

# WoR-Robots Applicatie

HAN Arnhem

561399

MWJ.Berentsen@student.han.nl

Berentsen M.W.J.

3 april 2017

## Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>2</b>
<b>2</b>	<b>Applicatie design</b>	<b>2</b>
2.1	Het berekenen van het pad . . . . .	2
2.1.1	Het berekenen van de nieuwe configuratie . . . . .	2
<b>3</b>	<b>Jacobein 4 dof naar 3 dimensies</b>	<b>4</b>
<b>4</b>	<b>Verloop opdracht</b>	<b>4</b>

# 1 Inleiding

Voor de eindopdracht van WoR-Robots is er de opdracht om een applicatie te leveren die aan de hand van een simpele TUI (Text User Interface) items kan oppakken en op een opgegeven plek neerleggen. Om deze opdracht uit te kunnen voeren wordt ROS (Robot Operating System) gebruikt om de arm aan te sturen, en er wordt gebruik gemaakt van OpenCV voor de beeldherkenning. Alle code die door de student geschreven wordt zal bestaan uit C++11

## 2 Applicatie design

### 2.1 Het berekenen van het pad

Om het pad te berekenen die de robotarm moet afleggen moet er eerst een configuratie gevonden worden. Dit omdat de robot niet aangestuurd wordt met coördinaten maar hij de hoeken van de arm moet ontvangen. Vervolgens kan er met het A\* algoritme berekend wordt het pad naar deze configuratie berekend.

#### 2.1.1 Het berekenen van de nieuwe configuratie

Voor het berekenen van de configuratie word er eerst een twee dimensionaal punt bepaald vanaf het zij aanzicht. In het twee dimensionale stelsel wordt er gewerkt met coördinaten X en Y. Deze coördinaten zetten we om naar een configuratie met behulp van het volgende algoritme dit daarna per punt wordt behandeld.

```
while ( e is too far from g)
Compute  $J(e, \theta)$  for the current pose  $\theta$ 
Compute  $J^{-1}$ 
 $\Delta e = \beta(g - e)$ 
 $\Delta \theta = J^{-1} \times \Delta e$ 
 $\theta_{nieuw} = \theta_{oud} + \Delta \theta$ 
Compute new e vector
```

1. while ( e is too far from g)  
Zolang het gevonden eindpunt  $e$  nog niet ongeveer gelijk is aan het doel  $g$  wordt het volgende uitgevoerd.

Voor het berekenen van de configuratie naar een x,y coördinaat wordt het forward kinematics gebruikt. Dit levert de volgende twee formules op:

$$x = x_0 + l_1 \cdot \sin(\theta_1) + l_2 \cdot \sin(\theta_1 + \theta_2) + l_3 \cdot \sin(\theta_1 + \theta_2 + \theta_3)$$

$$y = y_0 + l_1 \cdot \cos(\theta_1) + l_2 \cdot \cos(\theta_1 + \theta_2) + l_3 \cdot \cos(\theta_1 + \theta_2 + \theta_3)$$

2. Compute  $J(e, \theta)$  for the current pose  $\theta$

De Jacobi wordt afgeleid vanuit de ingaande onafhankelijke variabelen  $x, y$  en de uitgaande afhankelijke variabelen  $\theta_1, \theta_2, \theta_3$  ofwel  $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ .

Dit levert de volgende Jacobi matrix op:  $J = \begin{bmatrix} \frac{dx}{d\theta_0} & \frac{dx}{d\theta_1} & \frac{dx}{d\theta_2} \\ \frac{dy}{d\theta_0} & \frac{dy}{d\theta_1} & \frac{dy}{d\theta_2} \end{bmatrix}$

De afgeleiden die in deze matrix staan worden op de volgende manier berekend:

$$\frac{dx}{d\theta_0} = l_1 \cdot \cos(\theta_1) + l_2 \cdot \cos(\theta_1 + \theta_2) + l_3 \cdot \cos(\theta_1 + \theta_2 + \theta_3)$$

$$\frac{dx}{d\theta_1} = l_2 \cdot \cos(\theta_1 + \theta_2) + l_3 \cdot \cos(\theta_1 + \theta_2 + \theta_3)$$

$$\frac{dx}{d\theta_2} = l_3 \cdot \cos(\theta_1 + \theta_2 + \theta_3)$$

$$\frac{dy}{d\theta_0} = -l_1 \cdot \sin(\theta_1) - l_2 \cdot \sin(\theta_1 + \theta_2) - l_3 \cdot \sin(\theta_1 + \theta_2 + \theta_3)$$

$$\frac{dy}{d\theta_1} = -l_2 \cdot \sin(\theta_1 + \theta_2) - l_3 \cdot \sin(\theta_1 + \theta_2 + \theta_3)$$

$$\frac{dy}{d\theta_2} = -l_3 \cdot \sin(\theta_1 + \theta_2 + \theta_3)$$

3. Compute  $J^{-1}$

Om de inverse van deze matrix te berekenen word de pseudo inverse gebruikt. Dit resulteert in de volgende formule:  $J^- = A^T \cdot (A \cdot A^T)^{-}$

4.  $\Delta e = \beta(g - e)$

$\Delta e$  = doel coördinaten - huidige coördinaten  $(g - e) \cdot \beta$

5.  $\Delta \theta = J^- \cdot \Delta e$

Hier wordt de inverse van de Jacobi ( $J^-$ ) uit stap 3 maal het verschil van de coördinaten ( $\Delta e$ ) uit stap 4 gedaan om het schil in de configuratie te benaderen.

6.  $\theta_{nieuw} = \theta_{oud} + \Delta \theta$

Tel hier de het verschil in de configuratie bij de oude configuratie bij op.

7. Compute new e vector

Gebruik forward kinematics om te controleren of de gevonden configuratie klopt

### 3 Verloop opdracht

Bij de start van de opdracht is er eerst besproken hoe de samenwerking moest verlopen. Daar is gekozen voor het gebruik van Github, deze keuze is gebaseerd op het feit dat het gratis is en het versie beheer bekend is voor de studenten. Vervolgens is er geïnterviewd wat er al beschikbaar was. Zo is er door Berentsen, M al een applicatie geschreven die de figuren kan herkennen en in kaart brengen en door Tunc, A een applicatie die via ROS de arm kan aansturen. Deze zijn met een lichte aanpassing direct bruikbaar in deze opdracht en worden dan ook beiden gebruikt.