



Instituto Tecnológico de Estudios Superiores de Monterrey

Campus Puebla

Implementación de robótica inteligente (Gpo 501)

Actividad 1.6 (Trayectorias en lazo abierto)

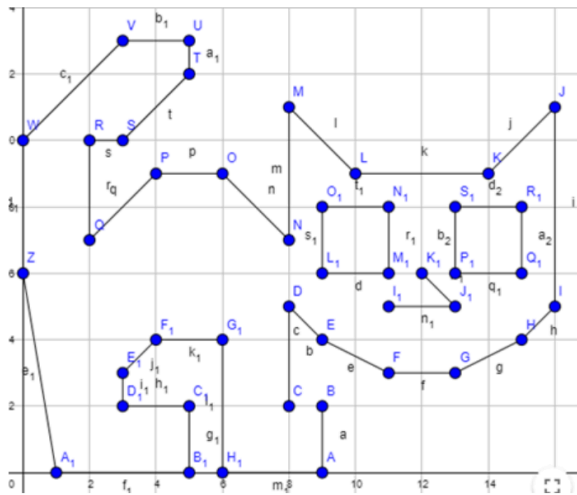
Alumno

José Diego Tomé Guardado A01733345
Pamela Hernández Montero A01736368
Victor Manuel Vázquez Morales A01736352
Fernando Estrada Silva A01736094

Fecha de entrega

Miércoles 24 de Abril de 2024

1. Código implementado: Figura Gato



Como podemos observar tenemos la primera figura del gato con la que tendremos diferentes traslaciones del robot ya sea como rotación y el avance del robot para poder formar cada línea. Tendremos el tiempo de simulación en segundos con 64 ya que van a ser la cantidad de movimientos que va a ser para los arreglos que declararemos después en las velocidades lineales y angulares para poder generar las muestras correspondientes y formar el dibujo del gato.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf = 64 ; % Tiempo de simulacion en segundos (s)
ts = 1; % Tiempo de muestreo en segundos (s)
t = 0: ts: tf; % Vector de tiempo
N = length(t); % Muestras
```

Por consiguiente tenemos las posiciones del robot definidas en $x1, y1$ donde también tendremos la orientación, es decir ϕ que será el ángulo en que se esté efectuando. Tenemos definido que la posición inicial en el eje x es de 0 y para la posición inicial para el eje y va a ser de 6 teniendo la coordenada (0,6) que es donde comenzará a hacer el trazado.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x1 = zeros (1,N+1); % Posición en el centro del eje que une las ruedas
(eje x) en metros (m)
y1 = zeros (1,N+1); % Posición en el centro del eje que une las ruedas
(eje y) en metros (m)
phi = zeros(1, N+1); % Orientacion del robot en radianes (rad)
x1(1) = 0; % Posicion inicial eje x
y1(1) = 6; % Posicion inicial eje y
phi(1) = 0; % Orientacion inicial del robot
```

Ahora nuestras velocidades de referencia que básicamente se basan en los movimientos que tendrá que realizar el robot para poder hacer nuestro trazado, comenzando primeramente por la angular para comenzar con la posición del robot correspondiente para comenzar a trazar la primera línea. Continuaremos dando más velocidades lineales y angulares para trazar cada una de las líneas hasta finalmente llegar de nuevo al punto de origen.

```

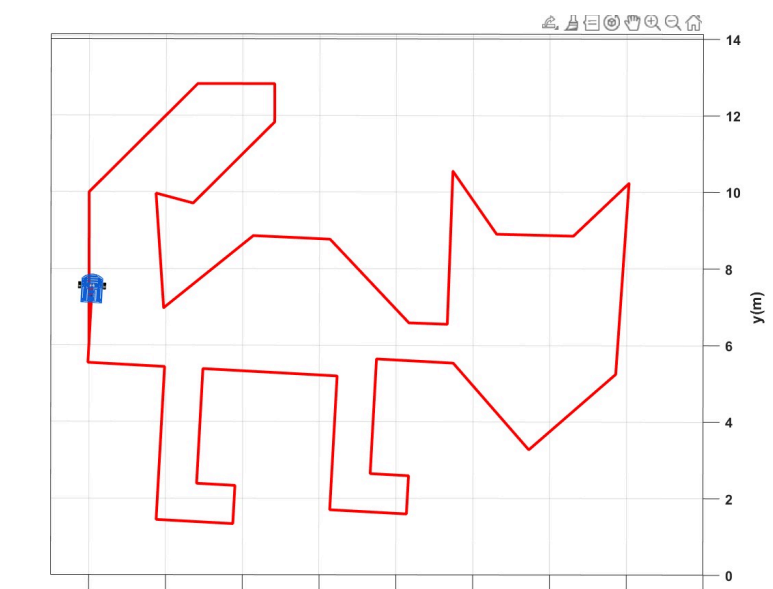
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% VELOCIDADES DE REFERENCIA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
u = [0,      4,      0,      4,      0,      2, 0,      1, 0,      3,
0,      1, 0,      3,      0,      3,      0,      2,      0,
3,      0,      1,      0,      4,      0,      2,      0,
2,      0,      2,      0,      5,      0,      3,      0,
3,      0,      2,      0,      3,      0,      1,      0,
1,      0,      2,      0,      3.5,      0,      3.5,      0,
3,      0,      1,      0,      1,      0,      2,      0,
4,      0,      2,      0,      2,0,0,1];

w = [pi/2, 0,      -pi/4, 0,      -pi/4, 0, -pi/2, 0, -pi/4, 0,
-pi/3, 0, 1.9,      0,      -4.1, 0,      -0.725, 0,      -0.77,
0,      0.78,      0,      pi/2,      0,      -2.5,      0,      0.94,      0,
pi/4,      0,      -2.4,      0,      -pi/4,      0,      -pi/2,      0,
0.8,      0,      pi/2,      0,      pi/2,      0,      -pi/2,      0,
-pi/2,      0,      -pi/2,      0,      pi/2,      0,      pi/2,      0,
pi/2,      0,      -pi/2,      0,      -pi/2,      0,      -pi/2,      0,
pi/2,      0,      -pi/2,      0,0,0,0]; %

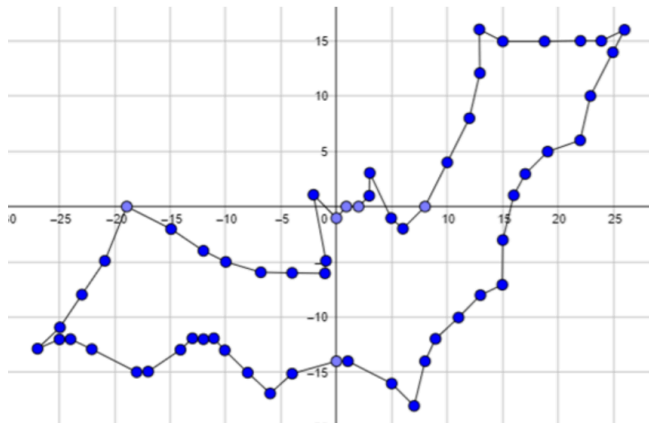
```

Simulación 3D

Por último vemos que correctamente tuvimos la trayectoria marcada por el dibujo y asemejando a lo que se nos pide para todo el trazado. Por ello también vemos que el robot llegó correctamente también a poder cerrar toda la figura y esto debido a que también tenemos un gran tiempo de muestreo que hace que pueda tener en consideración los últimos elementos del arreglo para finalizar con toda la figura.



2. Código implementado: Figura Murciélago



Ahora vamos a continuar formando la figura del murciélago donde tendremos trayectorias más marcadas, sobre todo con velocidades angulares más marcadas por la cabeza del murciélago es por ello que como podemos ver tendremos más tiempo de simulación en esta figura de 75 segundos porque tendremos que efectuar más movimientos para poder tener bien detallada la cabeza y también las alas con una buena forma que se asemeje al de la figura propuesta.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf = 75 ;                      % Tiempo de simulacion en segundos (s)
ts = 1;                        % Tiempo de muestreo en segundos (s)
t = 0: ts: tf;                 % Vector de tiempo
N = length(t);                 % Muestras
```

Siguiendo tendremos las condiciones iniciales donde basicamente expresaremos donde es que queremos que comience el trazado del robot donde inicialmente se pondra por ello tendremos que en el eje x con un 0 y la posicion inicial eje y que tendremos de -1 debido a que acá es donde mejor nos acomodamos para comenzar a declarar las velocidades lineales y angulares.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x1 = zeros (1,N+1); % Posición en el centro del eje que une las ruedas
(eje x) en metros (m)
y1 = zeros (1,N+1); % Posición en el centro del eje que une las ruedas
(eje y) en metros (m)
phi = zeros(1, N+1); % Orientacion del robot en radianes (rad)
x1(1) = 0;          % Posicion inicial eje x
y1(1) = -1;         % Posicion inicial eje y
phi(1) = 0;         % Orientacion inicial del robot
```

Por último nuestras velocidades lineales y angulares en los arreglos donde tenemos cada movimiento del robot que se efectuará para finalmente cerrar con la figura. Continuaremos dando más velocidades lineales y angulares para trazar cada una de las líneas hasta finalmente llegar de nuevo al punto de origen.

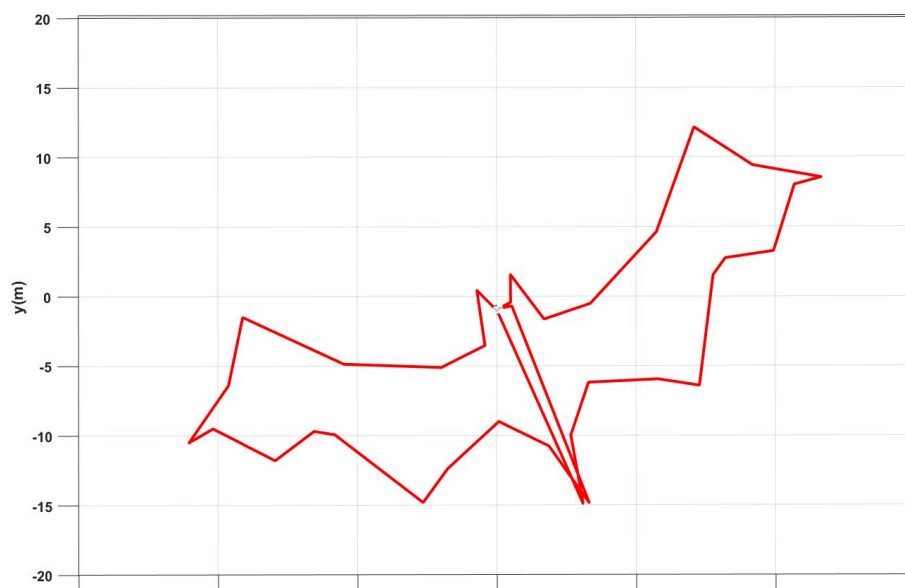
%% VELOCIDADES DE REFERENCIA %%

```
u = [0,      0,      2,      0,      4,      0,      3.5,      0,
7,      0,      8,      0,      5,      0,      5,      0,      2,
0,      5,      0,      3.5,      0,      1.5,      0,
8,      0,      3,      0,      5,      0,      4,      0,
5,      0,      15.2,      0,      1,      0,      1,
0,      0,      2,      0,      4,      0,      3.5,      0,
7,      0,      8,      0,      5,      0,      5,      0,      2,
0,      5,      0,      3.5,      0,      1.5,      0,
8,      0,      3,      0,      5,      0,      4,      0,      5,
0,      15.2,      0,      1,      0,      2.3];
```

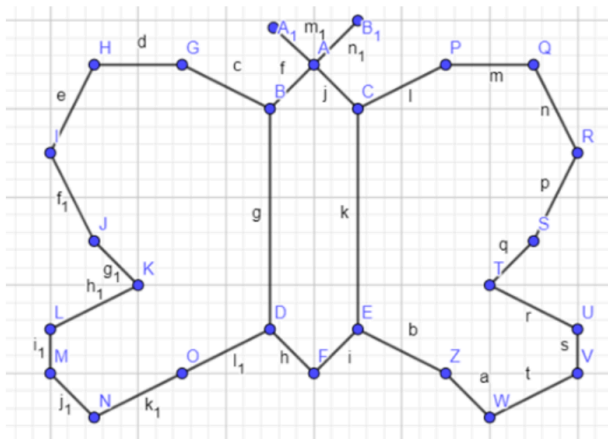
```
w = [pi/2,      pi/4,      0,      2.5,      0,      -1.25,      0,      -0.5,
0,      -0.4,      0,      1.8,      0,      -0.4,      0,      2.7,      0,
-1,      0,      1.12,      0,      -0.8,      0,      -0.5,
0,      1.6,      0,      -0.2,      0,      -1.2,      0,      -0.5,
0,      2.9,      0,      1.4,      0,      3.4,      0,
1.9,      -pi/4,      0,      -2.5,      0,      1.25,      0,      0.5,
0,      0.4,      0,      -1.8,      0,      0.4,      0,      -2.7,      0,
1,      0,      -1.12,      0,      0.8,      0,      0.5,      0,
-1.6,      0,      0.2,      0,      1.2,      0,      0.5,      0,
-2.9,      0,      1.4,      0,      3.14,      0];
```

Simulación 3D

Finalmente nuestra simulación en 3D que se logró correctamente los movimientos del robot y así poder continuar con cada línea del trazado, tuvimos algunos inconvenientes como se puede observar en medio debido a que quisimos tener un trazado más rapido de la cabeza y de la parte final de la cola y es por ello que se hicieron esas líneas demás en medio para que se pudiera ajustar de mejor manera al dibujo propuesto. También tenemos un gran tiempo de muestreo que hace que pueda tener en consideración los últimos elementos del arreglo para finalizar con toda la figura.



3. Código implementado: Figura Mariposa



Tenemos la última figura de una mariposa donde podremos ver que la parte más detallada será de la de las alas para poder tener algunos detalles bien para que se pueda ver con esa forma y también generar un buen recorrido para formar las antenas de la mariposa. Siguiendo con el tiempo tendremos de 55 segundos que significa los movimientos del robot efectuados en la figura para formar todo completamente, acá tuvimos un poco menos que el murcielago debido a que por problemas de definición y también por que en un momento comenzo a complicarse el tema de tomar los ángulos para formar esas trayectorias.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf = 55 ;                % Tiempo de simulacion en segundos (s)
ts = 1;                  % Tiempo de muestreo en segundos (s)
t = 0: ts: tf;           % Vector de tiempo
N = length(t);           % Muestras
```

Para las condiciones iniciales creimos conveniente comenzar en (0,0) para tomar estas coordenadas adecuadas y comenzar la trayectoria de toda la figura, podemos decir que de esta forma fue más sencillo comenzar a manejar las primeras rotaciones y los primeros trazados de líneas para las alas de la mariposa.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x1 = zeros (1,N+1); % Posición en el centro del eje que une las ruedas
(eje x) en metros (m)
y1 = zeros (1,N+1); % Posición en el centro del eje que une las ruedas
(eje y) en metros (m)
phi = zeros(1, N+1); % Orientacion del robot en radianes (rad)
x1(1) = 0;          % Posicion inicial eje x
y1(1) = 0;          % Posicion inicial eje y
phi(1) = 0;         % Orientacion inicial del robot
```

Tomando ahora tendremos las velocidades de referencia tenemos los movimientos del robot tanto para las velocidades lineales como angulares para formar cada una de las partes de la mariposa, sobre todo tomar en cuenta cada una de las angulares y también son importantes para formar correctamente cada parte. Teniendo igualmente en cuenta que debemos generar

que sea simétrico y es por ello que a partir de un punto comenzamos a repetir los valores de los arreglos para correctamente formar todo y generar un buen ajuste a la figura propuesta.

%% VELOCIDADES DE REFERENCIA %%%

```
u = [0,      0,      1.3,    0,      3.5,      0,      1,      0,
1,      0,      2,      0,      3.5,      0,      2.5,      0,
1,      0,      1,      0,      2.6,      0,      2.6,      0,
5.1,      0,      0,      1.3,      0,      0,      0,      0,
1.3,      0,      3.5,      0,      1,      0,      1,      0,
2,      0,      3.5,      0,      2.5,      0,      1,      0,
1,      0,      2.6,      0,      2.6,      0,      5.1,      0,
0,      0];
```

```
w = [pi/2, 0.78, 0, 1.4, 0, -1.6, 0, -0.6,
0, -0.85, 0, 1.4, 0, -1.4, 0, -0.74,
0, -0.6, 0, 1.4, 0, 3.14, 0, 0.8, 0,
0.7, 0, 0.9, pi/2, -0.78, 0, -1.4,
0, 1.6, 0, 0.6, 0, 0.85, 0, -1.4,
0, 1.4, 0, 0.74, 0, 0.6, 0, -1.4,
0, -3.14, 0, -0.8, 0, 0.7, 0,
0.9];
```

Simulación 3D

Para la simulación en 3D vemos que se hizo una gran simetría de las dos alas de la mariposa y también de las antenas pudiendo asimilar esta simulación a lo que es la figura propuesta final. Como ya habíamos indicado las alas en este caso hicimos un buen ajuste de los valores para las velocidades lineales y angulares solo que no hicimos un gran detalle debido a tomar en cuenta una rápida ejecución pero también no tener algunos errores por tomar ángulos menos precisos para cada una de las partes que conforman la ala.

