KU Leuven
Departement Computerwetenschappen

P&O: Computerwetenschappen

Tussentijdsverslag 1

Team:
Brons

Mattias Buelens
Vital D'haveloose
Dennis Frett
Stijn Hoskens
Matthias Moulin

Academiejaar 2012 – 2013

Samenvatting

Naar aanleiding van het project van Probleemoplossen en Ontwerpen: Computerwetenschappen 2012-2013 omtrent autonome intelligente robots, ontwerpt Team Brons een Lego Mindstorms NXT Robot. De NXT robot, genaamd MazeStormer, van Team Brons wordt aangestuurd door twee onafhankelijke servomotoren en heeft een vrij roteerbaar klein achterwiel om kort en precies te kunnen roteren. Na kalibratie zijn translatie en rotatie correcties doorgevoerd. De Java software rond de robot voorziet een in lagen gestructureerd software design en biedt naast een manuele aansturing van de fysieke NXT ook uitgebreide simulatiemogelijkheden voor een virtuele NXT, dewelke het gedrag en de functionaliteit van de fysieke NXT zo nauwkeurig mogelijk tracht te benaderen. Dit alles via een gebruiksvriendelijke user interface.

Inhoudsopgave

Inleiding

Het project van Probleemoplossen en Ontwerpen: Computerwetenschappen 2012-2013 kadert in het domein van autonome intelligente robots. Hierbij ontwerpt, bouwt, kalibreert en programmeert elk team een fysieke, Lego Mindstorms NXT, robot met behulp van de leJOS NXJ PC Application Programming Interface. Naast de aansturing van de fysieke robot, moet een virtuele robot de besturing en de functionaliteiten van de NXT zo nauwkeurig mogelijk simuleren.

De eerste deelopdracht bestaat erin te connecteren met de NXT om deze vervolgens manueel te besturen en om deze een regelmatige veelhoek te laten afleggen met willekeurige zijdelengte en willekeurig aantal hoeken. De nadruk ligt hierbij enerzijds op de communicatie tussen de gebruiker en de NXT, waarbij de gebruiker de robot aanstuurt en de robot de nodige feedback voorziet, en anderzijds op de uitvoeringsnauwkeurigheid van de NXT. Bovendien moet de simulator in staat zijn dezelfde opdrachten uit te voeren.

Het verslag geeft de voorbereiding op en de evolutie tot deze demonstratie weer van Team Brons met hun robot, genaamd MazeStormer. Hierbij wordt er aandacht besteed aan de bouw van de robot en het kalibreren van de verschillende parameters. De communicatie via Bluetooth wordt kort besproken. Het software design wordt gellustreerd aan de hand van UML klassendiagrammas, opgedeeld in interface-, controller- en modellaag of volgens de overgang tussen twee van deze lagen. Verder worden de ontwerpbeslissingen omtrent het simuleren van de NXT en de user interface in meer detail besproken. Ten slotte volgt er een opsomming van de geleverde inspanning en een korte reflectie over de bereikte resultaten van Team Brons tot nu toe.

1 Bouw robot

1.1 Fysieke bouw

Het eerste overwogen concept voor de bouw van de NXT is gebaseerd op het Steering Robot ontwerp. De aansturing van deze robot is vergelijkbaar met deze van een auto met achterwielaandrijving. De vier wielen zijn twee bij twee verdeeld over een voor- en achteras. De achteras wordt geroteerd door een eerste servomotor en de loodrechte as op de vooras wordt gedraaid door een tweede servomotor. Deze bouw maakt het snel draaien van de robot mogelijk. Dit komt echter ten koste van een grote draaicirkel. Deze draaicirkel reduceert enerzijds de bewegingsprecisie en vergt anderzijds een complexe draaitechniek om op een klein oppervlak volledig rond de as te kunnen draaien. [**?**])

Aangezien het nauwkeurigheidsaspect de bovenhand haalt op het snelheidsaspect is de uiteindelijke bouw grotendeels gebaseerd op de Express-Bot en de daarvan afgeleide Pivoting Head Explorer. Deze robot bestaat uit n onafhankelijk, aangedreven hoofdwiel aan weerszijden van de voorkant en een klein wieltje achteraan. Hierbij blijft de hoek tussen de hoofdwielen en NXT brick constant en kan het kleine achterwieltje vrij roteren. Deze opstelling maakt precies en kort draaien mogelijk. Verder is deze robot modulair opgebouwd. De aandrijving blijft hetzelfde en naargelang de functionaliteit kunnen deelcomponenten zoals sensoren worden toegevoegd. De vele uitbreidingsmogelijkheden in combinatie met eenzelfde aanstuurbasis bieden de zekerheid dat bij komende functionaliteitwijzigingen de kalibratie niet herdaan moet worden. Verder is de accu ook loskoppelbaar zonder de robot uiteen te moeten halen. [**?**]

Het verschil tussen de NXT van Team Brons en de Pivoting Head Explorer zijn de gekozen wielen. De grootste wielen geven tijdens het testen en kalibreren telkens fluctuerende afwijkingen doordat ze vervormen tijdens het roteren. Daarom wordt er geopteerd om kleinere en hardere wielen te gebruiken.

1.2 Meetresultaten

De nauwkeurigheid van de robot wordt beoordeeld op basis van drie marges: de afbuigingsmarge bij translatie, de foutmarge bij translatie en de foutmarge bij rotatie.

1.2.1 Afbuigingsmarge bij translatie

De 'MazeStormer' met de grote wielen buigt af bij het vooruit manoeuvreren. Dit wordt veroorzaakt door de vervormbaarheid van de wielen. Deze grote, vervormbare wielen worden hierdoor vervangen door kleinere minder, vervormbare wielen. Hierdoor wordt de afbuiging sterk verminderd, maar niet volledig weggewerkt. De resterende afwijking is het gevolg van een fout op de motor. Om de marge op deze fout te verkleinen, worden de wieldiameters verschillend gekozen. Door veelvuldig te testen is het optimale verschil diameterverschil bepaald.

In een finaal experiment legt de robot een afstand van 2.4 m af, waarbij op vier equidistante punten van dit pad de loodrechte afstand van de robot tot het referentie pad gemeten worden.

1.2.2 Foutmarge bij translatie

Om na te gaan of de afstand die de robot aflegt overeenkomt met de gevraagde afstand, wordt er een eenvoudige testopstelling gebruikt. Hierbij wordt de afstand gemeten tussen begin- en eindpunt en vergeleken met de afstand meegegeven in de software. De 'MazeStormer' met standaard parameters geeft een kleine afwijking. Door de wieldiameter parameters in de software echter lichtjes te corrigeren waarbij het diameterverschil bewaard blijft, legt de robot de juiste afstand af.

1.2.3 Foutmarge bij rotatie

Om een beeld te krijgen van de foutmarge bij rotatie, roteert de robot 360 graden en wordt gekeken of deze robot terug op zijn startpositie terecht komt. Door de breedte-parameter van de robot een klein beetje aan te passen, worden de resultaten nauwkeuriger.

Bij de volgende test moest de robot een vierkant met een zijdelengte van n meter beschrijven. Hierbij is de afwijking groter dan bij de vorige test. Dit komt doordat er nu zowel getransleerd als geroteerd wordt. Na elke rotatie moet de robot opnieuw accelereren. Hierdoor vergroot de fout op de hoek telkens een klein beetje. Door de breedte-parameter nauwkeuriger in te stellen, kan deze afwijking geminimaliseerd worden.

2 Algoritmes

3 Software

3.1 Bluetooth

Alle communicatie tussen de NXT robot en de gebruiker, via de user interface, verlopen via een Bluetooth connectie. Hiervoor is het LEGO MINDSTORMS NXT Communications Protocol (LCP) van Lego overgenomen. Hierbij worden er commandos naar de standaard Lego firmwire gestuurd. Een connectie via USB behoort eveneens tot de standaard communicatie mogelijkheden, maar biedt geen meerwaarde en is daarom niet opgenomen bij het ontwikkelen van de software. (Lejos, Understanding LCP) [**?**]

3.2 GUI

3.2.1 API

De grafische user interface (GUI) aan pc zijde maakt gebruik van de Swing application programming interface (API), ontwikkeld door Sun Microsystems. Deze API krijgt de voorkeur op de Abstract Windowing Toolkit (AWT), dewelke de standaard API is voor het ontwikkelen van GUIs in Java programmas. Swing maakt gebruikt van pure Java code wat de (over)draagbaarheid en dus de platform toegankelijkheid (Mac OS, Ubuntu,) vergroot. Dit komt ten koste van een verminderde snelheid en performatie in tegen stelling tot AWT. Deze maakt gebruik van platformafhankelijke native commandos, maar boet op die manier in aan platform toegankelijkheid. Verder biedt Swing ook meer mogelijkheden zoals het gebruik van iconen in tegen stelling tot AWT. De Standard Widget Toolkit (SWT), ontwikkeld door IBM, is de derde en laatste overwogen API. Door een te

beperkte basiskennis is dit pad niet verder gexploreerd en bleef het bij het uit proberen van enkele Java samples en demos. [**?**]

3.2.2 Ontwerpbeslissingen en evolutie

De eerste GUI versie is een Swing verkenning die gebruik maakt van een absolute layout, losstaande deelcomponenten en portable network graphics (png) als iconen. De gebruikservaring wordt echter beperkt door deze layout en iconen met vaste pixelgrootte. Verder vertonen de losstaande deelcomponenten meer samenhang dan oorspronkelijk gedacht. Hierdoor wordt voor de finale GUI geopteerd voor een Mig Layout, die zeer geavanceerde mogelijkheden ter beschikking stelt inzake het groeien, krimpen en schikken van de verschillende deelcomponenten. De iconen worden vervangen door vectoriconen, die niet gekenmerkt worden door kwaliteitsverlies bij vergroting. De oorspronkelijke deelcomponenten herleiden zich tot n van de volgende drie elementen:

- Een configuratiepaneel dat de gebruiker laat kiezen tussen een fysieke - of virtuele, simulerende robot. Hiernaast kan de gebruiker onafhankelijk van het gekozen robot type een mode selecteren, waarover verder meer toelichting volgt.
- Een tekstpaneel dat enerzijds opereert als proces log en error log en anderzijds de nodige feedback naar de gebruiker voorziet.
- Een mode afhankelijk paneel dat een console voorziet, waar de gebruiker de fysieke of virtuele robot kan aansturen naargelang de mode (bv.: regelmatige veelhoek afrijden). Sommige modes voorzien ook nog een paneel voor de grafische weergave van bijvoorbeeld een doolhof.

3.2.3 Modes en features

Hieronder volgen kort de verschillende modes met hun gebruiksmogelijkheden.

- De control mode laat toe de NXT manueel te bedienen alsook de snelheid van de servomotoren in te stellen.
- De polygon mode laat toe de NXT een regelmatige veelhoek af te leggen op basis van een te kiezen zijdelengte en aantal hoeken.

3.3 Simulator

- Beschrijving van de simulator.

3.4 Software design

- Geef hier een klassediagramma en een overzicht van de verschillende methodes.

4 Besluit

De MazeStormer van Team Brons is een Lego Mindstorms NXT Robot, ontwikkeld in het kader van autonome intelligente robots voor Probleemoplossen en Ontwerpen: Computerwetenschappen 2012-2013. Twee niet vervormbare, hoofdwielen aangedreven door onafhankelijke servomotoren in combinatie met een vrij roteerbaar klein achterwiel bieden een basis om kort en precies te roteren. Deze precisie is verder geoptimaliseerd door het kalibreren en invoeren van corrigerende factoren om motorafhankelijke afwijkingen tot een minimum te beperken.

Via het LEGO MINDSTORMS NXT Communications Protocol van Lego wordt er via Bluetooth gecommuniceerd tussen enerzijds de NXT robot en anderzijds de zelf ontwikkelde software. De software is geschreven in Java en maakt gebruik van de leJOS Application Programming Interface die reeds een grote waaier aan tools ter beschikking stelt om de NXT aan te sturen en te controleren. De software is wel overwogen opgedeeld in drie lagen; namelijk een interface-, controller- en modellaag. De interfacelaag houdt zich bezig met de representatie van de functionaliteit en feedback naar de gebruiker toe. De modellaag voorziet alle tools om de NXT zowel fysiek als virtueel aan te sturen. De controllerlaag zorgt voor de interactie tussen interface- en modellaag.

Naast het aansturen van een fysieke robot is het ook mogelijk een robot te simuleren. Deze virtuele robot tracht de functionaliteit en het gedrag van de fysieke robot zo nauwkeurig mogelijk te benaderen.

A Demo 1

A.1 Resultaten

...

A.2 Conclusies

...

A.3 Oplijsting aanpassingen verslag

Hier komt een summiere weergave van welke secties uit het vorige verslag gewijzigd werden.

B Beschrijving van het proces

- Welke moeilijkheden heb je ondervonden tijdens de uitwerking?
- Welke lessen heb je getrokken uit de manier waarop je het project hebt aangepakt?
- Hoe verliep het werken in team? Op welke manier werd de teamcoördinatie en planning aangepakt?

C Beschrijving van de werkverdeling

- Geef voor elk van de groepsleden aan aan welke delen ze hebben meegewerkt en welke andere taken ze op zich hebben genomen.
- Rapporteer in tabelvorm hoeveel uur elk groepslid elke week aan het project gewerkt heeft, zowel tijdens als buiten de begeleide sessies. Geef ook totalen per groepslid voor het volledige semester.

D Kritisch analyse

- Maak een analyse van de sterke en zwakke punten van het project. Welke punten zijn vatbaar voor verbetering. Wat zou je, met je huidige kennis, anders aangepakt hebben?

Referenties