EE615- Controls and Computing Lab

Experiment 2 :
Path Following Control of a ROS Robot

Submitted by

Sayali Ravindra Duragkar

203070025

Abhishek Verma

213070024

Under the guidance of

Prof. Debraj Chakraborty

Department of Electrical Engineering

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

TABLE OF CONTENTS

0.1 Objective

The objective of this experiment is to design Pure pursuit algorithm to make bot follow the correct path and implementing vector field histogram for obstacle avoidance so as to make robot choose optimal path to reach goal point by avoiding obstacles.

0.2 Pure Pursuit Algorithm

Pure pursuit is a tracking algorithm that works by calculating the curvature that will move a vehicle from its current position to some goal position. In this algorithm we choose some way-points that is ahead of our path. Our Vehicle could be thought of as chasing this point. The name is basically derived from how humans drive the vehicle. We look ahead the road and decide accordingly to change the direction of the vehicle. This point (look-ahead) changes every time we move forward.

0.2.1 Geometric Approach of this Algorithm

The pure pursuit approach is a method of geometrically determining the curvature that will drive the vehicle to a chosen way point.This goal point is a point on the path that is one look ahead distance from the current vehicle position. An arc that joins the current point and the goal point is constructed. The chord length of this arc is the look ahead distance, and acts as the third constraint in determining a unique arc that joins the two points.

Let point (x,y) be a way-point , which is a look ahead distance from the origin. The curvature of the arc that joins the origin to point (x,y) and whose chord length is l is calculated as :

$x^2 + y^2 = l^2$

$x + d = r$

$d = r - x$

$(r - x)^2 + y^2 = r^2$

$2rx = l^2$

$r = l^2/2x$

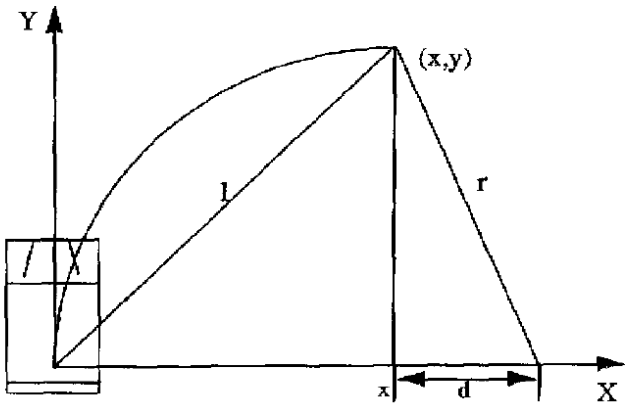


Figure 1. Geometry of the Algorithm.

0.2.2 Implementation of Pure Pursuit Algorithm

```
function [linvel,angvel,theta,i] = fcn(cpos,waypt,i)
linvel = 0.5;
angvel = 0;
theta = 0;
theta2 = 0;
c1 = 0;
c2 = 0;
xt = cpos(1,1);
yt = cpos(2,1);
theta1 = cpos(3,1);
if i < 4
    c1 = waypt(i,2) - yt;
    c2 = waypt(i,1) - xt;

    l = sqrt(c2^2 + c1^2);
    r = l^2/(2*xt);
    angvel = linvel/r;
    theta2 = atan2(c1,c2);
    %     if c1 < 0 || c2 < 0
    %         theta = theta2 - theta1;
    %     end
    %     if c1 >= 0 || c2 >= 0
    %         theta = theta1 - theta2;
    %     end
    theta = theta2 - theta1;
      if l < 0.5
         i = i+1;
      end
end
```

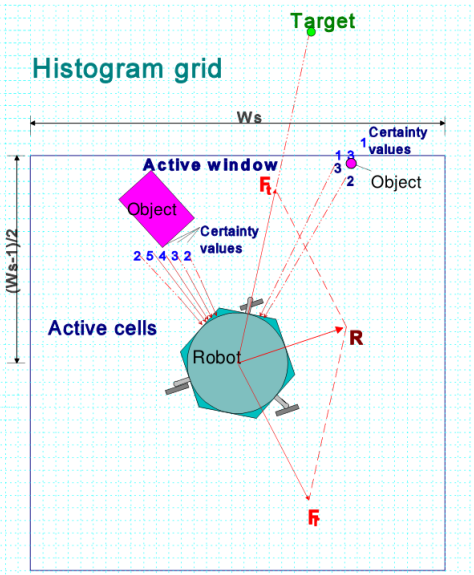
0.3 Vector Field Histogram

This method basically detects the unknown obstacles and avoids collisions while simultaneously steering the mobile robot toward the target. It uses a two-dimensional Cartesian histogram grid as a world model. The VFH method employs a two-stage data reduction technique and three levels of data representation exist:

1. The highest level holds the description of the robot's environment.The two-dimensional Cartesian histogram grid C is continuously updated in real-time with range data sampled by the on-board range sensors.

2. At the intermediate level, a 1-D polar histogram H is constructed around the robot's momentary location. H comprises n angular sectors of width a. This maps the active region C* onto H, resulting in each sector k holding a value hk that represents the polar obstacle density in the direction that corresponds to sector k.

3. The lowest level of data representation is the output of the VFH algorithm: the reference values for the drive and steer controllers of the vehicle.



0.3.1 Implementation of Vector Field Histogram Algorithm

```
function SD = fcn(Ranges,Angles,TD)
    a = 100 * ones(21,1);
    b = 20 ;
    distance = Ranges(1:21);
    Angles = Angles(1:21);
    m = ((0.89)^2) * ( a - b*distance) ;

    m_th = find(m>55) ;
    ang = find(abs(Angles(m_th)));
    if m > 55
        SD = ang(1) ;
    else
        SD = TD ;
    end

end
```

0.4 Block Diagram and Result

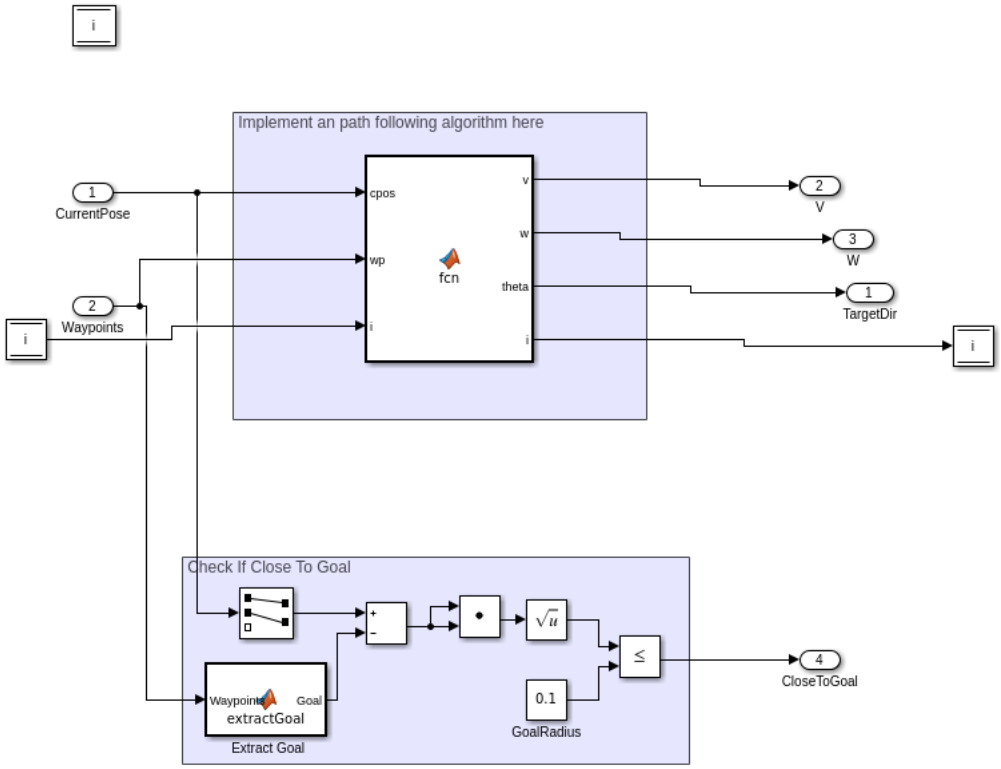


Figure 1: Pure Pursuit

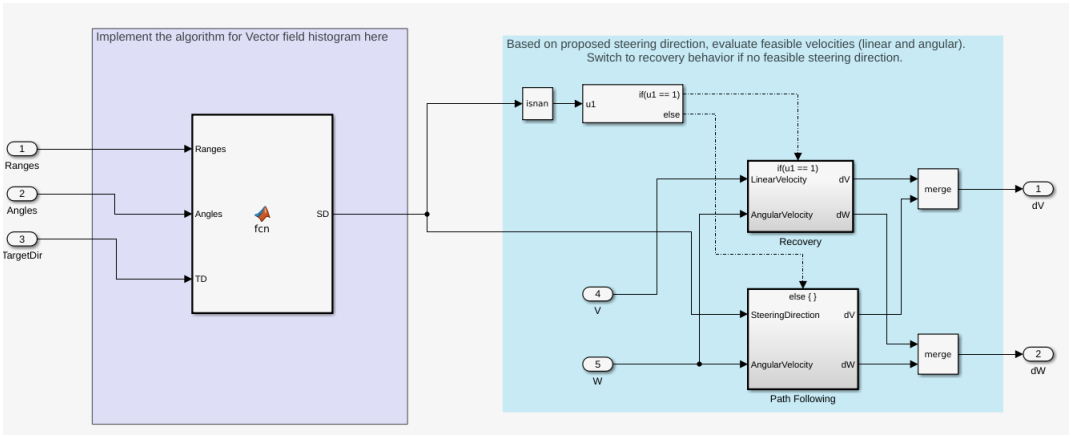


Figure 2: Vector Field Histogram

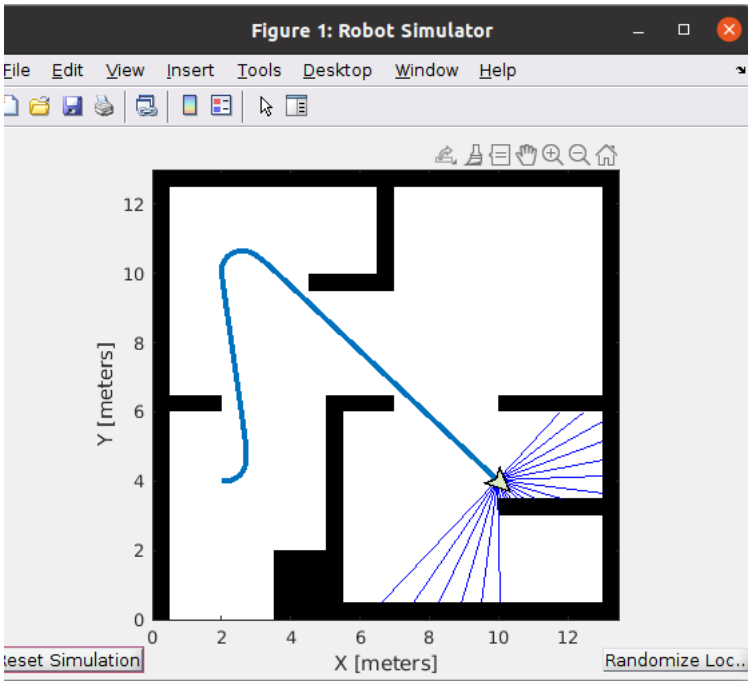


Figure 3: Result