



Instituto Tecnológico de Estudios Superiores de Monterrey

Campus Puebla

Implementación de robótica inteligente (Gpo 501)

Actividad 11.1: Evaluación (Trayectoria en lazo abierto)

Alumno

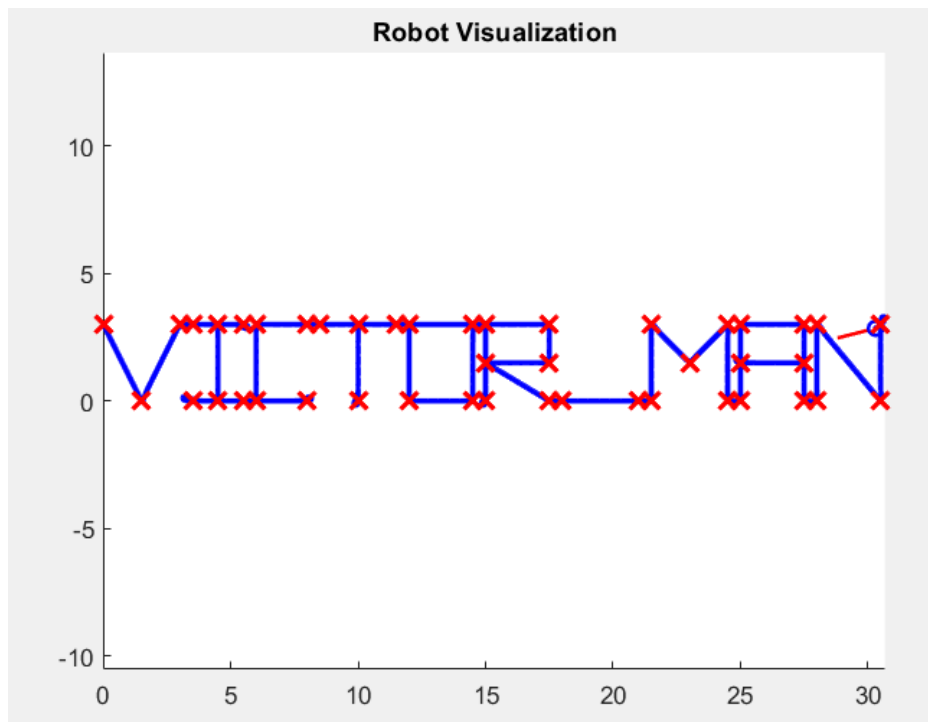
Víctor Manuel Vázquez Morales A01736352

Fecha de entrega

Viernes 17 de Mayo de 2023

1. Implementar el código requerido para generar las siguientes trayectorias a partir del tiempo y de las velocidades angulares y lineales en un plano 2D, según corresponda. La altura de cada letra debe ser de **3m.**, el ancho puede ser ajustable **a criterio propio** y la separación entre cada letra debe ser de **0.5m.**

Víctor Manuel Vázquez: **VICTOR_MAN**



Para atacar este problema, se optó por resolverlo a través de landmarks, declarándose de manera estratégica para que formaran cada una de las letras y, de igual forma, podemos observar que ciertas letras con curvatura se implementaron ajustando su forma a una más recta o cuadrada. A continuación se describen los pasos para realizar la trayectoria:

1. Definimos un vector de tiempo considerablemente alto para que pueda realizar la simulación de cada letra:

```
sampleTime = 0.05; % Sample time [s]
tVec = 0:sampleTime:200;
```

2. Declaramos la posición inicial, que en esta caso viene siendo 0,3 (un vértice de la letra V) y ajustamos la orientación inicial de tal forma que se encuentre viendo al próximo punto.

```
initPose = [0;3;-pi/4];
```

3. Declaramos los landmarks necesarios para trazar cada letra. En algunos casos, se repiten landmarks con el fin de generar una trayectoria más estética y con menos giros u oscilaciones.

```
waypoints = [0,3; 1.5,0; 3,3; %V
3.5,3; 5.5,3; 4.5,3; 4.5,0; 3.5,0; 5.5,0; %I
8,0; 6,0; 6,3; 8,3; %C
8.5,3; 10,3; 10,0; 10,3;11.5,3; %T
12,3; 14.5,3; 14.5,0; 12,0; 12,3; %O
15,3; 15,0; 15,3; 17.5,3; 17.5, 1.5; 15,1.5; 17.5,0; %R
18,0; 21,0; %_
21.5,0; 21.5,3; 23,1.5; 24.5,3; 24.5,0; %M
25,0; 25,3; 27.5,3; 27.5,1.5; 25,1.5; 27.5,1.5; 27.5, 0; %A
28,0; 28,3; 30.5,0; 30.5,3 %N
];
```

4. Por último, se declaran las velocidades, así como el LookAheadDistance. Para este ejercicio, observemos que nuestra velocidad lineal es pequeña en comparación con la velocidad angular.

```
controller = controllerPurePursuit;
controller.Waypoints = waypoints;
controller.LookaheadDistance = 0.2;
controller.DesiredLinearVelocity = 0.53;
controller.MaxAngularVelocity = 10;
```

2. Responder las siguientes preguntas en base al procedimiento empleado para obtener las trayectorias propuestas:

a) ¿Cuál fué el o los parámetros que se modifican para obtener una trayectoria recta? ¿Porqué?

Para este caso, podemos observar en la solución implementada que nuestro trazado para cada una de las letras es recto, presentando muy pocas oscilaciones o curvas que afectarán en la visualización final. Para lograr esto, fue necesario hacer que la velocidad angular máxima del robot fuera mucho mayor a la velocidad lineal, ya que de esta forma, al detectar un error muy grande entre un landmark primero corregirá el ángulo antes de avanzar (o al menos así es como se interpreta visualmente hablando). Para este ejercicio, las velocidades fueron las siguientes:

```
controller.DesiredLinearVelocity = 0.53;
controller.MaxAngularVelocity = 10;
```

b) ¿Cuál fué el o los parámetros que se modifican para obtener una trayectoria curva? ¿Porqué?

En este caso, optamos por representar ciertas letras con curvatura de una forma más recta para simplificar el problema. Sin embargo, en dado caso que deseamos trazar letras como la S,O,C o U sería necesario hacer todo lo contrario a lo que hicimos para nuestro ejercicio, es decir, sería necesario que la velocidad lineal de nuestro robot fuera mayor a la velocidad angular máxima del mismo.

Ahora bien, muy independientemente de la relación que exista entre las velocidades (que tan mayor es una con respecto a la otra), para realizar una buena aproximación a una letra con curvas sería mejor opción implementar una gran cantidad de landmarks continuos que permitan el trazado de la letra.

c) ¿Cuál fué el o los parámetros que se modifican para obtener un giro? ¿Porqué?

Para la generación de nuestra trayectoria podemos observar que en algunas ocasiones, entre landmark y landmark es necesario realizar giros bruscos o de gran magnitud. Previamente, ya se ha dado respuesta a este pregunta, y es que para lograr que nuestro robot realice buenos giros fue necesario hacer que la velocidad lineal fuera mucho menor a la velocidad máxima angular:

```
controller.DesiredLinearVelocity = 0.53;  
controller.MaxAngularVelocity = 10;
```

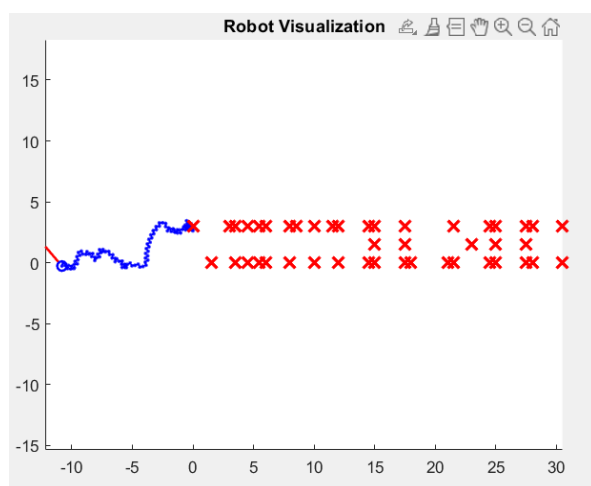
De esta forma, garantizamos que nuestro robot dé prioridad a la corrección de ángulos, provocando que realice buenos giros entre cada uno de los landmarks.

d) ¿Qué papel desempeña el vector del tiempo en la generación de la trayectoria?

Previo a responder esta pregunta, recordemos como se declara este vector:

```
sampleTime = 0.05;  
tVec = 0:sampleTime:200;
```

Este vector de tiempo se compone del tiempo inicial, periodo de muestreo y tiempo final. Prestaremos atención principalmente en el periodo de muestreo, ya que es esta variable (sampleTime) la que define la precisión de la simulación. Esto se debe a que, al tener un tiempo de muestreo mayor, los intervalos crecen provocando una simulación con errores y poco precisa, por lo que resulta mejor utilizar valores pequeños para esta variable. A continuación anexamos la simulación generada al usar un periodo de 0.5:



Podemos comprobar con esta imagen, que es necesario ajustar a un valor menor el periodo de muestreo, aunque esto también depende de la trayectoria deseada.

Por otro lado, no está demás mencionar que el tiempo final (200 en este caso) simplemente define la duración de la simulación. Este valor se debe cuidar de igual forma ya que un tiempo menor al adecuado no le permitirá al robot generar toda la trayectoria y un tiempo mayor hará que el robot, al finalizar la trayectoria, se pierda debido a que ya no hay más landmarks a los que dirigirse.

e) ¿Cuáles fueron los parámetros que se ajustaron para obtener las dimensiones de las trayectorias deseadas?

Para este caso, fue necesario realizar modificaciones en los landmarks o waypoints, definiéndoles valores que permitan el trazado de la trayectoria con las dimensiones especificadas. Para comprender mejor esto podemos observar los waypoints declarados para esta trayectoria:

```
waypoints = [0,3; 1.5,0; 3,3; %V
            3.5,3; 5.5,3; 4.5,3; 4.5,0; 3.5,0; 5.5,0; %I
            8,0; 6,0; 6,3; 8,3; %C
            8.5,3; 10,3; 10,0; 10,3;11.5,3; %T
            12,3; 14.5,3; 14.5,0; 12,0; 12,3; %O
            15,3; 15,0; 15,3; 17.5,3; 17.5, 1.5; 15,1.5; 17.5,0; %R
            18,0; 21,0; %_
            21.5,0; 21.5,3; 23,1.5; 24.5,3; 24.5,0; %M
            25,0; 25,3; 27.5,3; 27.5,1.5; 25,1.5; 27.5,1.5; 27.5, 0; %A
            28,0; 28,3; 30.5,0; 30.5,3 %N
];
```

Recordemos, por ejemplo, que para este caso se nos pidió realizar letras con una altura de 3 m, por lo que podemos observar en nuestro vector waypoints que los valores de y se encuentran delimitados entre 0 y 3.