Instituto Tecnológico de Estudios Superiores de Monterrey
Ingeniería en Robótica y Sistemas Digitales

Implementación de robótica inteligente TE 3002 B.501

Profesor: Dr. Hernán Abaunza González

“Reporte de la Actividad 1: Robot móvil, control de posición”

Manuel Alejandro Preciado Morán A01639643

Carlos Antonio Solano Vega A01540077

Jorge Carrillo Castro A01634630

Natalia Rodriguez Gonzalez A01639990

ÍNDICE

Esta primera actividad consistió en generar un control de velocidad para un robot Pioneer con respecto a su posición para alcanzar los puntos que le indicamos, haciendo una conexión con coppelia, la plataforma donde trabajamos con el robot, y matlab en donde están los códigos del control.

Controlador en matlab

Lo primero que hicimos para resolver fue crear el control de velocidad en matlab, para esto usamos las siguientes ecuaciones:

$$w = -kpr * tanh(\theta e)$$

$$v = vmax * tanh(\frac{kpt*d}{vmax})$$

Donde:

$w = Velocidad\ angular$

$kpr = Parte\ proporcional\ rotacional\ de\ nuestro\ controlador$

$tanh = tangente\ sigmoide$

$\theta e = Diferencia\ entre\ la\ orientación\ deseada\ y\ la\ orientación\ actual$

$v = Velocidad\ lineal$

$vmax = velocidad\ máxima$

$d = Distancia\ entre\ dos\ puntos\ (\sqrt{(x_{deseada} - x_{actual}) + (y_{deseada} - y_{actual})}$

$kpt = Parte\ proporcional\ translacional\ de\ nuestro\ controlador$

Ya obtenidas las ecuaciones pasamos a implementarlas en nuestro código de matlab, el cual fue desarrollado en clase por el profesor y Chat GPT, en este lo primero que hace es definir los parámetros del robot, como es el radio de las llantas y la distancia entre las mismas, el tiempo para la toma de muestras de la posición, la velocidad lineal máxima, que en este caso dejamos como 1, inicializamos la posición del robot en el origen con una orientación en 0, después decidimos cuáles eran las coordenadas en “x” y “y” que queríamos recorrer y las colocamos en dos vectores, definimos los valores para las partes proporcional rotacional y proporcional tangencial del controlador.

Ya teniendo nuestros datos hicimos un for que nos recorriera las tres posiciones y dentro de él colocamos un while en el que se queda menos de 5 segundos que nos realiza los cálculos cada 0.01 segundos, ahí calculamos cual es nuestra orientación

ideal con la tangente inversa de las coordenadas a las que queremos llegar, y la distancia que tenemos que recorrer con la raíz cuadrada de la diferencia entre la posición deseada con la actual, luego aplicamos las ecuaciones de control a cada motor y lo simulamos para nuestra gráfica donde podemos observar en tiempo real el movimiento del “robot”, ya que hasta este punto trabajamos con un sistema simulado.

Conexión con con coppelia

Antes de empezar con la siguiente parte hicimos un pequeño tutorial para la conexión entre matlab y coppelia. En esta pequeña práctica comenzamos con la configuración de todo nuestro ambiente, siguiendo algunas instrucciones sencillas como cambiar archivos de lugar y abrir Coppelia. Una vez en Coppelia, aprendimos a colocar diferentes formas, primero nos enfocamos en colocar un cuboide y modificar algunas de sus propiedades para poder hacerlos interactivo con el ambiente. Después colocamos el robot pioneer, el cual comenzó a avanzar en el momento que corrimos el script. Mediante el cubo hicimos la primera conexión de matlab, con un script muy sencillo el cual con la línea “simExtRemoteApiStart(19999)” logramos darle un puerto de comunicación a matlab.

Una vez que esto funcionó, comenzamos con el control del Pioneer. El primer paso fue lograr que el Pioneer moviera una sola llanta (la izquierda) y girara sobre esta misma. Después comenzamos con el uso del sensor ultrasónico que viene incluido con el Pioneer y veíamos la distancia desplegada en Matlab. Con el sensor funcionando y el Pioneer dando vueltas, ya podíamos ver la distancia que arrojaba el sensor en todo momento. Por último tuvimos que obtener la distancia que había entre ambos objetos, esto se hizo obteniendo la posición de ambos y haciendo una resta. Esta distancia también la desplegamos en matlab junto con los valores obtenidos con el sensor.

Integración de todo

Para el resultado final, en Coppelia decidimos remover el cubo y realizar la conexión con matlab directamente en el Pioneer 3dx. Para ello, en el script del Pioneer 3dx borramos las líneas de código establecidas previamente por el mismo programa para que sus movimientos dependieran de nuestros comandos establecidos en matlab y colocamos el robot en unas coordenadas específicas, así como unas esferas para especificar las coordenadas deseadas. Estas esferas les quitamos la propiedad de “dynamic and responsable” para que no haya ningún problema con el Pioneer.

Conclusión

En esta actividad se aprendieron dos tipos diferentes de controles, uno con una ecuación de velocidad donde se toma en cuenta la parte proporcional con la distancia y con el valor del error, mientras que en el segundo control usamos una velocidad máxima y una señal sigmoide la cual nos ayuda tener un rango de velocidad y poder cambiar la agresividad de la aceleración de los motores este a su vez utiliza una parte proporcional, la distancia y el valor de error. Para este caso se optó por utilizar el segundo control ya que se comportaba de una mejor manera, mientras que el primer control se movía en círculos el robot hasta llegar al objetivo, el segundo control nos ayudó a dirigir el robot de manera lineal, más rápida y precisa, así mismo se logró hacer una conexión entre el código de Matlab y Coppelia, lo cual fue muy interesante el poder ver como un código hace que en una simulación un robot se mueva y haga las instrucciones deseadas.

Link al video de resultados

1. https://youtu.be/XdWT-w_0Aa4