

Actividad 5 (Landmarks)

Los Landmarks son puntos de referencia importantes en un entorno, tales como edificios, monumentos, señales de tráfico, etc. En el campo de la ingeniería en robótica, los Landmarks son utilizados para permitir a los robots navegar y localizarse en un entorno desconocido. Esto se logra mediante el uso de técnicas de percepción sensorial que permiten al robot detectar y reconocer los Landmarks en su entorno. Una vez que el robot ha identificado los Landmarks, puede utilizar esta información para crear un mapa del entorno y determinar su posición relativa con respecto a los mismos. Los Landmarks son una herramienta fundamental en la ingeniería en robótica, ya que permiten a los robots moverse de manera autónoma en entornos complejos y desconocidos, lo que es esencial para aplicaciones prácticas como la exploración espacial, la navegación autónoma de vehículos y la automatización de procesos industriales.

Explicación del código;

Este código es un ejemplo que muestra cómo hacer que un vehículo robótico siga una ruta predefinida mediante el uso del algoritmo Pure Pursuit.

Primero, se definen las características del vehículo, como el tamaño de las ruedas y la distancia entre ellas. Luego se establecen los puntos de referencia que el vehículo debe seguir, y se define el controlador Pure Pursuit que ayuda al vehículo a seguir la ruta.

Después, el programa ejecuta un bucle de simulación que calcula la velocidad del vehículo y su posición en cada punto de tiempo para que pueda seguir la ruta de manera precisa. Finalmente, el programa muestra una visualización del vehículo y la ruta que sigue en la simulación.

%% EXAMPLE: Differential drive vehicle following waypoints using the

% Pure Pursuit algorithm(Trayectoria 1)

%

% Copyright 2018-2019 The MathWorks, Inc.

%% Define Vehicle

R = 0.01; % Wheel radius [m]

L = 0.5; % Wheelbase [m]

dd = DifferentialDrive(R,L);

%% Simulation parameters

sampleTime = 0.1; % Sample time [s]

tVec = 0:sampleTime:130; % Time array

initPose = [0;0;0]; % Initial pose (x y theta)

pose = zeros(3,numel(tVec)); % Pose matrix

pose(:,1) = initPose;

% Define waypoints

waypoints = [0,0; 4,4; -10,8; 7.7,-1; -7,-6; 0,5; -3,0; 2,-5; 0,0];

```

% Create visualizer
viz = Visualizer2D;
viz.hasWaypoints = true;
%% Pure Pursuit Controller
controller = controllerPurePursuit;
controller.Waypoints = waypoints;
controller.LookaheadDistance = 0.35;
controller.DesiredLinearVelocity = 0.75;
controller.MaxAngularVelocity = 1.5;
%% Simulation loop
close all
r = rateControl(1/sampleTime);
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

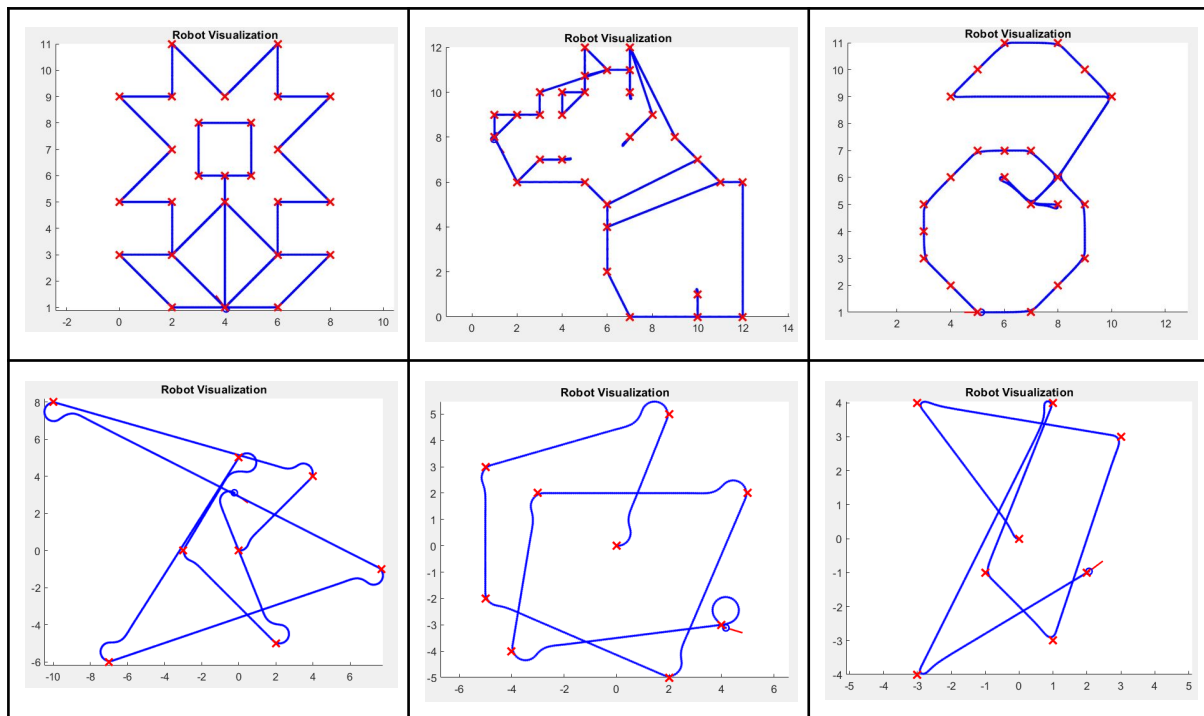
    % Perform forward discrete integration step
    pose(:,idx) = pose(:,idx-1) + vel*sampleTime;

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

```

Resultados:

Los resultados obtenidos se deben a que se variaron el número de puntos, la velocidad angular, la velocidad lineal y el tiempo de muestreo ya que para poder caracterizar estos sistemas es necesario el contar con una medida la cual nosotros consideremos adecuada ya que un ejemplo de esto es la velocidad lineal debido a que si esta la aumentamos o disminuimos podremos ver un aumento en la ganancia del sobretiro en nuestro sistema, por otra parte el número de landmarks dependerá directamente del modelo que nosotros estemos realizando siendo que en este caso el sistema con más landmarks fue el perro



Conclusión

En conclusión, el código presentado muestra un ejemplo práctico de cómo se puede utilizar el algoritmo Pure Pursuit para controlar un vehículo robótico y hacer que siga una ruta predefinida. El código ilustra la importancia de definir adecuadamente las características del vehículo y los puntos de referencia a seguir, así como la implementación del controlador para asegurar una trayectoria precisa. Además, se destaca la utilidad de la visualización para observar el progreso del vehículo en tiempo real y realizar ajustes si es necesario. En conjunto, este ejemplo muestra cómo la ingeniería en robótica puede ser utilizada para crear sistemas automatizados que sean precisos y eficientes en su desempeño.