

GEAVANCEERDE EV3 PROGRAMMEER-LESSEN

Proportionele Besturing



Door Droids Robotics
Code bijgedragen door The Construction Mavericks



Lesdoelen

1. Leren wat proportionele besturing betekent en waarom je het gebruikt
2. Leren om proportionele besturing toe te voegen aan de Gyro-, Kleur-, en Ultrasoon Sensoren

Voorkennis: Lessen gedaan over Rekenblokken, Kleursensor-kalibratie en Data-Verbindingen

Leren en discussiëren over proportionele besturing

- Als we het in ons team hebben over “proportioneel”, zien we het als een spel
- Een teamlid wordt geblinddoekt. Hij of zij moet zo snel mogelijk een kamer door en moet precies stoppen bij een lijn op de grond (gebruik afplakband om een lijn op de vloer te plakken)
- De rest van het team moet bevelen geven
- Als het teamlid ver weg is moet hij snel lopen en grote stappen nemen. Maar dan komt hij dichterbij de lijn, als hij door blijft rennen zal hij over de lijn heen gaan. Je moet dat teamlid dus vertellen dat hij langzamer moet gaan lopen en kleinere stappen moet zetten
- Je moet de robot op dezelfde manier programmeren!



Proportionele besturing?

- Wat betekent proportioneel?
 - De robot beweegt proportioneel – hij past zijn beweging aan (bv. een aanpassing aan zijn snelheid) op basis van hoever de robot is van zijn gestelde doel
 - Bij een proportionele lijnvolger zal de robot aanpassen hoe scherp zijn draai is op basis van hoe ver hij van de lijn verwijderd is
- Proportionele besturing kan preciezer en sneller zijn
- De Pseudocode voor elke proportionele besturings programma bestaat uit twee stappen:
 1. **De fout berekenen** → Hoe ver is de robot van het doel
 2. **Een verbetering maken** → de robot onderneemt een actie die proportioneel is met de fout (daarom heet dit proportionele besturing). Je moet de fout vermenigvuldigen met de vergrotingsfactor om de verbetering te bepalen

Opdrachten

- Om je te leren hoe je proportionele besturing gebruikt, hebben we 3 opdrachten:
 - Volgende Hond: Gebruik proportionele besturing met de ultrasoon-sensor om de robot constant 7cm van een persoon te laten (zelfs als dat persoon beweegt)
 - Lijnvolger: Gebruik proportionele besturing met de licht-sensor vloeiend een lijn te laten volgen. (Verdere details in the Proportionele Lijnvolger les)
 - Gyro Bocht: Gebruik proportionele besturing en de gyro-sensor om de robot met precisie naar een doelhoek te draaien

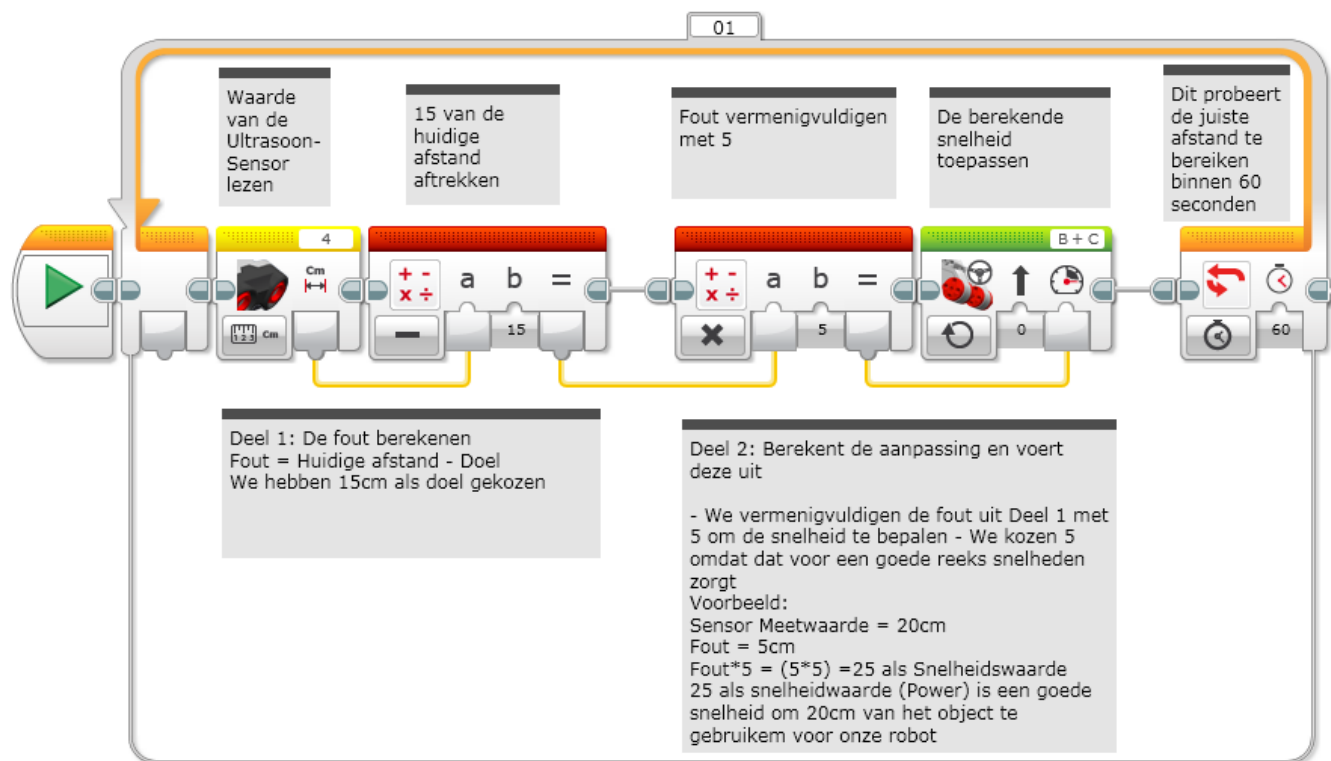
Pseudocode/Hints

| Toepassing | Doel | Fout | Samenhang |
|----------------------|---|---|---|
| Volgende Hond | Ga naar de afstand (die als doel gesteld is) van aan object | De afstand tot het doel (huidig_afstand – doel_afstand) | Snelheid v.d. beweging op basis van de afstand |
| Lijnvolger | Blijf op de rand van de lijn | Wat is het verschil in lichtvaarde tussen de huidige lokatie en de rand van de lijn (huidig_licht – doel_licht) | Aanpassing van hoe scherp de bocht is op basis van de afstand vanaf de lijn |
| Gyro Bocht | Draai tot de doel-hoek | Hoeveel graden tot de doel-hoek | Snelheid van de draai op basis van overblijvende aantal graden te draaien |

Oplossing: Volgende Hond (Ultrason)

Dit is een programma wat de robot 15cm van een (bewegend) object af houdt. Dit programma gebruikt proportionele besturing

Deze code is geschreven door Droids Robotics



Oplossing: Proportionele Lijnvolger

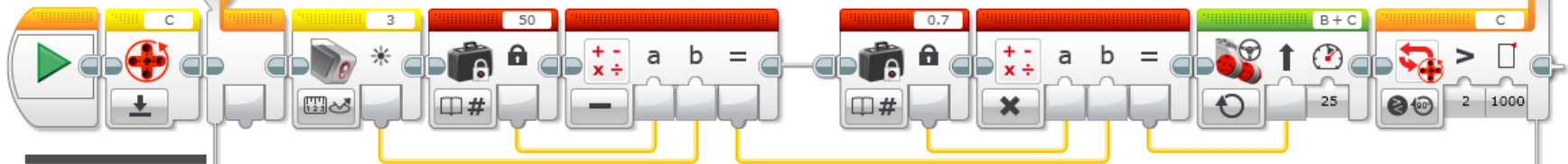
Let op!: Dit programma gebruikt Kleur-Sensoren in lichtmodus. Dit betekent dat je de sensor moet kalibreren. Lees a.u.b. onze kalibratie les voordat je hiermee verdergaat! :-)

We raden aan dat jullie team een proportionele lijnvolger gebruikt, zoals deze. Dit is namelijk een redelijk vloeiende lijnvolger. Er zijn ook betere lijnvolgers (die PID control gebruiken), maar een lijnvolger die de "P" gebruikt is een goed begin. Een proportionele lijnvolger past de hoek van zijn bochten aan op basis van hoe ver de robot van de lijn is.

Deze code is geschreven door Droids Robotics

01

Elk proportioneel besturingsprogramma (zoals deze lijnvolger) bestaat uit 2 delen: Deel 1 berekent de fout (in dit geval is dat hoe ver de robot van de lijn is) en Deel 2 berekent een aanpassing die proportioneel is aan de fout (in dit geval hoe ver je draait). Je kan proportionele besturing ook met andere sensoren gebruiken. Dat werkt ook goed!



Let op!: Je hoeft geen Constante-Blok met een data verbinding te gebruiken. We hebben dat gedaan om het duidelijker was dat je een zelf deze constanten kan veranderen zodat het programma meer aansluit bij je behoeften

Deel 1: Fout berekenen - Ons doel is op de rand van de lijn (lichtsensor-waarde = 50). Het rekenblok hierboven berekent hoe ver de robot is van het doel van 50 - Het Constant-Blok hierboven is ons doel. Je kan dit veranderen voor verschillende soorten lijnen - Let op dat in het ergste geval, je lichtsensor een waarde van 0 of 100 geeft (Ver van de lijn af). Dit geeft dan een fout van -50 en 50

Deel 2: Berekent de aanpassing en voert de uit - We vermenigvuldigen de fout uit Deel 1 met 0.7 om de draaiwaarde te berekenen - We kozen 0.7 zodat we als we het ergste geval van de fout (50 or -50) hebben, het Stuurwaarde in het Bewegings-Blok hierboven 35 or -35 zal zijn, wat een scherpe draai is.
- Je kan deze waarde aanpassen naar je benodigheden

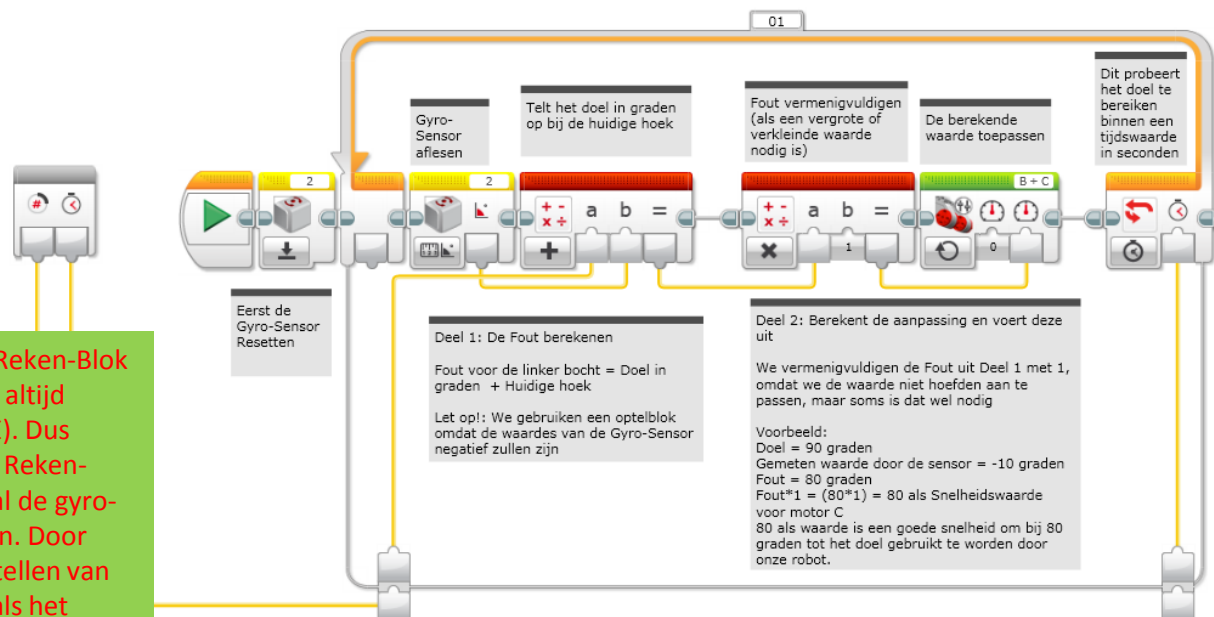
Deze lijnvolger eindigt na 1000 graden. Pas aan naar je benodigheden

Oplossing: Linker Gyro-Bocht

Het doel van dit programma is om een proportionele draai linksom te maken binnen een bepaalde tijd. Bedankt Construction Mavericks voor de originele code die we aangepast hebben! :-)

Dit is de herhaling die de proportionele draai bevat. (Het belangrijkste deel)
 BELANGRIJK - *Een Linker draai veroorzaakt de Gyro-Sensor negatieve waarden te geven*

- 1) Gyro-Sensor waarde aflezen
- 2) Gyro-waarde optellen bij het doel. Vergroting/Verkleining mogelijk
- 3) Resultaat naar de snelheidswaarde van linker motor, rechter motor blijft stilstaan
- 4) Herhaal tot een bepaalde tijd



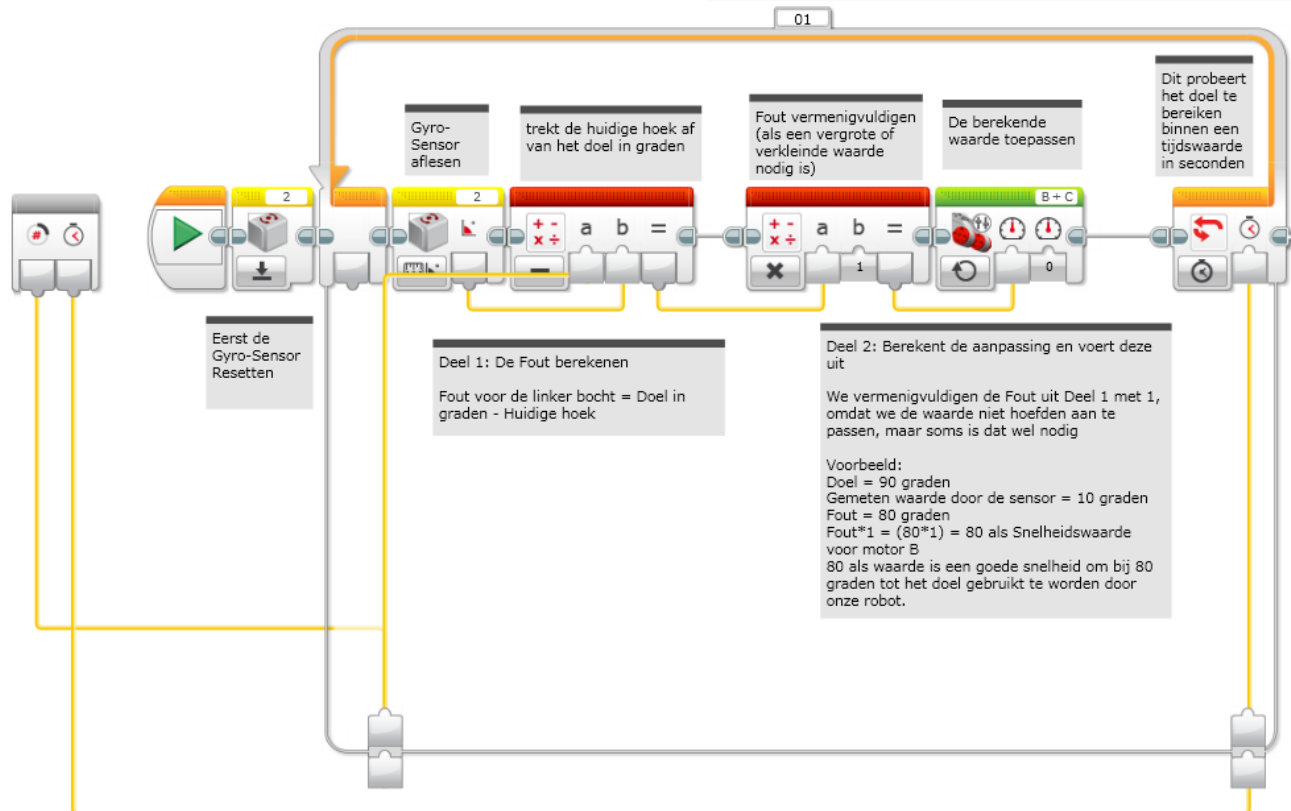
Wat is er aan de hand met het Reken-Blok in de linker bocht? Je berekent altijd (DOEL) MIN (HUIDIGE-WAARDE). Dus waarom een SOM/OPTTEL/PLUS Reken-Blok? Als je naar links draait, zal de gyro-sensor negatieve waarden geven. Door wiskunde weten we dat het optellen van een negatief getal hetzelfde is als het aftrekken van hetzelfde getal, maar dan positief. Dat is dus waarom we een Optellen Reken-Blok gebruiken om de fout te berekenen.

Oplossing: Rechter Gyro-Bocht

Het doel van dit programma is om een proportionele draai rechtsom te maken binnen een bepaalde tijd. Bedankt Construction Mavericks voor de originele code die we aangepast hebben! :-)

Dit is de herhaling die de proportionele draai bevat. (Het belangrijkste deel)

- 1) Gyro-Sensor waarde aflezen
- 2) Gyro-waarde aftrekken van het doel. Vergroting/Verkleining mogelijk
- 3) Resultaat naar de snelheidswaarde van linker motor, rechter motor blijft stilstaan
- 4) Herhaal tot een bepaalde tijd



Discussiepunten

1. Wat betekent “proportionele besturing” ?

Antw: Meer of minder bewegen op basis van hoever de robot vanaf een bepaald punt verwijderd is

2. Wat is een overeenkomst bij alle proportionele-besturings-code?

Antw: Het berekenen van de fout and en de verbetering daarvan

Credits

- Deze les is gemaakt door Sanjay Seshan en Arvind Seshan van Droids Robotics (team@droidsrobotics.org).
- Originele “Gyro Turn Code” (Gyro-Bocht-Code) geleverd door de Construction Mavericks (frank.levine@gmail.com)
- Meer Lessen bij www.ev3lessons.com
- Vertaald door Maerlant Robotica (maerlant-robotica.nl/)



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).