

ZOEKEN, STUREN EN BEWEGEN

Basic order picking met behulp van de
KUKA YouBot

Sébastien Negrijn - 10340912
Vincent Erich - 10384081
Martijn Wardenaar - 10348190
Jelle Koster - 10434399

5 juli, 2013

Samenvatting

Order picking is een belangrijk logistiek proces dat in warenhuizen en sorteercentra voorkomt. Het heeft een grote invloed op de zogenaamde ‘supply chain’: het systeem dat betrokken is bij het verplaatsen van een product of service van de leverancier naar de klant. In dit onderzoek wordt onderzocht of de KUKA YouBot basic order picking kan uitvoeren. Basic order picking is een order picking methode waarbij de order picker telkens één product ophaalt totdat de gehele order is afgehandeld. In dit onderzoek fungeert de YouBot als order picker. Via een interface kan worden opgegeven welke items de YouBot moet ophalen. Met behulp van een webcam die op de arm van de YouBot is bevestigd, zal de YouBot de items herkennen, oppakken, meenemen en op één punt verzamelen.

1 Inleiding

Dit onderzoek heeft als doel om een robot bepaalde objecten op te laten pakken die op verschillende plekken liggen en deze objecten vervolgens naar een andere vaste plek te verplaatsen. Dit wordt ook wel ‘order picking’ genoemd. ‘Order picking’ is een belangrijk logistiek proces dat voorkomt in bijvoorbeeld warenhuizen en sorteercentra. ‘Order picking’ heeft een grote invloed op de zogenaamde ‘supply chain’: het systeem dat betrokken is bij het verplaatsen van een product of service van de leverancier naar de klant.

Robots worden tegenwoordig voor verschillende doeleinden gebruikt, bijvoorbeeld voor de assemblage van goederen in fabrieken, voor het uitvoeren van huishoudelijke taken en voor het uitvoeren van militaire operaties. Robots voeren taken over het algemeen sneller en efficiënter uit met een hogere precisie. Het gebruik van robots voor ‘order picking’ zal daarom als voordeel hebben dat orders sneller en efficiënter worden afgehandeld.

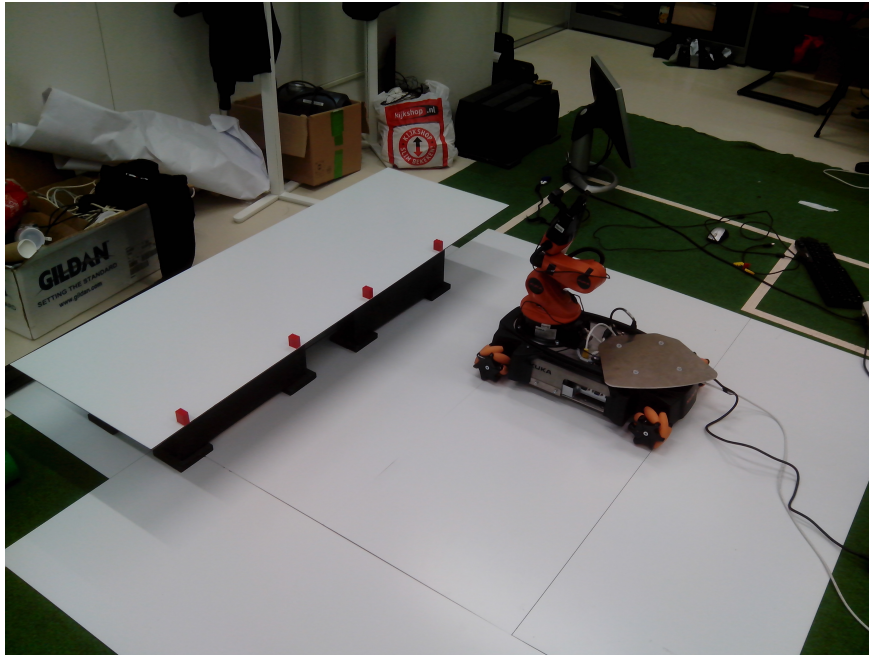
Er zijn verschillende ‘order picking’ methoden. Hieronder volgt een opsomming van de belangrijkste methoden [Piasecki,].

- **Basic order picking:** Bij deze methode haalt de ‘order picker’ telkens één product op totdat alle producten zijn opgehaald/totdat de gehele order is afgehandeld. De producten liggen op een vaste plaats en de ‘order picker’ volgt een bepaalde route die langs alle producten loopt.
- **Batch picking / Multi-order picking:** Bij deze methode zijn meerdere orders gegroepeerd in kleine partijen. De ‘order picker’ verwerkt in één keer alle orders van een bepaalde partij. De ‘order picker’ dient alle partijen af te handelen.
- **Zone picking:** Bij deze methode is de plek waar de producten zich bevinden, onderverdeeld in een aantal zones. Aan elke zone wordt een aantal ‘order pickers’ toegekend. Bij het afhandelen van een order halen de ‘order pickers’ alleen de producten op die binnen de eigen zone vallen. De order wordt telkens naar een andere zone verplaatst (nadat het ophalen van de producten in de vorige zone is afgerond) zodat de order stapsgewijs kan worden afgehandeld.
- **Wave picking:** Deze methode is een variatie op ‘batch picking’ en ‘zone picking’. In plaats van dat de order telkens naar een andere zone wordt verplaatst (zoal bij ‘zone picking’ het geval is), worden de producten in ieder zone tegelijkertijd opgehaald. De producten worden later bij elkaar gevoegd zodat de order kan worden afgehandeld.

Het gebruik van robots (robotarmen) voor ‘order picking’ wordt in de praktijk al veel toegepast. Zo worden er robotarmen gebruikt bij de assemblage van automobielen. De robotarmen pikken daarbij de juiste onderdelen uit voor een bepaald type auto en monteren de onderdelen

op de auto. De robotarmen worden ook gebruikt om binnengekomen orders van klanten bij een webshop bij elkaar te pakken in een magazijn. Door de toegenomen precisie van de robotarmen levert dit veel tijdswinst op en bespaart het kosten [Thatcher and Easterling, 2003].

In dit onderzoek ligt de focus op het uitvoeren van ‘basic order picking’ door een robot op kleine schaal. Er wordt daarbij gebruik gemaakt van de KUKA YouBot (zie figuur 1). De vraag die centraal staat in dit onderzoek is de volgende: Is het mogelijk om de KUKA YouBot op kleine schaal ‘basic order picking’ uit te laten voeren? Er wordt verwacht dat dit inderdaad mogelijk is.



Figuur 1: De proefopzet voor het testen van de onderzoeksvraag.

Voor het testen van de onderzoeksvraag is er gebruik gemaakt van een proefopzet. De proefopzet bestaat uit vier grote zwarte blokken met daarop een witte plaat. Op de witte plaat zijn vier torentjes van legoblokjes geplaatst (ieder torentje bestaat uit drie op elkaar gestapelde rode 3 x 2 legoblokjes). De torentjes, die verder in het verslag zullen worden aangeduid als items, staan op dezelfde afstand van de rand verspreid over de lengte van de plaat. De YouBot en de vier zwarte blokken bevinden zich op een ondergrond bestaande uit vijf witte platen. Deze ondergrond zorgt ervoor dat de YouBot tijdens het rijden zo weinig mogelijk rolweerstand ondervindt. De witte kleur speelt bovendien een rol bij het identificeren van de items. Hier zal later in het verslag meer uitleg over worden gegeven. In figuur 1 is de proefopzet te zien.

Het doel van de YouBot is om bepaalde items op te pakken, deze items op zijn rug te zetten en vervolgens weg te gooien in een doos naast de startpositie. Via een interface kan worden aangegeven welke items de YouBot moet oppakken. De proef wordt gestart door via de interface aan te geven welke items de YouBot moet oppakken. Als de gebruiker in de interface bijvoorbeeld item 1, item 2 en item 4 selecteert (het eerste, het tweede en het vierde lego torentje), dan zal de YouBot eerst richting het eerste item rijden. De YouBot zal met behulp van de webcam die op de arm is bevestigd naar de positie rijden waarbij het eerste item kan worden opgepakt. Na het eerste item opgepakt te hebben, zal de YouBot deze op zijn rug plaatsen (de plaat achterop de YouBot). Vervolgens zal de YouBot richting het tweede item rijden. Met behulp van de webcam zal de YouBot naar de positie rijden waarbij het tweede item kan worden opgepakt. Na het oppakken van het tweede item zal de YouBot deze op zijn

rug plaatsen, naar de startpositie rijden en de twee items in de doos laten vallen (er kunnen maar twee items op de rug van de YouBot worden geplaatst). Hierna zal de YouBot het laatste item (het vierde item) op eenzelfde manier oppakken en verplaatsen.

2 Materiaal

Hieronder volgt een opsomming van het materiaal dat gedurende het onderzoek is gebruikt.

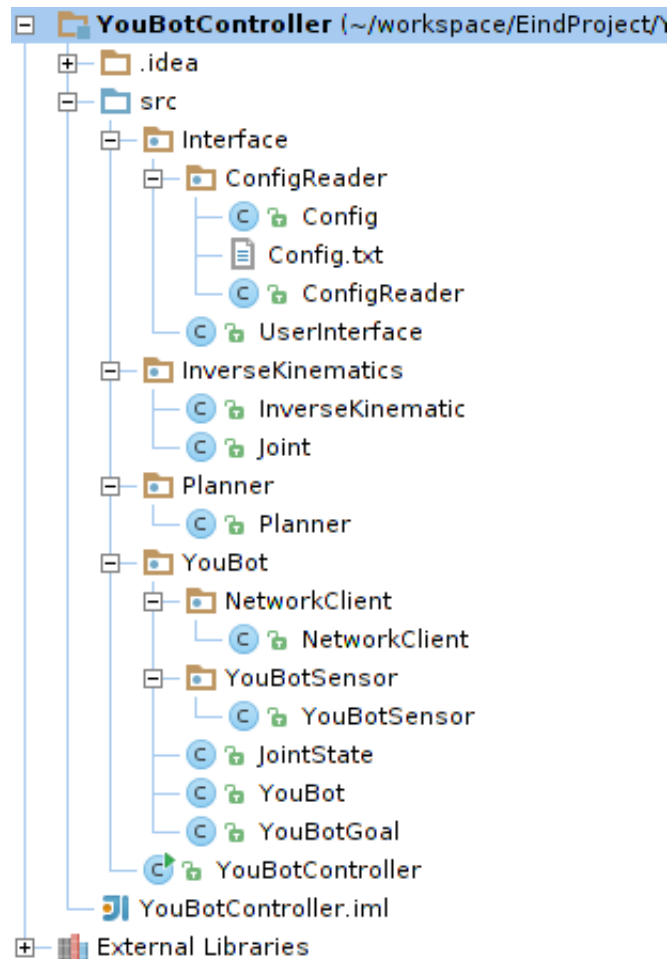
- De KUKA YouBot;
- De handleiding ‘KUKA YouBot User Manual’ [Locomotec, 2012].
- De YouBot API;
- Het programma ‘IntelliJ IDEA 12’ om de code te schrijven (zie <http://www.jetbrains.com/idea/download/>);
- Java SE Development Kit 7 (zie <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>);
- Microsoft Lifecam-HD5000 (zie <http://www.microsoft.com/hardware/nl-nl/p/lifecam-hd-5000>);
- OpenCV, de library die gebruikt wordt om kleuren en ‘blobs’ te herkennen met behulp van de webcam (zie www.opencv.org);
- Twaalf rode 3 x 2 legoblokjes om vier torentjes te maken. De torentjes fungeren als items en zullen door de YouBot worden opgepakt;
- Zes witte platen. De platen fungeren onder andere als ondergrond;
- Vier zwarte blokken. Op de zwarte blokken wordt een witte plaat gelegd waarop de items worden geplaatst.

3 Methode

Om de KUKA YouBot op kleine schaal ‘basic order picking’ uit te laten voeren, moet deze op een bepaalde manier worden geprogrammeerd. In figuur 2 is te zien hoe het programmeerwerk is onderverdeeld. Er is te zien dat het programmeerwerk is onderverdeeld in vier categorieën: de interface, de inverse kinematica, de planner en de YouBot. Hieronder wordt kort besproken wat iedere categorie precies doet. Iedere categorie zal ook afzonderlijk in meer detail worden besproken.

Via de interface kan de gebruiker aangeven welke objecten de YouBot moet ophalen. Via de interface kan het proces ook worden beëindigd voor het geval er een fout optreedt. Het inverse kinematica gedeelte bevat de code voor de inverse kinematica van de arm van de YouBot. De inverse kinematica zorgt voor de juiste configuraties van de arm aan de hand van een opgegeven doelpositie van de ‘gripper’. Het planner gedeelte bevat de code voor het op de juiste manier handelen van de YouBot. De YouBot moet op een bepaalde manier naar de juiste items bewegen, de items moeten worden opgepakt en op een bepaalde manier worden meegenomen en de items moeten uiteindelijk op een bepaalde plaats worden neergelegd. De laatste categorie, de YouBot, bevat de code voor het ‘communiceren’ met de YouBot. De commando’s om een bepaalde beweging te maken of om een bepaalde actie uit te voeren, worden via een ethernet kabel aan de YouBot doorgegeven. De code voor het doorgeven van de commando’s naar de YouBot en om variabelen van de YouBot op te vragen, bevindt zich in deze categorie.

Hieronder volgt een preciese beschrijving van iedere categorie en eventuele subcategorieën.



Figuur 2: Weergave van de verschillende categorieën waarin de code is onderverdeeld (de structuur zoals deze wordt weergegeven in het programma IntelliJ IDEA 12).

3.1 Van Java naar C++ en vice versa

Om het gebruiksgemak van Java te behouden en omdat de API van de YouBot alleen in C++ is uitgegeven, was het nodig om een interface te schrijven tussen deze twee talen. Deze interface bestaat uit een verbinding tussen een laptop (in het vervolg de controller genoemd) en de YouBot. De controller draait de Java code en de YouBot de C++ code. De YouBot functioneert in deze verbinding als een server en zal dus op een poortnummer luisteren naar commando's van de controller. Deze commando's bestaan uit strings en zien er als volgt uit:

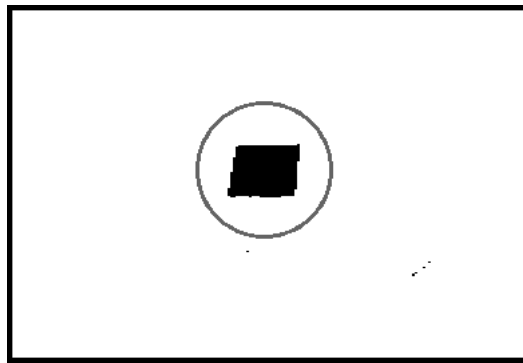
```
JOINTSTATE:joint0 : joint1 : joint2 : joint3 : joint4 : joint5 : gripper
DRIVE:speed
WEBCAM
```

Hierbij zijn 'joint0' tot en met 'joint5', 'gripper' en 'speed' variabelen. Deze commando's worden gegenereerd door de functies in het YouBot gedeelte van de Java code. Vervolgens zal de C++ code eerst de commando's splitsen op de dubbele punt en vervolgens verwerken. Zo worden de 'joint0' tot en met 'joint5' hoeken weggeschreven naar de arm van de YouBot en de 'speed' wordt weggeschreven naar de body.

3.2 Blob detection

Het andere commando dat in de voorgaande sectie wordt beschreven, WEBCAM, wordt gebruikt om een item te detecteren en de x- en y-coördinaten van het betreffende item naar de YouBot te sturen. Om een item te detecteren, zijn er een aantal stappen die moeten worden uitgevoerd. Deze stappen worden uitgevoerd met behulp van OpenCV [itseez, 2013]:

- Er wordt een nieuw beeld van de webcam opgevraagd;
- Het beeld wordt omgezet naar een beeld waarop alleen het item zichtbaar is;
- ‘Blob detection’ wordt uitgevoerd om de juiste x- en y-coördinaten te krijgen.



Figuur 3: Het resultaat wanneer de ‘blob detection’ wordt uitgevoerd.

In figuur 3 is te zien wat er overblijft van een beeld van de webcam wanneer er een item is gedetecteerd. Om het binaire plaatje te krijgen, worden bepaalde pixels wit en andere worden zwart. De pixels die zwart worden, hebben de kleur van een item (rood) en de witte pixels zijn alle andere kleuren. Aangezien de rode legoblokjes die gebruikt worden niet precies één kleur hebben, is er nogal wat variatie in de lichtomstandigheden. Om deze reden is het nodig om een range op te geven qua kleuren. Vervolgens wordt de laatste stap uitgevoerd: de ‘blob detection’. Deze geeft de coördinaten terug van de ‘blobs’ met een bepaalde grootte. In dit geval zijn de items de ‘blobs’. Om de ‘blobs’ wordt een cirkel getekend om duidelijk aan te geven dat er iets is gedetecteerd.

De exacte werking van de ‘blob detection’ is te lezen op de website van OpenCV. [itseez, 2013].

3.3 Interface

Zoals hierboven al is uitgelegd, kan de gebruiker via de interface aangeven welke items de YouBot moet ophalen. Via de interface kan het proces ook worden beëindigd voor het geval er een fout optreedt. De gebruiker kan aangeven welke items de YouBot moet ophalen door in de interface de juiste ‘checkboxes’ aan te vinken. Als de gebruiker de gewenste items heeft geselecteerd, moet er op de knop ‘Send order’ worden geklikt. De gebruiker kan dan nog eventueel andere items selecteren. Ook kan de gebruiker aangevinkte items verwijderen. Als de gebruiker zeker is van de gemaakte keuze, moet er op de knop ‘Confirm’ worden geklikt. De aangevinkte items worden dan opgeslagen in een ‘integer array’. Deze ‘integer array’ is de goal van de YouBot.

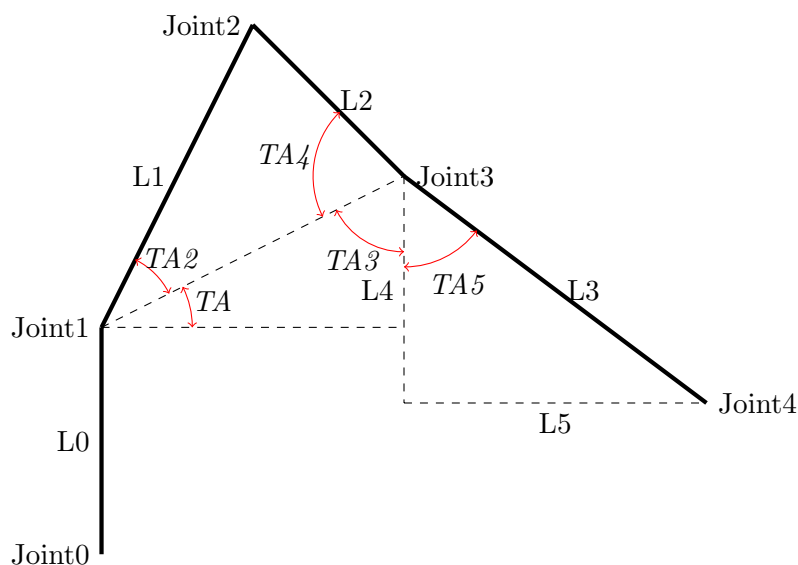
De ‘checkboxes’ worden aangemaakt met behulp van een ‘for-loop’ en worden opgeslagen in een ‘ArrayList’. Als er op de knop ‘Send order’ wordt geklikt, wordt deze niet meer zichtbaar. De knop ‘Confirm’ wordt op dat moment wel zichtbaar. Als er op deze knop wordt geklikt, wordt er gekeken welke ‘checkboxes’ zijn geselecteerd. De bijbehorende items worden opgeslagen in

een ‘ArrayList’. Als er geen ‘checkboxes’ zijn geselecteerd en de gebruiker klikt op de knop ‘Confirm’, dan verschijnt er een melding dat er minstens één item geselecteerd moet worden. Deze melding blijft zichtbaar totdat de gebruiker minstens één item heeft geselecteerd. Als de bijbehorende items in de ‘ArrayList’ zijn opgeslagen, worden de items met behulp van een ‘for-loop’ in een ‘integer array’ gezet. Deze ‘integer array’ wordt het doel van de YouBot. Als de gebruiker op de knop ‘Confirm’ heeft geklikt, wordt deze niet meer zichtbaar. De knop ‘Stop!’ wordt op dat moment wel zichtbaar. Als de gebruiker op deze knop klikt, wordt het programma beëindigd.

3.4 Inverse Kinematica

De inverse kinematica zorgt voor de juiste configuraties van de arm van de YouBot. Om ervoor te zorgen dat de arm de juiste configuraties heeft, moet het inverse kinematica probleem voor de arm van de YouBot worden opgelost. Het inverse kinematica probleem luidt als volgt: *‘Gegeven de positie van het object frame, vind de gewrichtshoeken die het daar naartoe brengen’* [Dorst, 2001].

De arm van de YouBot bestaat uit vijf links (vijf delen die afzonderlijk van elkaar kunnen bewegen). De draaiing van de ‘gripper’ wordt buiten beschouwing gelaten; deze heeft een vaste positie. Aangezien de draaiing van de ‘gripper’ buiten beschouwing wordt gelaten, wordt er voor het berekenen van de juiste ‘joint values’ (gewrichtshoeken) gerekend met vier links. De ‘gripper’ en de vierde link worden hierbij als één link beschouwd. In figuur 4 is te zien hoe de juiste ‘joint values’ zijn berekend.



Figuur 4: De methode waarmee de juiste ‘joint values’ voor de arm van de YouBot zijn berekend. De x-as loopt langs de horizontale as, de y-as loopt langs de verticale as. De coördinaten van joint0 zijn (0,0). TA staat voor temp angle.

De YouBot krijgt het x-coördinaat en het y-coördinaat mee van de positie waar naartoe bewogen moet worden. Ook krijgt de YouBot een variable hoek mee. Deze hoek is de hoek tussen de vierde link en de horizontale lijn door het punt waar naartoe bewogen moet worden (in figuur 4 de hoek tussen lijnstuk L_3 en lijnstuk L_5). In figuur 4 is te zien dat er een aantal hulplijnen zijn getekend. Deze hulplijnen dienen voor het berekenen van de ‘joint values’. L_0 , L_1 , L_2 en L_3 zijn de lengtes van de vier links.

Om te beginnen worden het x-coördinaat en het y-coördinaat van joint3 bepaald. Aangezien de lengte van de vierde link (L_3) en de hoek bekend zijn, kunnen de lengtes van de lijnstukken L_4 en L_5 worden berekend met behulp van de sinus en de cosinus van de hoek. Ook kan $tempAngle5$ worden berekend. L_3 , L_4 en L_5 vormen een rechthoekige driehoek. Het x-coördinaat van joint3 is gelijk aan het x-coördinaat van de positie waar naartoe bewogen moet worden min de lengte van lijnstuk L_5 . Het y-coördinaat van joint3 is gelijk aan het y-coördinaat van de positie waar naartoe bewogen moet worden plus de lengte van lijnstuk L_4 .

Als de coördinaten van joint3 bekend zijn, kunnen de lengtes van de andere hulplijnen worden berekend. De lengte van het lijnstuk van joint1 naar joint3 wordt als volgt berekend:

$$\sqrt{(joint3X * joint3X) + ((joint3Y - L_0) * (joint3Y - L_0))}.$$

De lengte van het lijnstuk van joint1 loodrecht op L_4 is gelijk aan het x-coördinaat van joint3. De lengte van het lijnstuk van joint3 loodrecht op het hiervoor besproken lijnstuk is gelijk aan het y-coördinaat van joint3 min L_0 . Deze twee lijnstukken vormen samen met het lijnstuk van joint1 naar joint3 een rechthoekige driehoek.

Als de lengtes van alle hulplijnen zijn berekend, kunnen de hoeken worden berekend die samen ‘joint values’ vormen (de *tempAngles*). $tempAngle$ en $tempAngle3$ kunnen worden berekend met behulp van de tanges. $tempAngle2$, $tempAngle4$ en de hoek van joint2 kunnen worden berekend met behulp van de cosinus regel. $tempAngle5$ kan als volgt worden berekend: $180 - 90 - hoek$.

Uit figuur 4 volgt dat de hoek van joint1 kan worden berekend door 90 , $tempAngle$ en $tempAngle2$ bij elkaar op te tellen. De hoek van joint2 kan, zoals hierboven al is uitgelegd, worden berekend met behulp van de cosinus regel. De hoek van joint3 kan worden berekend door $tempAngle3$, $tempAngle4$ en $tempAngle5$ bij elkaar op te tellen. De hoek van joint1 en de hoek van joint4 zijn constant.

Bovenstaande methode werkt alleen voor de situatie waarin de YouBot een item oppakt. Als de YouBot een item op zijn rug moet leggen (op de plaat achterop de YouBot), dan moet de berekening van het x-coördinaat van joint3 worden aangepast, aangezien de x-coördinaten dan negatief zijn. Het x-coördinaat van joint3 is dan gelijk aan het x-coördinaat van de positie waar naartoe bewogen moet worden *plus* de lengte van L_5 . Er is rekening gehouden met de negatieve coördinaten door met absolute waarden te rekenen.

De manier waarop het inverse kinematica probleem is opgelost, werkt niet voor de arm van de KUKA YouBot. Aangezien er niet genoeg tijd was om het probleem te achterhalen, is ervoor gekozen om de bewegingen van de arm op een ‘hardcoded’ manier te programmeren. In de discussie wordt het er verder ingegaan op deze oplossing.

3.5 Planner

Het planner gedeelte bestaat uit één while loop met als conditie dat zolang alle items niet zijn opgehaald, de loop moet worden uitgevoerd. Voordat deze loop wordt uitgevoerd, wordt er gewacht op input van de interface. Aangezien de YouBot in dit onderzoek alleen wordt blootgesteld aan zeer simpele configuratieruimtes, namelijk alle op te pakken items op één rij, is het voldoende om deze items in een vaste volgorde op te pakken. Bijvoorbeeld: wanneer de items 1, 3 en 5 moeten worden opgepakt, zal de YouBot eerst item 1 oppakken, vervolgens item 3 en daarna item 5. Wanneer de items verspreid zouden staan door de ruimte, zou er eerst nog een route berekend moeten worden en kan het zo zijn dat het efficiënter is om de items in een andere volgorde op te pakken.

Zodra de YouBot bij het eerste op te pakken item is aangekomen, zal hij de arm in de ‘hover

position' zetten. Deze positie wordt gebruikt om met behulp van de webcam te bepalen waar het item zich bevindt ten opzichte van de YouBot (in de sectie 'Blob detection' staat uitgelegd hoe een item wordt gedetecteerd en hoe de coördinaten van het item worden bepaald). Wanneer het item zich niet in het midden van het beeld bevindt, zal de YouBot kleine aanpassingen maken aan zijn positie om zo het item alsnog in het midden van het beeld te krijgen. Deze aanpassingen zijn in de vorm van rijden in de richting van het item.

Wanneer het item in het midden van het beeld is, zal de YouBot zijn arm naar de volgende positie brengen. Deze positie is net boven het op te pakken item. De reden om niet gelijk het betreffende item op te pakken, is dat de arm door de snelheid door kan slaan en het item of de ondergrond van het item kan raken. Zodra de arm zich in deze positie bevindt, zal deze zich naar de volgende positie verplaatsen om het item daadwerkelijk op te pakken.

Na het oppakken van het item zijn er twee posities op de rug van de YouBot waar het item kan worden neergezet. Zodra deze beide plekken bezet zijn, zal de YouBot terugrijden naar de startpositie en de opgepakte items afleveren. Wanneer dit gebeurd is, zal de YouBot verdergaan met het verwerken van de order totdat de gehele order is verwerkt.

3.6 YouBot

In het YouBot gedeelte bevinden zich alle methodes om manipulaties uit te kunnen voeren op de YouBot. Zo zijn er de functies *setSidewaySpeed(double speed)* en *setJointState(JointState state)*. Deze functies maken gebruik van de verbinding tussen de controller en de YouBot om de gegevens door te geven. Naast de functies om manipulaties uit te voeren, bevinden ook alle eigenschappen van de YouBot zich in dit gedeelte. Zo zijn bijvoorbeeld de lengtes van alle links van de arm op te vragen.

4 Resultaten

In de inleiding is een beschrijving gegeven van de proefopzet waarin de YouBot is getest (zie ook figuur 1). Er zijn een aantal tests uitgevoerd waarbij de YouBot telkens vier items moest oppakken (de eerste vier items). De YouBot startte tijdens het testen altijd op een bepaalde afstand van het eerste item met de arm in de standaard beginpositie. In de interface werden telkens de eerste vier items geselecteerd.

Zoals in de beschrijving van de proefopzet al is uitgelegd, staan de items op dezelfde afstand van de rand verspreid over de lengte van de plaat. De afstand tussen de items is niet helemaal willekeurig gekozen. Als de YouBot naar een item rijdt, dan is dat altijd een vaste afstand. Deze afstand wordt aan de YouBot doorgegeven. De items staan ongeveer op de plek die de YouBot bereikt na het rijden van deze vaste afstand. Als de YouBot deze afstand heeft afgelegd, beweegt de arm naar de zogenaamde 'hover position': de arm staat dan zodanig dat er met de webcam van bovenaf kan worden bepaald waar het item zich precies bevindt. Aan de hand van deze bepaling wordt de positie van de YouBot bijgesteld. De reden dat dit zo lang duurt, is omdat deze bewegingen zo klein zijn dat de YouBot deze niet altijd uitvoert. Hierdoor zal de afstand tussen de YouBot en het op te pakken item hetzelfde blijven en hetzelfde commando opnieuw worden doorgegeven net zolang totdat deze wel wordt uitgevoerd. Een ander probleem is dat de webcam een beperkt bereik heeft. Om deze reden moet de afstand tussen de positie van de YouBot en de positie van het item dat moet worden opgepakt, niet te groot zijn; het betreffende item kan dan niet met de webcam worden waargenomen en niet worden opgepakt. Aangezien de afstand die de YouBot aflegt om naar een volgend item te rijden constant is, kan de afstand tussen de items niet helemaal willekeurig worden gekozen.

Tijdens het testen werden niet altijd alle items opgepakt. In tabel 1 is voor iedere poging te zien

welke items wel en welke items niet door de YouBot werden opgepakt. Tijdens iedere poging moesten de eerste vier items worden opgepakt. Via de interface kunnen maximaal zeven items worden geselecteerd. Er is gekozen voor de eerste vier items, zodat er kon worden gekeken of de YouBot na het ophalen van de eerste twee items de andere twee items nog ophaalt. Er was niet voldoende materiaal en ruimte om de YouBot meer dan vier items op te laten halen.

Poging	Item 1	Item 2	Item 3	Item 4
1	x	-	x	x
2	x	-	-	-
3	x	x	x	-
4	x	x	x	-

Tabel 1: Een overzicht van welke items wel en welke items niet door de YouBot werden opgepakt tijdens iedere poging. Tijdens iedere poging moesten de eerste vier items worden opgepakt: x = opgepakt, - = niet opgepakt

Van iedere poging is een filmpje gemaakt. Hieronder staan de links naar de filmpjes van iedere poging:

- Poging 1: http://www.youtube.com/watch?v=Yam2dMkD_eg
- Poging 2: <http://www.youtube.com/watch?v=sQnsE5z3MHg>
- Poging 3: <http://www.youtube.com/watch?v=XheQRnMvB4o>
- Poging 4: <http://www.youtube.com/watch?v=TwpcJfopMAO>

5 Conclusie

Uit dit experiment blijkt dat ‘basic order picking’ zeker mogelijk is met een KUKA YouBot. ‘Multi-order picking’ is ook mogelijk, met als restrictie het aantal objecten dat tegelijkertijd kan worden opgehaald. Dit staat in verband met de grootte van het platform achterop de YouBot. Vanwege zijn geringe omvang en snelheid zal dit model van de YouBot niet geschikt zijn voor commerciële doeleinden.

6 Discussie

Tijdens dit experiment hebben zich een aantal problemen voorgedaan die niet direct te verhelpen waren. Deze problemen hebben betrekking op de inverse kinematica van de arm van de YouBot en het autonome gedeelte van de YouBot.

De inverse kinematica werkt nog niet naar behoren. Dit is bij de uitleg van de inverse kinematica al kort ter sprake gekomen. Het probleem zit waarschijnlijk in de berekening van de hoek van joint2; deze wordt waarschijnlijk niet correct berekend. Om deze reden komt de ‘gripper’ uiteindelijk niet op de gewenste positie te staan. Ook wordt er nog geen rekening gehouden met de y-as (links en rechts). Dit probleem is opgelost door de arm vaste waarden mee te geven wanneer er naar een item bewogen moet worden en wanneer er een item moet worden neergelegd.

Een ander probleem dat niet direct kon worden opgelost heeft betrekking op het autonome gedeelte van de YouBot. Het zou ideaal zijn als de YouBot zelf de hoogte van een item zou kunnen berekenen. Helaas is dat op dit moment nog niet mogelijk. De YouBot kan wel de x- en y-coördinaten van een item berekenen met behulp van de webcam die op de arm is bevestigd.

De hoogte (het z-coördinaat) is op dit moment niet variabel, aangezien deze niet kan worden berekend door de YouBot.

De computervision die in de YouBot is verwerkt, is niet optimaal. De webcam is geprogrammeerd om alleen de kleur rood te detecteren, maar detecteert vaak ook andere donkere objecten. Dit probleem is deels opgelost door de ondergrond wit te houden. Verder zit er een vrij grote vertraging in het beeld. Deze vertraging wordt veroorzaakt door het filter dat ervoor zorgt dat alleen het te detecteren item in het binaire plaatje voorkomt. Door deze vertraging duurt het bijstellen van de positie van de YouBot redelijk lang. Dit probleem wordt mede veroorzaakt door de specificaties van de computer die zich in de YouBot bevindt.

De YouBot heeft geen informatie over zijn omgeving. Er bevinden zich geen sensoren op of in de YouBot die de YouBot van informatie voorzien over eventuele obstakels zoals muren of andere objecten. Dit betekent dat de YouBot in een beperkt aantal situaties kan functioneren. De YouBot krijgt informatie over de afstand die gereden moet worden tot een bepaald item. Als de YouBot deze afstand heeft afgelegd, wordt er met behulp van de webcam bepaald waar het item zich precies bevindt. Aan de hand van deze bepaling wordt de positie van de YouBot bijgesteld. Het zou ideaal zijn als de YouBot zelf de gehele afstand kan bepalen die gereden moet worden om een bepaald item op te pakken.

Een ander probleem dat veroorzaakt wordt door de afwezigheid van informatie over de omgeving, heeft betrekking op de oriëntatie van de YouBot. De YouBot rijdt niet altijd in een rechte lijn wanneer deze zijwaarts rijdt. Hierdoor kan er een afwijking ontstaan met betrekking tot de afstand tot een item. Aangezien de YouBot geen informatie heeft over zijn locatie/oriëntatie, kan deze afwijking niet door de YouBot zelf worden bijgesteld. Het kan dus voorkomen dat de YouBot op een gegeven moment niet recht voor een item staat dat moet worden opgepakt, terwijl de YouBot dat wel denkt. Aangezien de bewegingen om een item op te pakken vaste bewegingen zijn, kan het voorkomen dat een item niet wordt opgepakt of verkeerd wordt opgepakt.

Door bovenstaande verbeterpunten in de YouBot en de bijbehorende code te verwerken, kan de YouBot in meerdere situaties functioneren en is deze minder afhankelijk van een bepaalde, specifieke situatie.

Referenties

- [Dorst, 2001] Dorst, L. (March 2001). *An Introduction to Robotics for the Computer Sciences*. University of Amsterdam. Department of Computer Science.
- [itseez, 2013] itseez (2013). Opencv open source computer vision. <http://opencv.org/>.
- [Locomotec, 2012] Locomotec (2012). *KUKA youBot User Manual*, 0.86.1 edition. http://youbot-store.com/downloads/KUKA-youBot_UserManual_v0.86.1.pdf.
- [Piasecki,] Piasecki, D. Order picking: Methods and equipment for piece pick, case pick, and pallet pick operations. http://www.inventoryops.com/order_picking.htm.
- [Thatcher and Easterling, 2003] Thatcher, J. L. and Easterling, A. (2003). Automated order filling method and system. Publicatienummer: US 6505093 B1, Patenteigenaar: Si Handling Systems.