```
clc; close all; format compact;

addpath('../rc-labs/rc-matlab-lib');

% Only need to run once, just don't clearvars
%RC = RCCar();

% Put the robot at the origin, clear map
resetMap(RC);

%Second dimension is left and right
Waypoint = [.5 0]; % m

% How close the robot needs to be to the waypoint to be considered
% succcessful
threshold = .05; % m

% Initially, the robot is stopped with no steering angle
RC.setSteeringAngle(0);
RC.setSpeed(0);

% while the robot isn't close enough, get closer to it
while sqrt((Waypoint(1) - RC.X)^2 + (Waypoint(2) - RC.Y)^2) > threshold

    phi = RC.Phi;

    % finding WP coords in the robot frame
    RotMatrix = [cos(phi) sin(phi); -sin(phi) cos(phi)];
    local = RotMatrix * [(Waypoint(1) - RC.X); (Waypoint(2) -RC.Y)];

    % Turn radius calculations
    l_squared = local(1)^2 + local(2)^2;
    radius = l_squared / (2 * local(2));

    % Angle from the robot x axis to the WP
    localPhi = atan2(local(2), local(1));

    % Experimentally determined relationship between gamma and radius
    gamma = radius * (.25/.715);

    % If it needs to steer too hard, back up a bit to make the turn less
    % brutal

    % if the waypoint is outside of a 'cone' in front of the robot, back up
    % in a straight line
    if localPhi < -pi/4
        RC.setSpeed(-.25);
        RC.setSteeringAngle(0)
    elseif localPhi > pi/4
        RC.setSpeed(-.25);
        RC.setSteeringAngle(0)
    else
```

```
        RC.setSpeed(.25);
        RC.setSteeringAngle(gamma)
    end

    % Making sure steering angle doesn't get too large (past physical
    % limitation)
    if gamma > .25
        RC.setSteeringAngle(.25);

    elseif gamma < -.25
        RC.setSteeringAngle(-.25);
    end

    pause(.05);

end

% Turn the robot 'off'
RC.setSteeringAngle(0);
RC.setSpeed(0);
```

Published with MATLAB® R2024a