Proyecto 1: Generación de trayectorias en ROS
Robótica: SDI-11911

Manuel Abraham Mejía Medina
Instituto Tecnológico Autónomo de México

I. INTRODUCCIÓN

El objetivo del proyecto era que al aprender los fundamentos de ROS, así como entender la construccin de nodos, se pudiera crear un programa para controlar a la tortuga de Turltesim. Esto se lograría pidiendo a un usuario que introdujera coordenadas X y Y, así como el ángulo para determinar la direccón hacía donde se movería, incluido el tiempo en segundos en que le gustaría que lo completara.
Para hacer este programa se necesitó tener conocimientos previos básicos de programación, así como varias ejecuciones del programa usando el método de pureba y error, hasta que este funcionara de la manera mínima que se esperaba. Finalmente, se corrigieron los errores encontrados y se trató de optimizar, con los conocimientos hasta ese momento, la forma en la que está hecho el progama.

II. MARCO TEÓRICO

A. Qué es ROS?

Antes de haber empezado a usarlo, se necesitó responder esta pregunta y lo que se encontró fue que ROS es una abreviación para Robot Operating System. Es un framework creado para poder desarrollar software para robots proveyendo los servicios que da un sistema operativo, tales como controladores de dispositivo, abstracción de hardware, librerías, adminitración de paquetes, entre otras cosas.

ROS se creó con la idea de facilitar el diseo y construcción de software de robots multiusos, ya que normalmente, es muy difícil coordinar de manera propia lo que el robot realmente hace, con lo que se quiere que haga. Además, ayuda a que diferentes personas puedan tabajar en el mismo proyecto aunque no estén en el mimso lugar, para poder resolver así, problemas que como humanos parecen triviales, pero para el robot son muy comlicados.

B. Conceptos de ROS aplicados en la práctica

Los conceptos principalmente utilizados fueron turtlesim, roscpp, rospack, workspace y node. El primero es un simulador que ayuda a ensear los conceptos básicos de ROS a usuarios que apenas están aprendiendo. En general, todos los conceptos mencionados están relacionados entre sí, ya que turtlesim usa nodos, los cuales son ejecutables que se utilizan para comunicarse con otros nodos. Además, estos últimos pueden ser utilizados dentro de paquetes (packages), los cuales se instancian con el comando rospack, dentro de un workspace, que como su traducción al espaol lo indica, es un espacio de trabajo donde puedes llamar varios packages.

Por último, el comando de roscpp fue utilizado para poder programar en lenguaje C++ dentro ROS, ya este crea un ambiente de trabajo en donde se puede escribir un programa y luego ejecutarlo desde la terminal de Ubuntu para ejecutarlo y ver que funcione.

Fig. 1. Logo de ROS

C. Trayectorias de robots

Una de las finalidades más básicas en un robot es buscar su movimiento con el fin de alcanzar alguna posición definida. Para esto, es necesario conocer el modelo cinemático y dinámico del robot al que se quiera usar. En el caso de este proyecto, no fue necesario ya que es un programa, pero sí fue importante controlar el movimiento, ya que el simulador tiene limitaciones en cuanto al tamanio de la pantalla.
Los robots al realizar algún trabajo o tarea, deben seguir una trayectoria, para la cual se debe suponer un punto de inico y un punto final o destino. Para ir de ese punto de inicio al destino, se calculan una enorme cantidad de rutas posibles, en las que no siempre se elige la más rápida sino las más eficiente tomando en cuenta el terreno que se deba recorrer.
Existen diversos métodos para trazar la trayectoria pero la que se usó fue la de espacio cartesiano, en el que se da alguna coordenada (x,y), que es a donde deberá llegar la tortuga, y un ángulo, que es en la dirección en la que deberá moverse.

III. EXPERIMENTOS

La experimentación se dividió en varias etapas, en las que se fue solucionando poco a poco los problemas que

presentaba el programa, así como la solución final del movimiento de la tortuga.
Al inicio se buscó que la tortuga solamente se moviera en las coordenadas dadas, esta parte salió con ayuda del libro "A Gentle Introduction to ROS" y con lo visto en clase sobre los planos cartesianos. Inmediatamente se agregó que el usuario ingresara la dirección en la que la tortuga tenía que girar antes de dirigirse a una dirección, al probarlo solo surgieron errores de compilación pero por no haber colocado bien alguna variable
Después se trató de colocar el tiempo en el que el usuario quisiera que la tortuga completara el proceso, pero al probarlo surgieron diversos errores como que avanzaba demasiado lento al estar llegando al punto dado o que lanzaba error al ingresarle valores altos en la variable del ángulo, así como en la de coordenadas. Al final, se logró solucionar consultandolo con otros compaeros de la clase que también tenían ese problema.
En general el programa funcionó sin problemas con las pruebas finales que se le hicieron, pero al ser completado en poco tiempo es posible que pueda haber algún error no detectado al momento de las pruebas.

IV. CONCLUSIONES

Este proyecto realmente requirió de aprender nuevas habilidades al momento de realizarlo, ya que se hizo en un software del que se tenía conocimiento nulo al usarlo, con ayuda del libro, de la clase, compaeros y algunas búsquedas en internet fue posible completarlo de una manera satisfactoria. Algunas veces parecía que no iba a funcionar o a ocurrirse una código que solucionara el problema, pero aun así fue una experiencia importante y educativa, ya que además demostró que es necesario estar investigando y actualizandose para poder hacer eficientes funciones en un robot que para un humano son tan sencillas que no siquiera se piensan la mayoría de las veces.