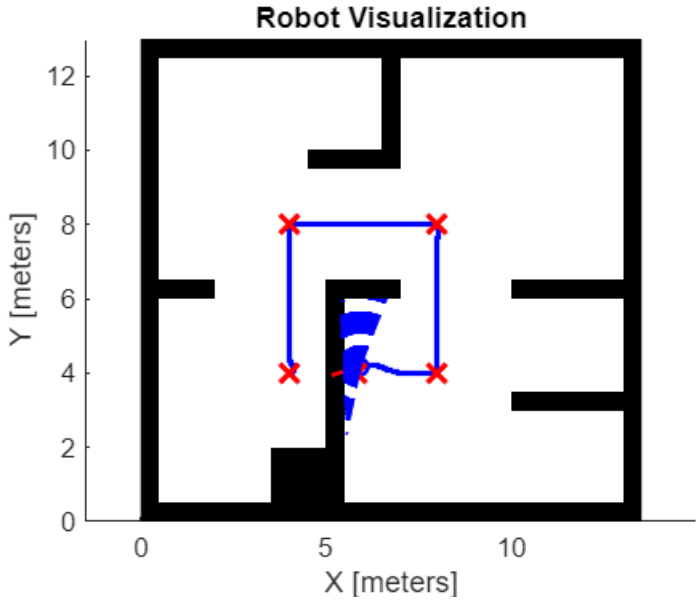
```
%% SIMULACIÓN 1: Recorrido sencillo en exampleMap

% Configuración inicial
R = 0.05; L = 0.18;
dd = DifferentialDrive(R, L);

% Tiempo y LIDAR
sampleTime = 0.1;
tVec = 0:sampleTime:80;
scanAngles = linspace(-pi, pi, 360);
maxRange = 2;

% Inicialización
initPose = [4; 4; 0];
load exampleMap

% Sensor LIDAR
lidar = LidarSensor;
lidar.sensorOffset = [0, 0];
lidar.scanAngles = scanAngles;
lidar.maxRange = maxRange;

% Visualizador
viz = Visualizer2D;
viz.hasWaypoints = true;
viz.mapName = 'map';
attachLidarSensor(viz, lidar);

% Waypoints (trayectoria en forma de U)
waypoints = [initPose(1:2)';
             4, 8;
             8, 8;
             8, 4;
             5.8, 4];

% Controlador Pure Pursuit
controller = controllerPurePursuit;
controller.Waypoints = waypoints;
controller.LookaheadDistance = 0.2;
controller.DesiredLinearVelocity = 0.5;
controller.MaxAngularVelocity = 5;

% Controlador VFH (evitación de obstáculos)
vfh = controllerVFH;
vfh.DistanceLimits = [0.05 3];
vfh.NumAngularSectors = 900;
vfh.HistogramThresholds = [5 10];
vfh.RobotRadius = L;
vfh.SafetyDistance = L;
vfh.MinTurningRadius = 0.1;
```

```
% Ciclo de simulación
pose = zeros(3, numel(tVec));
pose(:,1) = initPose;
r = rateControl(1/sampleTime);

for k = 2:numel(tVec)
    cur = pose(:,k-1);
    ranges = lidar(cur);
    [vRef, wRef, lookPt] = controller(cur);
    dirDes = atan2(lookPt(2)-cur(2), lookPt(1)-cur(1)) - cur(3);
    steer = vfh(ranges, scanAngles, dirDes);
    if ~isnan(steer) && abs(steer - dirDes) > 0.2
        wRef = 0.3 * steer;
    end
    vel = bodyToWorld([vRef; 0; wRef], cur);
    pose(:,k) = cur + vel * sampleTime;
    if norm(pose(1:2,k) - waypoints(end,:)') < 0.2, break; end
    viz(pose(:,k), waypoints, ranges);
    waitfor(r);
end
```

```
Warning: System Object 'LidarSensor' is inherited from mixin class 'matlab.system.mixin.Propagates' that
will no longer be supported. Remove 'matlab.system.mixin.Propagates' and define corresponding System object
methods instead.
Warning: System Object 'LidarSensor' is inherited from mixin class 'matlab.system.mixin.CustomIcon' that
will no longer be supported. Remove 'matlab.system.mixin.CustomIcon' and define corresponding System object
methods instead.
Warning: System Object 'Visualizer2D' is inherited from mixin class 'matlab.system.mixin.CustomIcon' that
will no longer be supported. Remove 'matlab.system.mixin.CustomIcon' and define corresponding System object
methods instead.
```

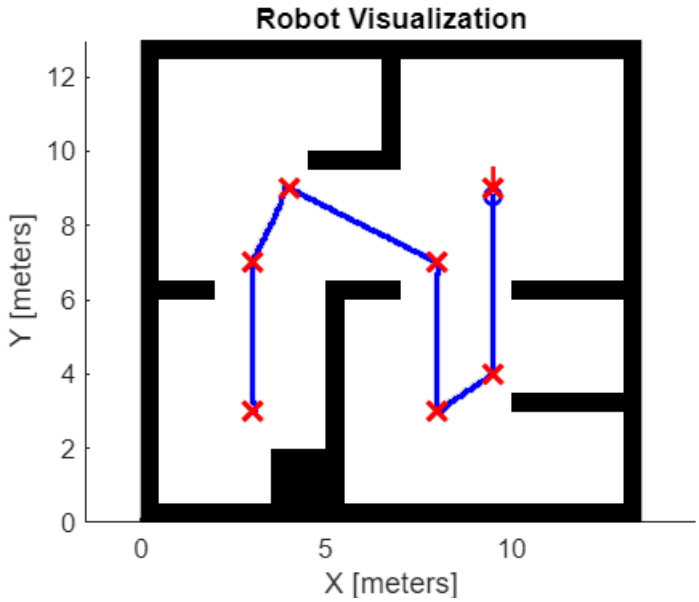



```
%% SIMULACIÓN 2: Trayectoria intermedia en exampleMap

% Inicialización
```

```
sampleTime = 0.1;
tVec = 0:sampleTime:80;
initPose = [3; 3; 0];

load exampleMap
scanAngles = linspace(-pi, pi, 360);
maxRange = 2;

% LIDAR
lidar = LidarSensor;
lidar.sensorOffset = [0, 0];
lidar.scanAngles = scanAngles;
lidar.maxRange = maxRange;

% Visualización
viz = Visualizer2D;
viz.hasWaypoints = true;
viz.mapName = 'map';
attachLidarSensor(viz, lidar);

% Waypoints (forma de 7 invertido)
waypoints = [initPose(1:2)';
             3, 7;
             4, 9;
             8, 7;
             8, 3;
             9.5, 4;
             9.5, 9];

% Controladores
controller = controllerPurePursuit;
controller.Waypoints = waypoints;
controller.LookaheadDistance = 0.2;
controller.DesiredLinearVelocity = 0.6;
controller.MaxAngularVelocity = 7;

vfh = controllerVFH;
vfh.DistanceLimits = [0.05 3];
vfh.NumAngularSectors = 900;
vfh.HistogramThresholds = [5 10];
vfh.RobotRadius = 0.18;
vfh.SafetyDistance = 0.18;
vfh.MinTurningRadius = 0.1;

% Simulación
pose = zeros(3, numel(tVec));
pose(:,1) = initPose;
r = rateControl(1/sampleTime);

for k = 2:numel(tVec)
```

```
    cur = pose(:,k-1);
    ranges = lidar(cur);
    [vRef, wRef, lookPt] = controller(cur);
    dirDes = atan2(lookPt(2)-cur(2), lookPt(1)-cur(1)) - cur(3);
    steer = vfh(ranges, scanAngles, dirDes);
    if ~isnan(steer) && abs(steer - dirDes) > 0.2
        wRef = 0.3 * steer;
    end
    vel = bodyToWorld([vRef; 0; wRef], cur);
    pose(:,k) = cur + vel * sampleTime;
    if norm(pose(1:2,k) - waypoints(end,:)') < 0.2, break; end
    viz(pose(:,k), waypoints, ranges);
    waitfor(r);
end
```

```
Warning: System Object 'LidarSensor' is inherited from mixin class 'matlab.system.mixin.Propagates' that
will no longer be supported. Remove 'matlab.system.mixin.Propagates' and define corresponding System object
methods instead.
Warning: System Object 'LidarSensor' is inherited from mixin class 'matlab.system.mixin.CustomIcon' that
will no longer be supported. Remove 'matlab.system.mixin.CustomIcon' and define corresponding System object
methods instead.
Warning: System Object 'Visualizer2D' is inherited from mixin class 'matlab.system.mixin.CustomIcon' that
will no longer be supported. Remove 'matlab.system.mixin.CustomIcon' and define corresponding System object
methods instead.
```




```
%% Simulation setup
% Define Vehicle
R = 0.1;
L = 0.5;
dd = DifferentialDrive(R,L);

% Sample time and time array
sampleTime = 0.3;
tVec = 0:sampleTime:60;
```

```
% Initial conditions
initPose = [1;2;0];
pose = zeros(3,numel(tVec));
pose(:,1) = initPose;

% Load map
close all
load exampleMap
lidar = LidarSensor;
lidar.sensorOffset = [0,0];
lidar.scanAngles = linspace(-pi,pi,250);
lidar.maxRange = 2;

% Visualizer
viz = Visualizer2D;
viz.hasWaypoints = true;
viz.mapName = 'map';
attachLidarSensor(viz,lidar);

%% Path planning and following

% Waypoints originales + puntos intermedios en el regreso
waypoints = [1 2;
             2 10;
             11 8;
             8 2;
             2 2];    % ← Punto final

% Pure Pursuit Controller
controller = controllerPurePursuit;
controller.Waypoints = waypoints;
controller.LookaheadDistance = 0.7;
controller.DesiredLinearVelocity = 0.6;
controller.MaxAngularVelocity = 6;

% VFH for obstacle avoidance
vfh = controllerVFH;
vfh.DistanceLimits = [0.05 5];
vfh.NumAngularSectors = 900;
vfh.HistogramThresholds = [5 10];
vfh.RobotRadius = L;
vfh.SafetyDistance = L;
vfh.MinTurningRadius = 0.15;

%% Simulation loop
r = rateControl(1/sampleTime);
for idx = 2:numel(tVec)
    curPose = pose(:,idx-1);
    ranges = lidar(curPose);
```

```
    [vRef,wRef,lookAheadPt] = controller(curPose);
    targetDir = atan2(lookAheadPt(2)-curPose(2),lookAheadPt(1)-curPose(1)) -
curPose(3);
    steerDir = vfh(ranges,lidar.scanAngles,targetDir);
    if ~isnan(steerDir) && abs(steerDir-targetDir) > 0.1
        wRef = 0.5*steerDir;
    end

    velB = [vRef;0;wRef];
    vel = bodyToWorld(velB,curPose);
    pose(:,idx) = curPose + vel*sampleTime;

    viz(pose(:,idx),waypoints,ranges)
    waitfor(r);
end
```

```
Warning: System Object 'LidarSensor' is inherited from mixin class 'matlab.system.mixin.Propagates' that
will no longer be supported. Remove 'matlab.system.mixin.Propagates' and define corresponding System object
methods instead.
Warning: System Object 'LidarSensor' is inherited from mixin class 'matlab.system.mixin.CustomIcon' that
will no longer be supported. Remove 'matlab.system.mixin.CustomIcon' and define corresponding System object
methods instead.
Warning: System Object 'Visualizer2D' is inherited from mixin class 'matlab.system.mixin.CustomIcon' that
will no longer be supported. Remove 'matlab.system.mixin.CustomIcon' and define corresponding System object
methods instead.
```