

INGENIØRHØJSKOLEN AARHUS UNIVERSITET

ITROB2 - GRUPPE 13

AUTONOMOUS MOBILE ROBOTS ASSIGNMENT - 3

DATO: 18/04/2021

VEJLEDER:

PETER AHRENDT

MADS DYRMANN



Studerende:

Dorte Sølund Hansen (SWT)

Stine Falkenberg Gravesen (SWT)

Berat Kaya (SWT)

Casper Fevre Hansen (SWT)

Stud. nummer:

201808180

201608108

201610971

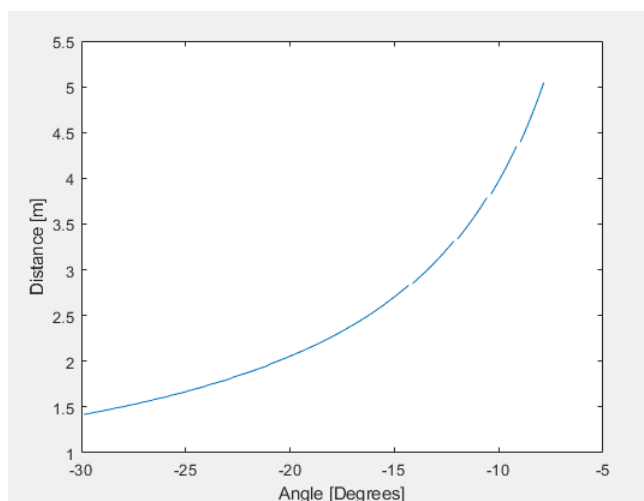
201501949

1 Wall-following

Wall-following algoritmen får turtlebotten til at køre langs en væg i en bestemt afstand (defineret som et interval på 20 cm). Hvis robotten kører for tæt på væggen, vil den dreje væk indtil den igen er indenfor intervallet, og på samme måde vil den dreje ind mod væggen, hvis den er for langt væk. I nedenstående afsnit vil dele af algoritmen forklares yderligere.

1.1 Scanningsdata

Vinkel og afstand til væggen er fundet ved at benytte scanningsdata. Som det ses i Listing 1 subscribes der til scanningsdata og afstanden kan så udtrækkes i meter, samt vinklerne i radianer.



Figur 1: Plot af vinkler i grader og afstand i meter

Det ses desuden at vinklerne omregnes til grader og at vinkler og afstand plottes i et koordinatsystem som det ses på Figur 1.

```

1  if ismember('/scan',rostopic('list'))
2      scansub = rossubscriber('/scan');
3
4      while(a_flag)
5          linescan = receive(scansub); %Receive message
6          ranges = linescan.Ranges; % Extract scan
7          angles = linescan.AngleMin:linescan.AngleIncrement:linescan.AngleMax;
8
9          angleInDegrees = rad2deg(angles);
10
11         plot(angleInDegrees, ranges)
12         xlabel('Angle [Degrees]')
13         ylabel('Distance [m]')
14         a_flag = false;
15     end
16 end

```

Listing 1: Kodeudsnit for Laserscan

Efterfølgende subscribes til scanningsdata og den modtagne besked indlæses som cartesianske koordinater som det ses i Listing 2.

```
1 scansub = rossubscriber('/scan');
2 scan = receive(scansub); %Receive message
3 cart = readCartesian(scan);
```

Listing 2: Kodeudsnit for Læsning af Cartesianske koordinater

Herefter tjekkes der i et while loop om den modtagne scanning indeholder nok data, se Listing 3. Variablen cart indeholder en matrice bestående af sæt af x- og y-koordinater, der repræsenterer væggen ved at lave punkterne fra Figur 1 om til en lige linje.

Hvis scanningen ikke indeholder nok data, roterer robotten højre om sig selv, indtil den får mindst 200 data punkter.

```
1 %Checks if scan contains enough data
2 while(length(cart) < 200)
3
4     velmsg.Angular.Z = spinVelocityRight % Angular velocity (rad/s)
5     velmsg.Linear.X = 0; % Linear velocity (m/s), +forward, -reverse
6     send(robot, velmsg);
7
8     scan = receive(scansub); %Receive message
9     cart = readCartesian(scan);
10    disp("Searching for more data-points")
11 end
```

Listing 3: Kodeudsnit for genberegning af variablen cart

Cart fittes med funktionen `fitlm`, og hældningskoefficient og skæringspunkt estimeres. Der laves en funktion med funktionsforskrift $y = ax + b$, baseret på cart-punkterne. Afstanden til væggen findes ved at beregne den korteste afstand til linjen som det ses på Listing 7 linje 21.

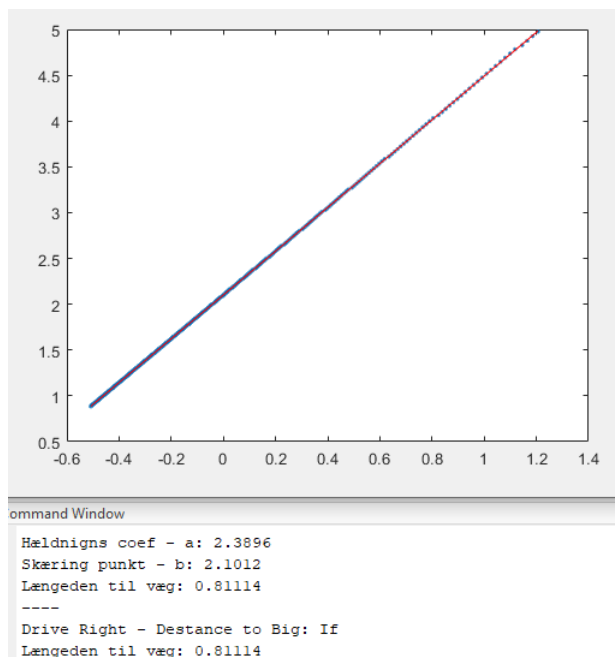
```

1      x = cart(:,2); % x-pos (all elements of the 2. column)
2      d = cart(:,1); % depth (all elements of the 1. column)
3
4      % Fitting one line..
5
6      % only left side
7      xleft = x(100:200);
8      dleft = d(100:200);
9      plot(xleft, dleft, '.')
10
11     mdl = fitlm(xleft,dleft);
12     coef=mdl.Coefficients.Estimate;
13
14     plot(x,d, '.'), hold on % Plot cartesian - Blue bullet's
15     plot(x, coef(1) + coef(2)*x, 'r') % Plot redline
16
17     a = coef(2);
18     b = coef(1);
19
20     % Compute distance of the closest obstacle
21     dist = abs(b)/(sqrt(a.^2+1));

```

Listing 4: Kodeudsnit for beregning af afstand til væg

På Figur 2 ses grafen for den fittet cart, hvor robotten står i [0;0]. Det ses, at den vinkelrette afstand til væggen i dette specifikke tilfælde er 0.81 meter. Desuden ses hældningskoefficient 'a' og skæringspunkt 'b' med y-aksen, som benyttes til beregning af afstanden til væggen, under grafen i commandWindow.



Figur 2: Scan efter cartesianske beregning

Når afstanden til væggen er beregnet, sendes en besked til robotten med hastighed og vinkelhastighed baseret på om robotten er for tæt på væggen, i den bestemte afstand fra væggen, eller for langt fra væggen.

På Listing 5 ses at når robotten er i den rigtige afstand, skal den køre ligeud med en vinkelhastighed på 0.

```

1      if ((dist > distanceThresholdMin) && (dist < distanceThresholdMax))
2          % Continue on forward path
3          velmsg.Linear.X = forwardVelocity;
4          velmsg.Angular.Z = 0;
5          disp("Drive forward")

```

Listing 5: Kodeudsnit for at få robotten til at køre lige ud

På Listing 6 ses, at når robotten er for tæt på væggen, skal den køre ligeud og samtidig dreje mod venstre. Når robotten drejer mod venstre og kigger væk fra væggen får den ikke nok scanningsdata, da den ikke kan se så langt.

For at undgå at robotten roterer om sig selv for at lede efter scanningsdata (se Listing 3) benyttes et while-loop til at tvinge robotten til at køre lige ud og lidt mod venstre. Der er valgt 1000 iterationer for while-loopet, da det viste sig at være et passende antal.

```

1      elseif(dist < distanceThresholdMin) % Distance to Small
2          disp("Drive Left")
3          velmsg.Linear.X = forwardVelocity;
4          velmsg.Angular.Z = spinVelocityLeft;
5          disp("Længden til væg: " + dist)
6
7          count = 0;
8          countMax = 1000;
9          while count < countMax
10             velmsg.Linear.X = forwardVelocity;
11             velmsg.Angular.Z = spinVelocityLeft;
12             send(robot,velmsg);
13         end

```

Listing 6: Kodeudsnit for at få robotten længere væk fra væggen

På Listing 7 ses at når robotten er for langt væk fra væggen, skal den køre ligeud og samtidig dreje med højre.

```

1      elseif(dist > distanceThresholdMax) % Distance to big
2          disp("Drive Right")
3          velmsg.Angular.Z = spinVelocityRight;
4          velmsg.Linear.X = forwardVelocity;
5          disp("Længden til væg: " + dist)
6      end

```

Listing 7: Kodeudsnit for beregning af afstand til væg

Efter en af if-sætningerne er udført og 'velmsg.Linear.X' og 'velmsg.Angular.Z' er sat bliver beskenden sendt til robotten.