



Instituto Tecnológico de Estudios Superiores de Monterrey

Campus Puebla

Implementación de robótica inteligente (Gpo 501)

Actividad 1.9 (Landmarks)

Alumno

José Diego Tomé Guardado A01733345

Fecha de entrega

Lunes 6 de Mayo de 2024

1. Seguimiento de los waypoints

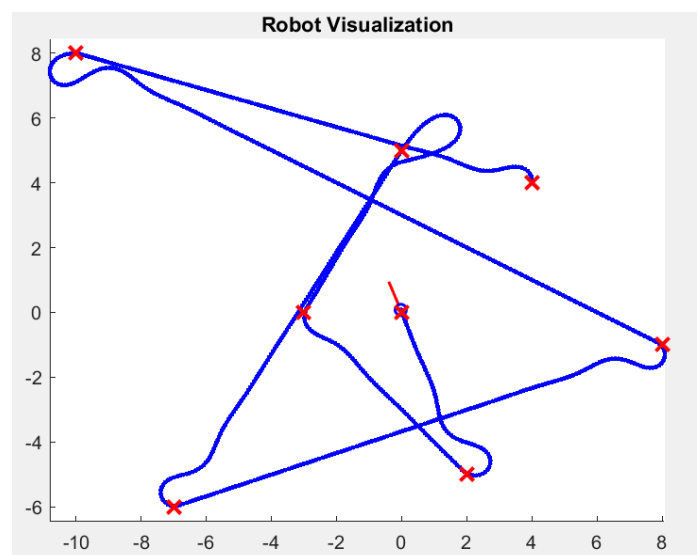
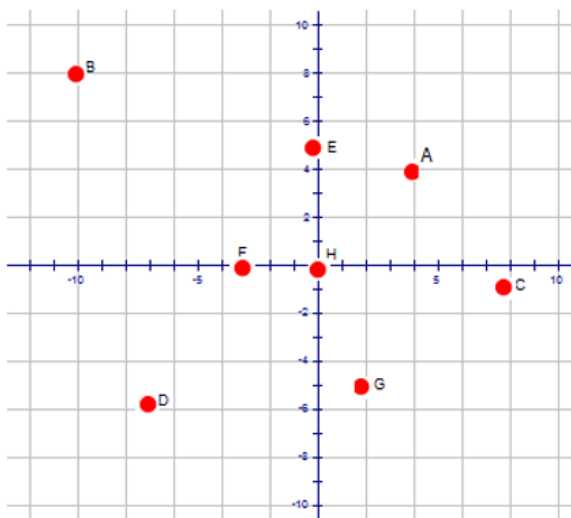
Primera trayectoria

Primeramente vamos a tener unas trayectorias que debemos de seguir conforme a los waypoints que declaremos, además de que debemos de ir cambiando los parametros de simulacion para ver cómo funciona y sobre todo para qué siga la trayectoria correctamente. Para esta primera trayectoria consideramos que tendremos los cambios de orientación con los giros que dará el robot por lo que necesitaremos qué sea rápido y tenga un buen giro para que llegue a cada uno de los puntos, Es importante resaltar que cambiamos la posición inicial conforme al ángulo debido a que necesita estar viendo del otro lado para llegar al siguiente punto B y de esta forma comienza más conciso el recorrido.

```
%% Simulation parameters
sampleTime = 0.05;           % Sample time [s]
tVec = 0:sampleTime:45.7;    % Time array
initPose = [4;4;pi/2];       % Initial pose (x y theta)
pose = zeros(3,numel(tVec)); % Pose matrix
pose(:,1) = initPose;

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

controller.LookaheadDistance = 0.4; %muy corto no verá el waypoint
controller.DesiredLinearVelocity = 2;
controller.MaxAngularVelocity = 10;
```



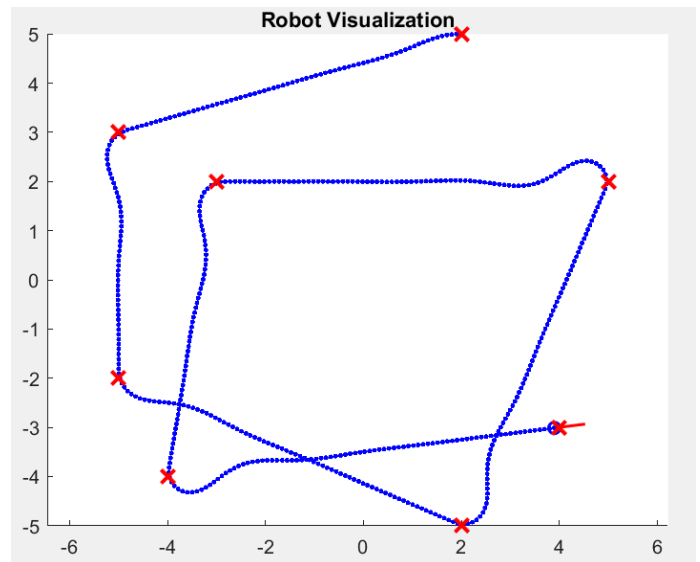
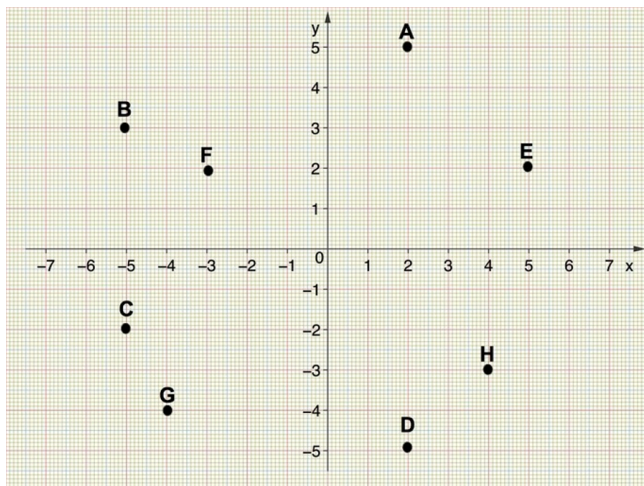
Segunda trayectoria

Para la segunda trayectoria cambiaremos los parametros principalmente del time array debido a que es un recorrido con distancias más concisas y no tiene giros tan bruscos como la primera trayectoria, es por ello que tenemos un time array más pequeño. Podemos observar que al no tener los puntos tan juntos se tiene una trayectoria más ajustada sin giros tan bruscos y es más suave la ruta, no genera algun cambio repentino. También agregamos una posición inicial de $2\pi/2$ qué basicamente son 180 grados ya que el punto B está del otro lado.

```
%% Simulation parameters
sampleTime = 0.05;           % Sample time [s]
tVec = 0:sampleTime:25.4;    % Time array
initPose = [2;5;2*pi/2];     % Initial pose (x y theta)
pose = zeros(3,numel(tVec)); % Pose matrix
pose(:,1) = initPose;

%----- Segunda trayectoria -----
waypoints = [2,5; -5,3; -5,-2; 2,-5; 5,2; -3,2; -4,-4; 4,-3];

controller.LookaheadDistance = 0.4; %muy corto no vera el waypoint
controller.DesiredLinearVelocity = 2;
controller.MaxAngularVelocity = 10;
```



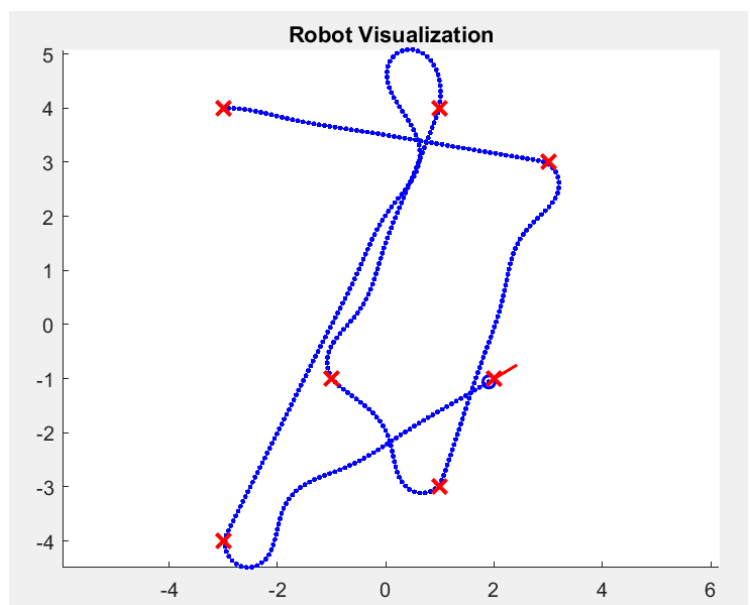
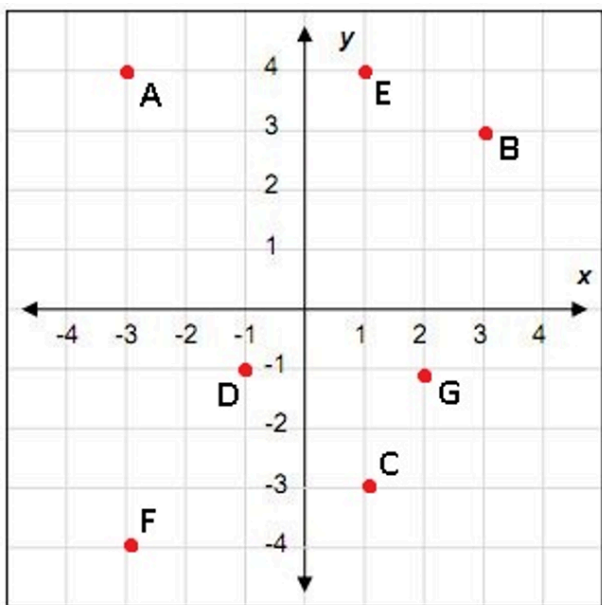
Tercera trayectoria

La última trayectoria no tuvo tampoco gran complicación debido a que utilizamos los mismos valores para el lookaheadDistance y la velocidad lineal y angular se quedaron de la misma manera ya que funcionó muy ajustado a lo que necesitaba la trayectoria y todos los cambios en el recorrido. Al tener un buen valor para el giro con la angular y un valor de velocidad lineal correcto pude darme cuenta que si lograba pasar correctamente por cada uno de los landmarks, tendría el ajuste perfecto para terminar exactamente en el punto final G. Tuvimos un recorrido también ajustado sin tener cambios bruscos solo cuando pasa de E a F hace ese giro un poco abierto igualmente por la velocidad lineal pero logró un buen recorrido.

```
%% Simulation parameters
sampleTime = 0.05;           % Sample time [s]
tVec = 0:sampleTime:19.8;    % Time array
initPose = [-3;4;0];         % Initial pose (x y theta)
pose = zeros(3,numel(tVec)); % Pose matrix
pose(:,1) = initPose;

%----- Tercera trayectoria -----
waypoints = [-3,4; 3,3; 1,-3; -1,-1; 1,4; -3,-4; 2,-1];

controller.LookaheadDistance = 0.4; %muy corto no vera el waypoint
controller.DesiredLinearVelocity = 2;
controller.MaxAngularVelocity = 10;
```



2. Seguimiento de las figuras

Figura 1. Perro

Para este caso necesitamos considerar que se hicieran buenos giros y ajustados a que formaran las líneas rectas sin hacer un giro que hiciera redondeos tan enormes cada que daba una vuelta para formar el ojo, las orejas o el hocico. Tenemos también declarado una posición inicial de $\pi/2$ ya que comencé desde el lomo del perro, de esta forma dejamos la misma velocidad angular que en las trayectorias de arriba ya que vimos que si daba giros cerrados en las trayectorias. Pero la velocidad lineal se bajó considerablemente a un valor más pequeño y esto fue debido a que se necesita un valor mayor en la angular que en la lineal y la lineal un pequeño valor para que no de tan brusco el giro si no que con la combinación de estos valores se pueda ir lentamente dando el giro logrando los picos necesarios para visualizar de forma más ajustable la figura.

```
%% Simulation parameters
sampleTime = 0.05;           % Sample time [s]
tVec = 0:sampleTime:185;     % Time array
initPose = [12;6;pi/2];      % Initial pose (x y theta)
pose = zeros(3,numel(tVec)); % Pose matrix
pose(:,1) = initPose;

%Dibujo perro
waypoints = [12,6; 11,6; 10,7; 9,8; 7,12; 8,9; 7,8; 8,9; 7,12; 7,10; 7,11;
             6,11; 5,10.5; 5,12; 6,11; 3,10; 4,10; 5,10; 4,9; 4,10; 3,10; 3,9; 2,9;
             1,9; 1,8; 2,9; 1,9; 1,8; 2,6; 3,7; 4,7; 3,7; 2,6; 5,6; 6,5; 6,4; 11,6;
             10,7; 6,5; 6,2; 7,0; 10,0; 10,1; 10,0; 12,0; 12,6]

controller.LookaheadDistance = 0.36; %muy corto no vera el waypoint
controller.DesiredLinearVelocity = 0.45;
controller.MaxAngularVelocity = 10;
```

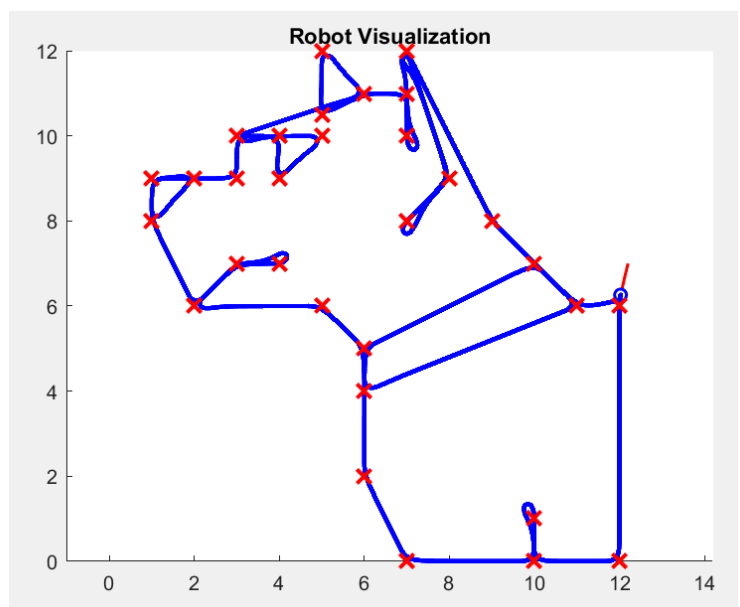
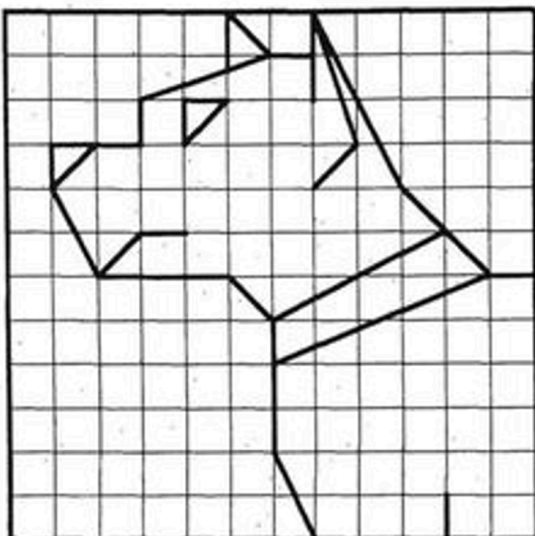


Figura 2. Flor

Siguiendo con la misma consideración de lograr giros más cerrados tuve un cambio en el valor de lookaheadDistance haciéndolo más pequeño para que el robot pasará adecuado en cada waypoint debido que con un valor más alto podía ver otros waypoints como los del cuadrado de enmedio y tenía desviaciones en su ruta. Tenemos una velocidad lineal más rápida ya que la flor no es tan detallada entonces puede hacer la trayectoria de manera más rápida que el perro y los girs los puede hacer de una mejor forma ya que el ángulo que forma no es tan pequeño como el de la figura del perro. La orientación la tenemos en $\pi/4$ ya que fue la que mejor se ajustó para seguir el segundo waypoint que estaba aumentado en y solamente.

```
% Simulation parameters
sampleTime = 0.05; % Sample time [s]
tVec = 0:sampleTime:115; % Time array
initPose = [4;1;pi/4]; % Initial pose (x y theta)
pose = zeros(3,numel(tVec)); % Pose matrix
pose(:,1) = initPose;

% ----- Dibujo flor -----
waypoints = [4,1; 4,5; 2,3; 2,5; 0,5; 2,7; 0,9; 2,9; 2,11; 4,9; 6,11; 6,9;
8,9; 6,7; 8,5; 6,5; 6,3; 8,3; 6,1; 2,1; 0,3; 2,3; 4,1; 6,3; 4,5; 4,6; 5,6;
5,8; 3,8; 3,6; 5,6]

controller.LookaheadDistance = 0.25; %muy corto no vera el waypoint
controller.DesiredLinearVelocity = 0.6;
controller.MaxAngularVelocity = 10;
```

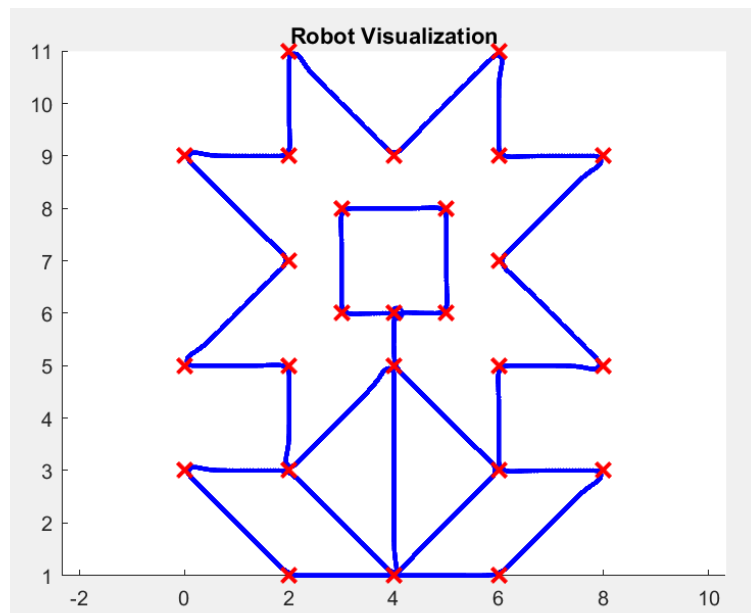
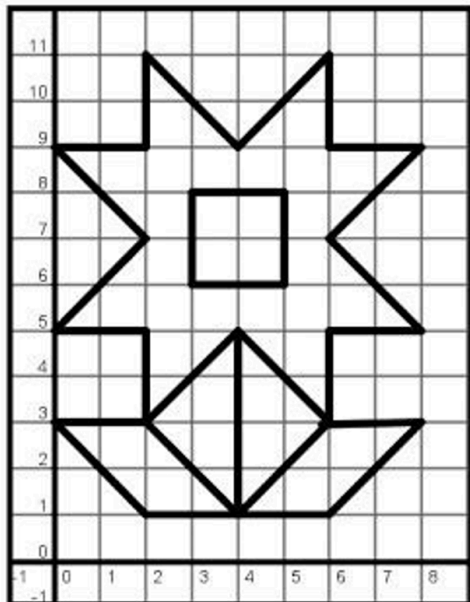


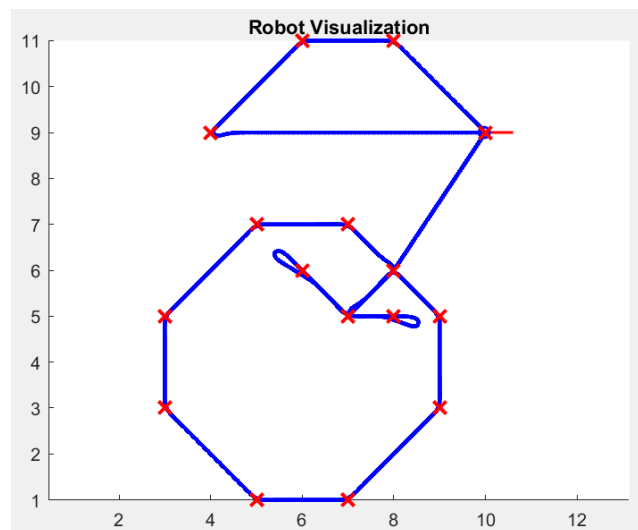
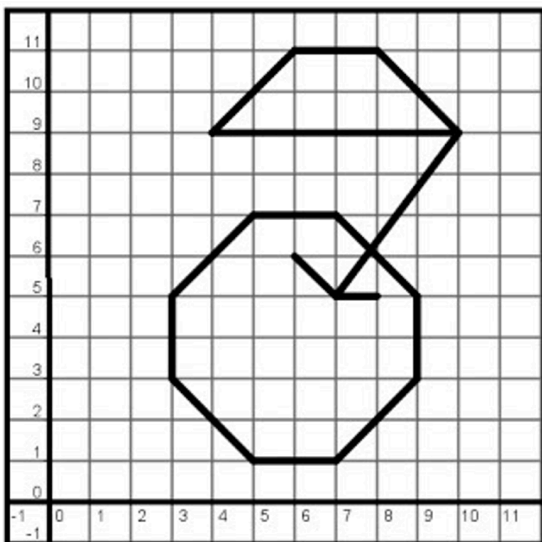
Figura 3. Cereza

Finalmente tenemos la figura de la cereza donde declaramos la orientación en $\pi/2$ debido a que comenzamos apuntando casi del otro lado y el robot tuviera un mejor comienzo para trazar el recorrido, de otra manera tenemos un valor de lookaheadDistance un poco más abajo debido a que al inicio los detalles donde esta el tallo de la cereza tenia un poco de desajuste en seguir los waypoints correctamente. Para la velocidad lineal la aumente un poco y también se aumentó la angular ya que debíamos de tener una corrección que hiciera que las líneas fueran más rectas, sobre todo que los giros tuviera prioridad para formar la figura cuadrada ya que se observa que era una figura con giros más abiertos.

```
%% Simulation parameters
sampleTime = 0.05;           % Sample time [s]
tVec = 0:sampleTime:71.5;    % Time array
initPose = [8;6;pi/2];       % Initial pose (x y theta)
pose = zeros(3,numel(tVec)); % Pose matrix
pose(:,1) = initPose;

% ----- Dibujo cereza -----
waypoints = [8,6; 7,7; 5,7; 3,5; 3,3; 5,1; 7,1; 9,3;
             9,5; 8,6; 7,5; 6,6; 7,5; 8,5; 7,5; 8,6; 10,9; 8,11; 6,11; 4,9; 10,9]

controller.LookaheadDistance = 0.22; %muy corto no vera el waypoint
controller.DesiredLinearVelocity = 0.65;
controller.MaxAngularVelocity = 27;
```



Como pudimos observar durante todo el desarrollo de la actividad, se tuvo un gran desempeño para formar cada una de las figuras, igualmente en las figuras pude observar que el time array iba disminuyendo y esto se debe a que comenzamos con la figura del perro que tenía mucho detalle y más recorrido para formar cada uno de los detalles y finalmente llegando a la cereza donde no tenemos mucho detalle solo líneas más concisas sin tanto giro ni waypoints.