekomen: . De waarden en zijn waarden tussen -1g en 1g.  
Om θg te bekomen: . Waarbij de x waarden van de gyroscoop, de offset waarde , S de nauwkeurigheid en dt de tijdscontante zijn.  
Om de pitch te berekenen zetten we de HPF op en de LPF op de , als de constante β kiezen als 0.95 dan krijgen we: **->**

### Uitleg functie initialiseren

(5.) Alle nodige registers instellen:

Hier worden alle registers van de IMU juist ingesteld. Door dit te doen gaat de IMU:

1. uit slaapmodus
2. de gyroscoop instellen
3. de accelerometer goed instellen
4. controleren of er communicatie is tussen de Arduino UNO en de GY-521.

Als tijdens de controle blijkt dat er geen communicatie is met de Arduino UNO, dan komt er op het seriële scherm een error te staan.

(6.) Motoren initialiseren:

De motoren moeten worden geïnitialiseerd worden om ze te   
kunnen gebruiken

(2.) Meet versnelling (a) en hoeksnelheid (ω):

Hier wordt een functie die in het hoofdprogramma zit,   
opgeroepen. (zie hoofdprogramma)

(3.) a en ω filteren:

Hier wordt een functie die in het hoofdprogramma zit,   
opgeroepen. (zie hoofdprogramma)

(7.) Calibreren om beginpositie te bepalen:

De eerste waarden wordt opgeslagen en blijven constant in het hele programma. Deze waardenzorgen ervoor dat de andere   
positiewaarden worden vergeleken met de beginpositie worden gecalibreerd.

### Uitleg functiemotor aansturen:

(8.) PID regelaar:

Dit is een regelaar die de versnelling en hoeksnelheid ontvangt van de filter. Hij berekent de PID waarde die de motoren moeten uitvoeren. De PID is liefst zo efficiënt mogelijk. (Zie verder voor uitleg en instellingen PID)

(9.) Servomotoren aansturen:

De berekende waarden van de PID regelaar worden naar de servomotoren gestuurd.

### Regelaars

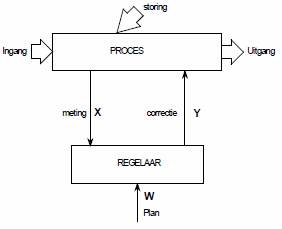
Wat is een regelaar?  
Een regelaar stuurt bij naar een bepaalde georiënteerde positie. In dit geval is dit 90°.

Het doel van een regelaar?  
Het onder controle houden van een proceseenheid.

Waarom regelen?

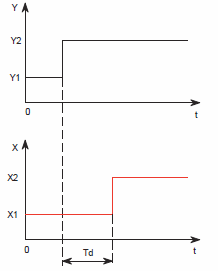
Als we niet zouden regelen zou het proces niet werken, het zou niet recht kunnen blijven staan of bijsturen.

We kunnen regelen doormiddel van vele regelaars. De regelaar om te balanceren is een PID-regelaar (later meer uitleg). De regelaar moet ook juist ingesteld worden anders zou het balanceren ook mislopen. (later meer uitleg)

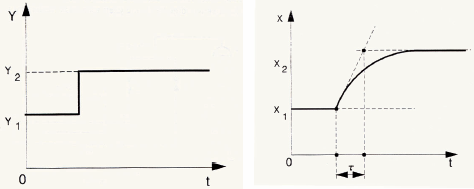
Hoe regelen?

* Bepaal de basisvoorwaarden voor het regelen van het proces:
  + X (de meting): Alle X, Y en Z-waarden van de gyroscoop en Accelerometer
  + Y (de correctie): de snelheid en de stappen berekent door de regelaar.
  + W (het plan): loodrecht blijven staan t.o.v. de grond m.a.w. 90° met de grond blijven staan.
* Bepaal de ingang, uitgang en de eventuele storing.
  + de ingang: de richting en snelheid van het onevenwicht
  + de uitgang: de richting en snelheid van de motoren.
  + de storing: uit balans geduwd worden.
* Bepaal het proces:
  + met de motoren laten balanceren op 2 wielen.
* Bepaal de regelaar:

voor we een proces kunnen gaan regelen, is het noodzakelijk om met een aantal parameters en eventuele storing rekening te houden:

* + - de dode tijd van het proces:

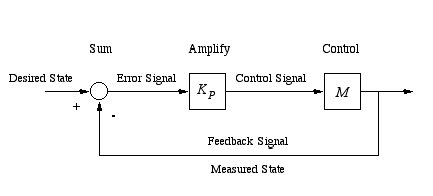
Dit is een vertraging van het proces, deze is in miniem als we het hebben over het overbrengen van data/correctie. Er komen heel veel dode tijden aan te pas wanneer de motoren aan het draaien zijn, want na elke stap wachten de motoren met een tijdsparameter die de regelaar berekend heeft, dit geeft ook de snelheid van de motoren aan

* + - De tijdsconstante en traagheid van het proces: Y1 is het de huidige positie en Y2 is de gewenste waarde. Om terug naar de gewenste waarde te balanceren zal dit tijd nemen. De traagheid kan worden versneld door een goed ingestelde regelaar.
    - Versterkingsfactoren van het proces: de interne versterking van een proces bepaalt hoe sterk de gemeten variabele X (de meting) reageert op een wijziging van het corrigerende stelsignaal Y. de versterkingsfactor is de verhouding van de gemeten waarde ΔX op een wijziging van ΔY, dus:

Bij een PID regelaar zijn er 3 versterkingswaarden die moeten ingesteld worden (, , ).

#### De proportionele regelaar of P-regelaar

Een proportionele regelaar is een lineair feedback regelaar.



Cruise control is een voorbeeld van een proportionele regelaar. De P-regelaar geeft een kracht wanneer de auto niet op zijn ingestelde snelheid rijdt en geeft geen kracht wanneer hij te snel of de juiste snelheid rijdt.

Formule:

: uitgang van de proportionele regelaar  
 : Proportionele instelwaarden

Fout in de functie van de tijd:  
 : ingestelde waarden - gemeten waarden  
  
De voordelen van een P-regelaar zijn:

* sneller maken van het systeem en verkleinen van de standfout of dus vergroten van de statische nauwkeurigheid naar mate Kr groter wordt.
* verschuiven van de polen van het gesloten systeem t.o.v. het open systeem en daardoor een totaal ander systeem creëren.
* goede ruisonderdrukking bij grote Kr waarden.

De nadelen zijn:

* mogelijk instabiel systeem bij te grote Kr waarden.
* te hevige systemen bij te grote Kr waarden.
* geen ruisonderdrukking bij te kleine Kr waarden.
* ontstaan van een standfout, die groter is naarmate Kr kleiner is

#### De integrerende regelaar of I-regelaar.

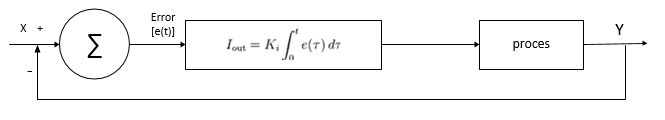
I_{\mathrm{out}}=K_{i}\int_{0}^{t}{e(\tau)}\,{d\tau}Wanneer er een fout optreedt in het proces is het nodig dat de gewenste waarde dezelfde blijft dan de gemeten waarde. Om dit te verwezenlijken moet aan een P-regelaar iets worden toegevoegd. Dit iets noemen we een integrerende werking of I-functie of ook nog herstelwerking genoemd.

Formule:

: uitgang van de integrerende regelaar  
 : integrerende instelwaarden  
 : integraal van de error in functie van de tijd vanaf het tijdstip 0.

Fout in de functie van de tijd:  
 : ingestelde waarden - gemeten waarden

De integraal werkt fouten weg die in het verleden zijn gemaakt en werkt ze geleidelijk aan bij. De regelkring is weergegeven in de volgende figuur.



Het foutsignaal e wordt geïntegreerd en vermenigvuldigd met 1/τi . De staprespons van de I-regelaar is een lineair toenemende functie hetgeen ook gebruikt wordt als blokdiagram voor de I-regelaar(zie bovenstaande figuur). We kunnen de uitgang van de I-regelaar ook op de volgende wijze schrijven in functie van de ingang:

De voordelen van een I-regelaar zijn:

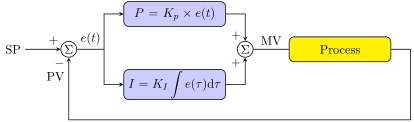
* het elimineren van de standfout of om het even welke statische fout die optreedt na de integrator, omdat de regelaar blijft integreren en dus het stuursignaal blijft verhogen totdat de fout nul wordt of omdat de statische versterkingsfactor van de regelaar oneindig is.

De nadelen zijn:

* mogelijk instabiel systeem bij een foutieve (te kleine) τi waarde (te snelle integrator).
* te langzame systemen bij te grote τi waarden (te langzame integrator).

#### De proportioneel-integrerende regelaar of PI-regelaar.

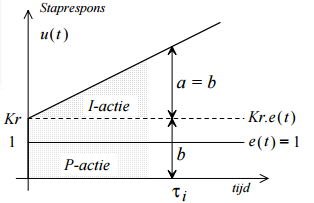
Formule:



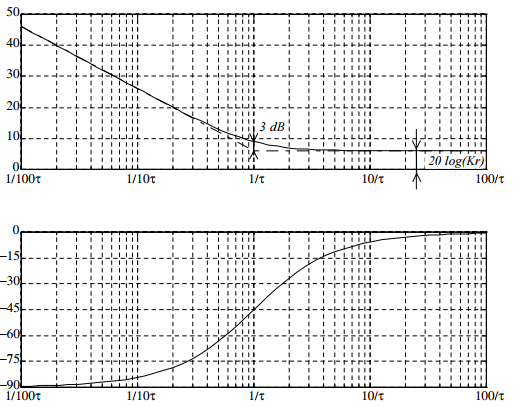
Zoals we uit de formule kunnen uitmaken, is de PI-regelaar niets anders dan de combinatie van een P-regelaar en een I-regelaar. De voor- en nadelen van de P- en de I-regelaar worden hier dan ook gecombineerd. Wel staat nu in de teller van de I-regelaar eveneens de versterkingsfactor Kr. We kunnen de PI-regelaar dus opsplitsen in een P- en een I-actie zoals de vorige figuur vertoonde.

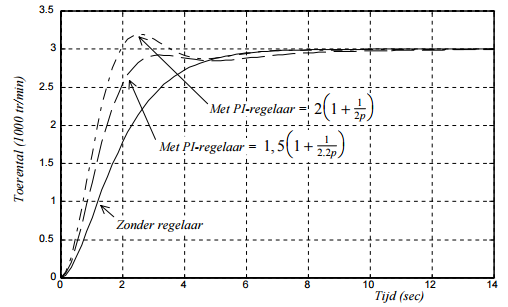
De uitgang van de PI-regelaar is dus gelijk aan de som van de integraal van het ingangssignaal en het ingangssignaal zelf, beide met een bepaalde factor versterkt (of verzwakt).

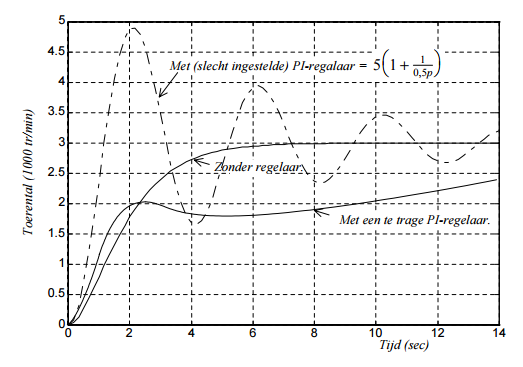
uit deze formule volgt de stapsrespons van de PI-regelaar, weergegeven in de volgende figuur.



De reactie van de PI-regelaar bestaat uit een P-actie en een I-actie. Beide acties zijn gelijk op het ogenblik . (daar waar (a) = (b)). Indien klein is, is de integrerende actie heel snel. Indien groot is, is de integrerende actie heel traag. De P-actie is daarentegen onafhankelijk van de tijd. De volgende figuur geeft het Bode-diagram van de PI-regelaar. De statische versterking is oneindig, zoals bij de I-regelaar, zodat de standfout voor het gesloten systeem meestal nul is. Voor grote frequenties komt de versterking met de P-regelaar. De fase is bijna -90° voor lage frequenties en 0° bij hoge frequenties. Omwille van de (gedeeltelijk) negatieve fase van de PI-regelaar kan de stabiliteit van het gesloten systeem in het gedrang komen bij een slechte keuze .



Voorbeeld met en zonder een PI-regelaar: Voorbeeld met een slecht en goed ingestelde PI-regelaar:



# Kostprijs

|  |  |
| --- | --- |
| onderdelen | prijs |
| Arduino UNO R3 | € 35,00 |
| 2 servomotoren | € 30,00 |
| Gy-521 | € 12,00 |
| kabeltjes | € 0,50 |
| Batterij-houder | € 1,50 |
| 4 AA batterijen 1,5V | € 4,50 |
| Totaal kostrprijs | € 83,50 |

# besluit

Bij het starten van deze proef waren mijn verwachtingen veel hoger dan het uiteindelijke haalbare. Dit kwam vooral door de hoeveelheid problemen die ik tegenkwam tijdens het realiseren van mijn GIP. Deze problemen waren vooral in verband met de motoren en de constructie van de robot. De motoren heb ik 2 keer moeten veranderen omdat ik de verkeerde motoren gekozen had. Dee constructie heb ik ook 2 keer moeten veranderen. De eerste constructie was enkel in hout gemaakt en was niet stevig. De tweede constructie is met Meccano gemaakt. Deze was het stevigste. Door de problemen heb ikmoeten nadenken over verschillende mogelijke opslossingen. Het oplossen van de problemen was een hele leerschool voor mij. Uiteindelijk was mijn doel “het laten balanceren” van de robot. Dit doel is tot nu toe niet helemaal behaald omdat de robot bij te grote hoekafwijkingen niet in balans blijft. Bij kleine hoekafwijkingen balanceert de robot wel.

Ik wil mijn mentor, mijn ouders, mijn vriendin, en mijn medeleerlingen en leerkrachten bedanken voor de steun en de hulp bij de problemen die ik tegenkwam in de loop van mijn project.

# bijlagen

## bronnenlijst

1. <https://www.google.be/?gfe_rd=cr&ei=YF7DVoX-KeLU8geNnbv4AQ&gws_rd=ssl>
2. <http://www.sensor.nl/microsonic+ultrasoon+sensoren/het+ultrasone+werkingsprincipe/cDU2645-Het+ultrasone+werkingsprincipe.aspx>
3. [http://www.limetrace.co.uk/Interfacing-HC-SR04-ultrasonic-to-Arduino UNO](http://www.limetrace.co.uk/Interfacing-HC-SR04-ultrasonic-to-Arduino)
4. <http://www.intorobotics.com/interfacing-programming-ultrasonic-sensors-tutorials-resources/>
5. [http://winkleink.blogspot.ro/2012/05/Arduino UNO-hc-sr04-ultrasonic-distance.html](http://winkleink.blogspot.ro/2012/05/arduino-hc-sr04-ultrasonic-distance.html)
6. [http://www.tautvidas.com/blog/2012/08/distance-sensing-with-ultrasonic-sensor-and-Arduino UNO/](http://www.tautvidas.com/blog/2012/08/distance-sensing-with-ultrasonic-sensor-and-arduino/)
7. <https://nl.wikipedia.org/wiki/Corioliseffect>
8. <http://www.osmtec.com/unipolar_vs_bipolar.htm>
9. <http://websites.khlim.be/jbaeten/cursussen/REG1H4.pdf>
10. [http://www.limetrace.co.uk/Interfacing-HC-SR04-ultrasonic-to-Arduino UNO](http://www.limetrace.co.uk/Interfacing-HC-SR04-ultrasonic-to-Arduino)
11. [https://www.Arduino UNO.cc/en/Main/Arduino UNOBoardUno](https://www.arduino.cc/en/Main/arduinoBoardUno)
12. <http://www.starlino.com/imu_guide.html>
13. <https://nl.wikipedia.org/wiki/Schijnkracht>
14. <https://en.wikipedia.org/wiki/PID_controller>
15. <https://en.wikipedia.org/wiki/Proportional_control>
16. <https://nl.wikipedia.org/wiki/Proportionele_regelaar>
17. <https://en.wikipedia.org/wiki/PID_controller>
18. <https://nl.wikipedia.org/wiki/Integraalrekening>
19. <https://nl.wikipedia.org/wiki/Transferfunctie>
20. <http://cs.brown.edu/~tld/courses/cs148/02/sensors.html>
21. <http://sebastiannilsson.com/en/tag/selfbalancing-robot-2/>
22. <http://jjrobots.com/pid/>
23. <http://www.pieter-jan.com/>
24. <https://nl.wikipedia.org/wiki/Hoeksnelheid>
25. <http://kampis-elektroecke.de/?page_id=3066>
26. <http://educatie-en-school.infonu.nl/werkstuk/84921-een-conclusie-schrijven.html>
27. <http://www.x-io.co.uk/res/doc/madgwick_internal_report.pdf>
28. <http://www.avrfreaks.net/forum/tut-hard-gyros-and-accelerometers-basics?name=PNphpBB2&file=viewtopic&t=89294>
29. <http://www.flightsimworld.com/forums/topic/108038-what-is-gyro-drift/>
30. <http://www.mouser.co.uk/applications/sensor_solutions_mems/>
31. <https://nl.wikipedia.org/wiki/I%C2%B2C-bus>
32. Datasheets van de IMU: <http://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>

## Belangrijke datasheets

**Arduino UNO Rev3**

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](http://www.atmel.com/Images/doc8161.pdf) |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328P) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |
| Clock Speed | 16 MHz |
| Length | 68.6 mm |
| Width | 53.4 mm |
| Weight | 25 g |

**Elektrisch schema van de IMU: GY-521**

