

Opstel Kinematica

WoR World

Dibran Dokter 587252

Inhoudsopgave

1. Inleiding.....3

2. Ontwerp.....4

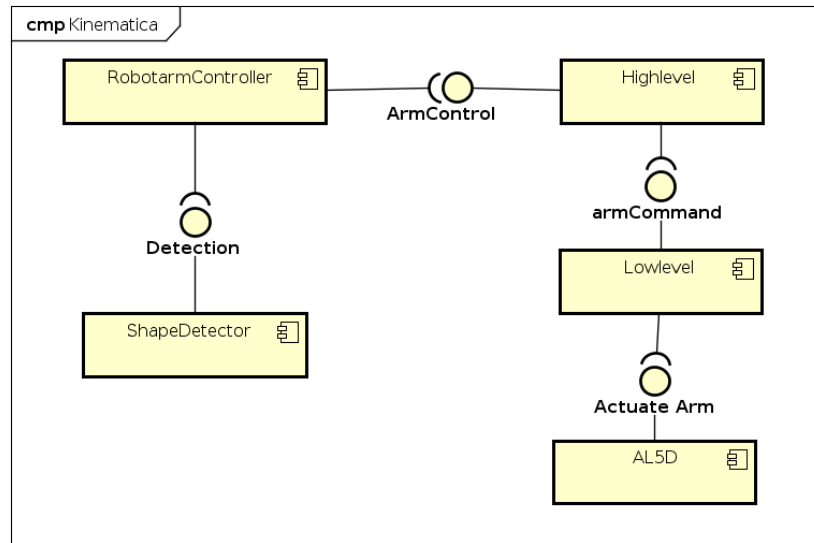
3. Bijdrage.....6

1. Inleiding

In dit document is mijn opstel voor de Kinematica opdracht van WoR Robots terug te vinden. Dit product heb ik gemaakt samen met Marnix Lukasse. De opdracht is een programma te maken welke een houten blokje kan detecteren aan de hand van ingegeven dimensies en kleur en deze op kan pakken en neerleggen in een vastgestelde plek. Om dit te kunnen doen moet er een vision gedeelte gemaakt worden welke de verschillende targets kan herkennen en de coördinaten kan berekenen. Deze coördinaten worden als input gegeven aan een implementatie van de inverse kinematica functie. Deze zal dan uiteindelijk de juiste hoeken berekenen welke doorgestuurd worden naar de robotarm.

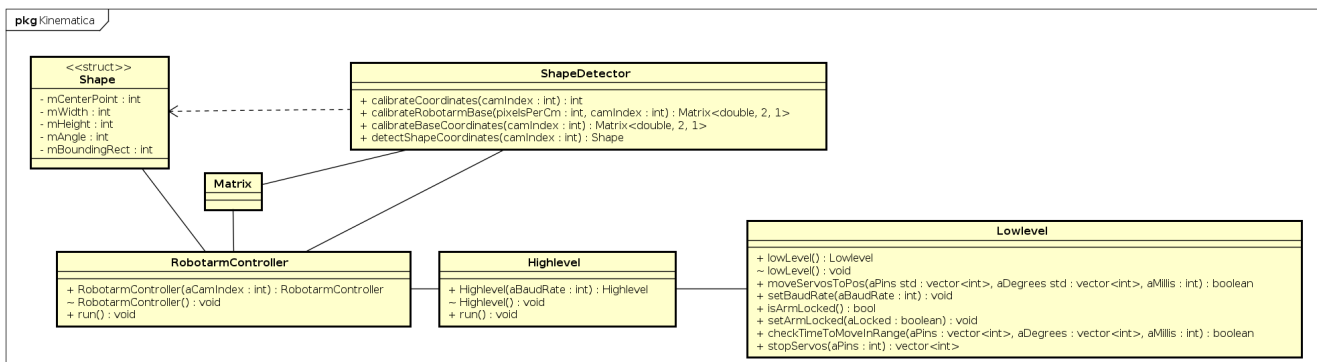
2. Ontwerp

Zoals in de inleiding benoemd bestaat dit product uit een aantal onderdelen. Bij het maken van deze onderdelen hebben we gebruik gemaakt van gemaakte producten voor andere opdrachten. Dit zorgt ervoor dat we relatief snel een werkende applicatie hebben waarbij de detectie en aansturing al gedeeltelijk werkt.



Afbeelding 1: Component Diagram

Deze componenten zijn in het component diagram terug te zien. Het shapedetector component is het stuk vision uit de OpenCV opdracht die we gebruiken om de blokken te detecteren. Deze stuurt de coördinaten van de verschillende voorwerpen door naar de robotarm controller. Dit is het component waarin de coördinaten omgezet worden naar de hoeken van de verschillende servo's. Wanneer de servo hoeken zijn berekend worden deze door middel van een ROS topic doorgestuurd naar een stuk aansturing. Dit stuk hebben we uit de interface opdracht gehaald. Dit component zet de graden van de servo's om naar de juiste pulsbreedte en stuurt het commando naar de AL5D.



Afbeelding 2: Klasse Diagram

Hierboven is het bijhorende klasse diagram te zien van de applicatie. Hierbij is de RobotarmController da alles omvattende klasse welke alle andere objecten gebruikt. De run functie is de blocking functie die gebruikt wordt om in een while loop aan te roepen en de rest van het programma uitvoert. Zo wordt er binnen deze functie een aantal functies van de shapedetector aangeroepen om de objecten te herkennen. Ook worden hier de coördinaten omgerekend naar de juiste graden van de servo's. Daarna worden de juiste aanroepen aan de highlevel driver gedaan.

Het gedeelte van de highlevel en lowlevel driver is een aangepaste versie van de interface opdracht.

Naast deze delen van het programma zijn er nog een aantal hulp klassen zoals de Matrix klasse welke gebruikt wordt om te kunnen matrix rekenen en Shape wat een wrapper is voor alle data die we nodig hebben van het object dat opgepakt moet worden.

3. Bijdrage

Zoals in de inleiding besproken doe ik deze opdracht samen met Marnix Lukasse. Echter was ik bij het begin van de opdracht ziek. Ik heb zo goed als het kon vanuit huis een opzet gemaakt voor de inverse kinematica zodat we deze als basis konden gebruiken bij het opzetten van het programma. Nadat ik me weer wat beter voelde heb ik samen met Marnix overlegd wie welk deel op zo pakken. We hebben er toen voor gekozen dat hij verder zou werken aan de inverse kinematica omdat hij dit al had uitgedacht. Ik ben toen aan de gang gegaan met het kalibreren en detecteren van de verschillende objecten. Hiernaast ben ik aan de gang gegaan met het omzetten van de gedetecteerde objecten naar coördinaten welke gebruikt kunnen worden in het inverse kinematica algoritme.

Nadat Marnix een werkend inverse kinematica algoritme had opgezet en ik de juiste data aan kon leveren uit de detectie hebben we deze twee delen samengevoegd en kregen we een aantal hoeken welke juist leken te zijn. Echter om te controleren of dit echt klopte moesten we de robotarm aansturen. We hebben er toen voor gekozen dat Marnix nog het een en ander zou verbeteren en opschonen aan het algoritme terwijl ik de interface opdracht aan ons programma bouwde om de robotarm daadwerkelijk aan te kunnen sturen.

Na deze twee dingen gemerged te hebben konden we eindelijk testen. Bij het testen bleek er nog wel wat meer calibratie nodig te zijn om het blokje succesvol op te kunnen pakken. De afwijkingen van de servo's hebben we gemeten door de servo's naar de uiterste punten te sturen en te kijken hoe de servo's stonden. Na deze afwijkingen geïmplementeerd te hebben was de robotarm accuraat tot op 2 millimeters.

Nadat de robotarm blokjes op kon pakken hebben we nog een aantal andere features ingebouwd zoals het draaien van de gripper en het kunnen ingeven van de dimensies van het blokje. Toen we alle functies in hadden gebouwd hebben we de code opgeschoond en refactored.