



# Tecnológico de Monterrey

## **Instituto Tecnológico y de Estudios Superiores de Monterrey**

TE3002B.502

### **Implementación de robótica inteligente (Gpo 101)**

Semestre: febrero - junio 2023

#### **Actividad 3.2 (Trayectorias en lazo abierto)**

**Alumno:**

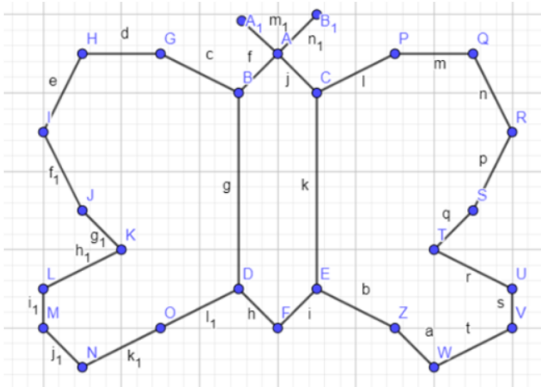
Daniel Ruán Aguilar

A01731921

**Profesor: Dr. Alfredo García Suárez**

## Objetivo.

El objetivo de esta actividad es Implementar el código requerido para generar una figura empleando trayectorias a partir de las velocidades angular y lineal en un plano 2D. En este ejercicio se trazará una **mariposa**.



## Procedimiento

```
%Limpieza de pantalla  
clear all  
close all  
clc
```

### Se declara el tiempo de ejecución

Por medio del método heurístico de prueba y error, se declaró que el tiempo final ideal sería de **55**, ya que el movimiento de carro para formar la figura contaría de **26** giros (velocidad angular) y **29** tramos de distancia (velocidad lineal), con una duración de 1 segundo cada movimiento.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
tf=55; % Tiempo de simulación en segundos (s)  
ts=0.1; % Tiempo de muestreo en segundos (s)  
t=0:ts:tf; % Vector de tiempo  
N= length(t); % Muestras
```

### Se declaran las condiciones iniciales a tomar en cuenta

```
phi(1)=0; %Orientación inicial del robot
```

```
hy(1)= y1(1); % Posición del punto de control en el eje (Y) metros (m)
```

```
disp(distancias);
```

Y de esta manera poder calcular cuál será la distancia que habrá entre cada coordenada (x,y) y poderlo incluir en la variable de velocidad lineal (v)

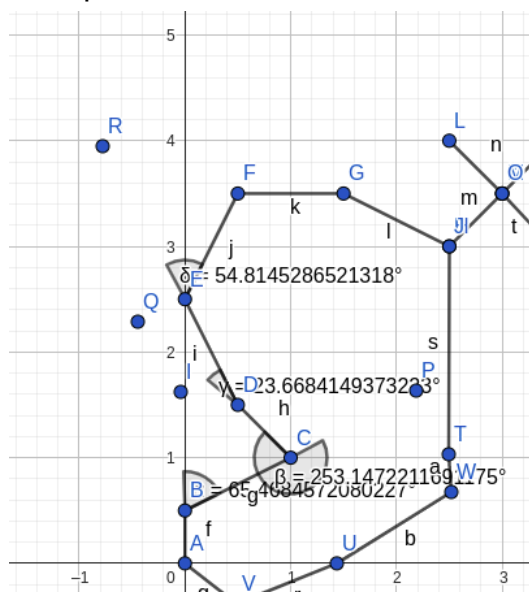
Las distancias entre los puntos son:

0.5000  
1.1180  
0.7071  
1.1180  
1.1180  
1.0000  
1.1180  
2.5000  
2.5000  
0.7071

2.5000  
2.5000  
1.1180  
1.0000  
1.1180  
1.1180  
0.7071  
1.1180  
0.5000  
0.7071  
1.1180  
1.1180  
0.7071  
0.7071  
1.1180  
1.1180  
0.7071

>> |

Para encontrar las velocidades angulares ( $w$ ) se usó la herramienta de Geogebra dónde se puede usar la herramienta de “Ángulos” para obtener los ángulos entre cada punto manualmente.



Ya con esto, se pueden ingresar los valores obtenidos a los vectores “v” y “w” multiplicándose cada valor por “ones(1,N)”, donde N es el número de muestras, haciendo un total de 55 movimientos del robot.

$$v = [\theta^*\text{ones}(1,N) \quad 0.5\theta^*\text{ones}(1,N) \quad \theta^*\text{ones}(1,N) \quad 1.118\theta^*\text{ones}(1,N) \quad \theta^*\text{ones}(1,N) \quad 0.7071\theta^*\text{ones}(1,N) \quad \theta^*\text{ones}(1,N) \quad 1.118\theta^*\text{ones}(1,N)]$$
$$w = [1.5707\theta^*\text{ones}(1,N) \quad \theta^*\text{ones}(1,N) \quad -1.142\theta^*\text{ones}(1,N) \quad \theta^*\text{ones}(1,N) \quad ((107\pi)/180)\theta^*\text{ones}(1,N) \quad \theta^*\text{ones}(1,N) \quad -0.413\theta^*\text{ones}(1,N) \quad \theta^*$$

**Se inicia bucle de simulación en el cual se declara el modelo cinemático**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BUCLE DE SIMULACION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:N

    %Aplico la integral a la velocidad angular para obtener el angulo "phi" de la orientación
    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODELO CINEMATICO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    xp1=v(k)*cos(phi(k));
    yp1=v(k)*sin(phi(k));

    %Aplico la integral a la velocidad lineal para obtener las cordenadas
    %"x1" y "y1" de la posición
    x1(k+1)=x1(k)+ ts*xp1; % Integral numérica (método de Euler)
    y1(k+1)=y1(k)+ ts*yp1; % Integral numérica (método de Euler)

    % Posicion del robot con respecto al punto de control
    hx(k+1)=x1(k+1);
    hy(k+1)=y1(k+1);

end

```

## Ahora se inicia con la simulación virtual en 3D

Para esta parte se importan los archivos compartidos por el profesor: MobilePlot.m, MobileRobot.m y Uniciclo.mat que se encuentran en la misma carpeta que este código, esto con el fin de poder graficar el entorno en 3D simultáneamente con este programa.

A continuación se muestran los pasos para la visualización:

## a) Configuración de escena

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACION VIRTUAL 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% a) Configuración de escena

scene=figure; % Crear figura (Escena)
set(scene,'Color','white'); % Color del fondo de la escena
set(gca,'FontWeight','bold'); % Negrilla en los ejes y etiquetas
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
set(scene,'position',sizeScreen); % Configurar tamaño de la figura
camlight('headlight'); % Luz para la escena
axis equal; % Establece la relación de aspecto para que las unidades de datos sean las mismas en los ejes
grid on; % Mostrar líneas de cuadrícula en los ejes
box on; % Mostrar contorno de ejes
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los ejes

view([135 35]); % Orientación de la figura
axis([-1 7 -1 5 0 2]); % Ingresar límites mínimos y máximos en los ejes x y z [minX maxX minY maxY minZ maxZ]
```

## b) Graficar robots en la posición inicial

```
% b) Graficar robots en la posición inicial
scale = 4;
MobileRobot;
H1=MobilePlot(x1(1),y1(1),phi(1),scale);hold on;
```

## c) Graficar Trayectorias

```
% c) Graficar Trayectorias
H2=plot3(hx(1),hy(1),0,'r','linewidth',2);
```

#### d) Bucle de simulación de movimiento del robot

```
step=1; % pasos para simulacion

for k=1:step:N

    delete(H1);
    delete(H2);

    H1=MobilePlot(x1(k),y1(k),phi(k),scale);
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'r','l:

    pause(ts);

end
```

### Resultado Final

