Tutorial 4: Answers

Differential drive vehicle following waypoints using the Pure Pursuit algorithm

- For this tutorial, you need Mobile Robotics Simulation Toolbox.
- In Matlab, go to Home>Add-Ons>
- Search 'Mobile Robotics Simulation Toolbox', and add. You need to log in to your Matlab account (if you don't have an account create a one using UniSA email).
- URL for toolbox - https://au.mathworks.com/matlabcentral/fileexchange/66586-mobile-robotics-simulation-toolbox
- After installing the toolbox, click on Home>Help>
- At the bottom of the help document, find a link to Mobile Robotics Simulation Toolbox.
- Click on the link 'GettingStarted'
- This document provide links to very useful examples

1. Matlab Example: Waypoint following using the Pure Pursuit Algorithm (Differential Drive)

- Click on 'Waypoint following using the Pure Pursuit Algorithm (Differential Drive)'. This example will open in Matlab (mrsDiffDrivePurePursuit.m). Make a copy and save it as 'mrsDiffDrivePurePursuit_YOURNAME.m'
- Run the example and see the results.
- Lets understand each block of that code.

```
%% Define Vehicle
```

Go to the help documentation and click on 'Differential Drive'. It will open a live script. Using this live script, find answers for (i) What is forward kinematics? (ii) What is inverse kinematics?

```
%% Simulation parameters
```

tVec is set to stop after 15 Sec.

```
% Define waypoints
```

Define any number of waypoints. First waypoint is the start position and last one is the end position.

```
% Create visualizer
```

Go to the help documentation and click on 'Robot Visualizer'. It will open a live script. Using this live script, find answers for (i) What are the two functionalities of Robot Visualizer? (ii) Can you use it in Simulink? (iii) What is the mandatory input and how do you represent it? (iv) What are the other inputs?

In this live script, few links are provided for related examples. Check them understand the usage of Robot Visualizer.

```
%% Pure Pursuit Controller
```

You learned about this algorithm in Tutorial 3. More details can be found here, https://au.mathworks.com/help/robotics/ug/pure-pursuit-controller.html

```
%% Simulation loop
```

This is the main loop. Go though the comments and understand the loop.

Exercise:

(i) Change tVec upper limit to 35 and see what happens. Keep tVec at 35 and change the simulation loop to meet this condition – the robot should stop when it is closer to less than 0.1m to the final waypoint.

```
% answer for (i)
    dist_to_lastWP = sqrt((waypoints(end,1)-pose(1,idx))^2 +
(waypoints(end,2)-pose(2,idx))^2);
    if dist_to_lastWP<0.1
        break
    end
```

(ii) At the 100th time sample, the 4th waypoint (2,4) disappears. Modify the code to accommodate this change.

```
% answer for (ii)
    if idx==100
        waypoints = [0,0; 2,2; 4,2; 0.5,3];
        controller.Waypoints = waypoints;
    end
```

(iii) Let's make the last waypoint dynamic. Add the following code block to the end of simulation loop. Note that this is not the complete code. You need to do a minor addition to make it work.

```
% answer for (iii)
    if count<5
        waypoints(end,2) = waypoints(end,2) + 0.2;
        controller.Waypoints = waypoints;
        count = count + 1;
    end
    if count>=5
        waypoints(end,2) = waypoints(end,2) - 0.2;
        controller.Waypoints = waypoints;
        count = count + 1;
        if count==10
            count = 0;
        end
    end
```

(iv) Now modify your code ('mrsDiffDrivePurePursuit_YOURNAME.m') to do path planning instead of waypoint navigation as follows:

- Start position – (1.5,1.5), end position – (11,2)
- Use 'exampleMap' (see the next step for more info)
- When you are doing this part, comment out the codes correspond to (ii) and (iii).
- Go to the help documentation and click on 'Path planning and following of a differential drive robot'. It will open a Matlab script. Using this script, learn how to code the path planning. Copy the necessary code from this script to your script.

Codes correspond to (iv) are commented with 'answer (iv)'. Find this word and you will see the modified lines.

```
%% EXAMPLE: Differential drive vehicle following waypoints using the
% Pure Pursuit algorithm (robotics.PurePursuit)
%
```

```
% Copyright 2018 The MathWorks, Inc.

%% Define Vehicle
R = 0.1;                % Wheel radius [m]
L = 0.5;                % Wheelbase [m]
dd = DifferentialDrive(R,L);

%% Simulation parameters
sampleTime = 0.05;              % Sample time [s]
tVec = 0:sampleTime:35;          % Time array

initPose = [1.5;1.5;0];              % Initial pose (x y theta) %  %answer
(iv)-changed from [0;0;0] to [1.5;1.5;0]
pose = zeros(3,numel(tVec));     % Pose matrix
pose(:,1) = initPose;

% Define waypoints
% waypoints = [0,0; 2,2; 4,2; 2,4; 0.5,3]; %commented for answer (iv)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% added for answer (iv)
%% Path planning
% Load map and inflate it by a safety distance
close all
load exampleMap
inflate(map,R);

% Create a Probabilistic Road Map (PRM)
planner = robotics.PRM(map);
planner.NumNodes = 75;
planner.ConnectionDistance = 5;

% Find a path from the start point to a specified goal point
startPoint = initPose(1:2)';
goalPoint  = [11, 2];
waypoints = findpath(planner,startPoint,goalPoint);
show(planner)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% end answer (iv)

% Create visualizer %commented for answer (iv)
% viz = Visualizer2D;
% viz.hasWaypoints = true;

load exampleMap % Reload original (uninflated) map for visualization %added
for answer (iv)
viz = Visualizer2D; %added for answer (iv)
viz.hasWaypoints = true; %added for answer (iv)
viz.mapName = 'map'; %added for answer (iv)

%% Pure Pursuit Controller
controller = robotics.PurePursuit;
controller.Waypoints = waypoints;
controller.LookaheadDistance = 0.35;
controller.DesiredLinearVelocity = 0.75;
controller.MaxAngularVelocity = 1.5;

%% Simulation loop
% close all  %commented for answer (iv)
r = robotics.Rate(1/sampleTime);
count =0;
```

```
for idx = 2:numel(tVec)
    % Run the Pure Pursuit controller and convert output to wheel speeds
    [vRef,wRef] = controller(pose(:,idx-1));
    [wL,wR] = inverseKinematics(dd,vRef,wRef);

    % Compute the velocities
    [v,w] = forwardKinematics(dd,wL,wR);
    velB = [v;0;w]; % Body velocities [vx;vy;w]
    vel = bodyToWorld(velB,pose(:,idx-1));  % Convert from body to world

    % Perform forward discrete integration step
    pose(:,idx) = pose(:,idx-1) + vel*sampleTime;

    % Update visualization
    viz(pose(:,idx),waypoints)
    waitfor(r);


% answer for (ii)
%     if idx==100
%         waypoints = [0,0; 2,2; 4,2; 0.5,3];
%         controller.Waypoints = waypoints;
%     end

% answer for (i)
    dist_to_lastWP = sqrt((waypoints(end,1)-pose(1,idx))^2 +
(waypoints(end,2)-pose(2,idx))^2);
    if dist_to_lastWP<0.1
        break
    end

% answer for (iii)
%     if count<5
%         waypoints(end,2) = waypoints(end,2) + 0.2;
%         controller.Waypoints = waypoints;
%         count = count + 1;
%     end
%     if count>=5
%         waypoints(end,2) = waypoints(end,2) - 0.2;
%         controller.Waypoints = waypoints;
%         count = count + 1;
%         if count==10
%             count = 0;
%         end
%     end

end
```

Simulink Example: Waypoint following using the Pure Pursuit Algorithm (Differential Drive)

Go to the help documentation and click on 'Waypoint following using the Pure Pursuit Algorithm (Differential Drive)' Simulink example. A Simulink model will be opened. Run and see the results. This is the same waypoint following implementation discussed in the previous example.

Open a blank Simulink model and create a similar new model using the blocks from Simulink Library. Do not copy paste blocks from the example model, do it from the scratch. The required blocks are available in the Simulink Library, under “Mobile Robotics Simulation Toolbox”.