**Visualización remota del mapa y la posición en el mapa del robot Pioneer 3-DX**

Jhon Kevin Muñoz Peña

Escuela de ingeniería Eléctrica y Electrónica

Universidad del Valle

Abstract—En el presente documento se explica brevemente el proyecto final presentado para el curso de interfaces, comenzando con una breve descripción de las partes que componen el proyecto y terminando con una guía corta de cómo hacer uso de la aplicación.

I. DESCRIPCION DEL PROYECTO

En este proyecto se busca transmitir de manera remota el mapa del entorno de trabajo del Pioneer 3-DX, y la posición del robot en dicho mapa para su visualización en una interfaz de usuario alojada en un computador diferente al computador embebido que posee el robot. El acceso a esta interfaz será controlado a través de la información de una base de datos, de tal manera que solo podrán ingresar usuarios registrados en esta base de datos y en el horario que tengan asignados.

A continuación se describirá brevemente lo referente a la administración de los usuarios y después se abordara lo referente a la aplicación central.

*A. Administración de usuarios.*

Para controlar el acceso de los usuarios al mapa de robot y su posición se utiliza una base de datos llamada robotdb. En esta base de datos se encuentran 4 diferentes tablas con la siguiente información:

**Tipo de usuario:** Se tienen dos tipos de usuario posibles, administrador y usuario normal.

**Usuarios:** Se consigan el nombre, apellido, correo y clave de cada usuario registrado.

**Proceso:** Se creó una tabla para manejar el proceso a administrar, en este momento solo se cuenta con el proceso del mapa del P3DX.

**Horarios:** Se registran los horarios validos en que cada usuario puede ingresar para visualizar el mapa del P3DX.

El acceso a esta tablas, por parte del administrador de la aplicación, se hace a partir de un módulo de administración en PHP al cual solo puede acceder un usuario tipo administrador, a través de este módulo se pueden crear y editar nuevos usuarios normales, así como también se pueden crear y modificar horarios para los usuarios registrados.

B. Aplicación

El análisis de la aplicación se dividirá en dos partes principales, los programas y archivos usados en el robot para la aplicación, y la interfaz en el computador del usuario.

1. Aplicación en el Pioneer P3-DX

La aplicación en el robot hace uso de la arquitectura de trabajo de ROS (Robot Operating System) para obtener la posición del robot basada en su odometría. ROS usa el concepto de nodos, tópicos, servicios y mensajes. Un nodo no es más que un archivo ejecutable dentro de la arquitectura de ROS. La información que usan los nodos para comunicarse se conoce como mensajes, estos mensajes viajan a través de buses de datos llamados tópicos.

En el desarrollo de este proyecto se hizo uso de dos nodos dentro de la arquitectura de ROS, fueron rosaria del paquete Rosaria y odometry\_Interface\_socket.py del paquete Beginner\_Tutorials, para cumplir con el objetivo del envió remoto de la posición del robot, además de un tercer nodo, interface del paquete rosaria\_client, para probar la aplicación, . Los 3 nodos mencionados anteriormente, así como otros involucrados, se explican brevemente a continuación.

**Roscore:** Es una colección de nodos y programas que son pre-requisitos para todos sistema basado en ROS. Es indispensable tener a roscore ejecutándose para que exista comunicación entre los diferentes nodos de la aplicación.

**Rosaria/Rosaria:** ROSARIA es un paquete que provee una interfaz a ROS para usar las librerías de Adept MobileRobots en el robot Pioneer 3DX. En este proyecto se hace uso de este paquete para acceder a la odometría del robot. Para hacer uso de RosAria se debe especificar el puerto al que está conectado el robot al PC embebido.

**Beginner\_Tutorials/odometry\_Interface\_socket.py**: Es un ejecutable creado específicamente para este proyecto, este nodo se suscribe al tópico de odometría de RosAria y envía la posición dada por la odometría a través de un socket por el puerto 9090.

**rosaria\_client/interface**: El paquete rosaria\_client cuenta con un conjunto de ejemplos de cómo usar la interfaz RosAria, de este paquete se usara el nodo interface para mover el robot con el teclado y probar la capacidad del proyecto de mostrar la posición de manera remota.

Además de los ejecutables de la arquitectura de ROS usados, para el desarrollo del proyecto en el robot también se implementó un archivo en java, Image\_Server\_Interfaces.java, para poder enviar de manera remota la imagen del mapa del robot. El mapa escogido para la prueba fue el mapa del espacio situado enfrente de las salas de sistema y el laboratorio de robótica móvil, este mapa se obtuvo haciendo uso de interfaz de programación ARIA propia del robot y se exporto a una imagen PNG. Image\_Server\_Interfaces toma esta imagen y la convierte a un arreglo de bytes, tras lo cual la envía por sockets en el puerto 9999 a la interfaz de usuario.

1. Interfaz de Usuario.

A continuación se analizarán las partes principales que componen la interfaz de usuario:

**Acceso de Usuarios:** El usuario ingresa su correo y contraseña, la aplicación se conecta a la base de datos y evalúa si los datos ingresados son correctos en relación con los datos de usuario registrados en la base de datos se evalúa si la hora de ingreso está dentro del horario programado para el respectivo usuario. Si todo está en orden se permite el ingreso del usuario, caso contrario se niega el acceso informando la razón al usuario.

**Conexión con el mapa del robot:** En el momento en que el usuario ingresa de manera correcta a la aplicación, esta se conecta por sockets en el puerto 9999 para obtener el mapa del robot, este mapa se recibe en forma de bytes y se convierte a imagen haciendo uso de OpenCV. Esto se hace una sola vez cada vez que el usuario ingresa los datos correctamente y presiona el botón de acceso, cerrándose el canal de comunicación una vez se ha obtenido el mapa.

**Conexión con la posición del robot:** Una vez se ha obtenido el mapa del robot, el usuario puede solicitar obtener la posición del robot en el mapa, cuando esto ocurre la aplicación ejecuta un hilo, dentro del cual se ejecuta la comunicación por sockets en el puerto 9090 para obtener la odometría del robot y actualizar el punto que representa al robot en el mapa en otro hilo que se ejecuta en paralelo.

II. GUIA DE LA APLICACIÓN.

Antes de ejecutar la aplicación se debe situar al robot en la posición inicial escogida para trabajar como referencia en la interfaz, es con respecto a esta posición que RosAria publicara su información de odometría. En este caso la posición inicial configurada es la puerta de enfrente del laboratorio de robótica. Una vez hecho esto se procede a poner en marcha la aplicación.

En el robot:

1.1 Se ingresa a /home/p3dx/Jhon\_Kevin y se ejecuta Image\_Server\_Interfaces.java:

$ java Image\_Server\_Interfaces 9999

1.2. Se ejecutan los nodos de ROS

En una terminal:

$ roscore

En una nueva terminal:

$ rosrun rosaria Rosaria \_port:=/dev/ttyS0

En la interfaz de usuario:

2.1 El usuario ingresa su correo y contraseña

2.2 Si los datos ingresados y el horario de ingreso son válidos la aplicación se conecta por sockets al mapa del robot, en el caso de que el robot no este transmitiendo ningún mapa se informara al usuario de esto y se deberá volver a oprimir login cuando se sepa que el mapa está transmitiéndose. Si existe un mapa, el usuario deberá esperar unos segundos mientras se crea la comunicación entre las dos computadoras, después de esto aparecerá el mapa en la interfaz y se habilitara el botón de acceso a la posición.

2.3 El usuario Oprime el botón de acceso a la posición y se queda esperando por la información de odometría.

En el robot:

3. se ejecuta el nodo de odometría.

En una nueva terminal:

$ rosrun beginner\_tutorials odometry\_Interface\_socket.py

Una vez realizados los pasos anteriores el usuario podrá ver la posición dada por la odometría del robot en tiempo real mientras el robot se mueve en el mapa de prueba escogido. El movimiento del robot puede llevarse a cabo antes o después de ejecutarse la aplicación en el lado del usuario y su posición, según su odometría, se podrá observar en la interfaz de usuario.

Para efectos de prueba se ejecuta el nodo de rosaria\_client/interface:

$ rosrun rosaria\_client interface

Y se ingresa la opción 4 para teleoperacion.

El movimiento del robot puede hacerse de manera teleoperada o autónoma, dentro o fuera del sistema de ROS, y su posición se mostrara en la interfaz de usuario siempre que se hallan llevado a cabo los pasos 1~3.

REFERENCIAS

- ROS. . [Consultado 05/03/2016] Disponible en: http://wiki.ros.org/

- How to use ROSARIA. [Consultado 05/03/2016] Disponible en: http://wiki.ros.org/ROSARIA/Tutorials/How%20to%20use%20ROSARIA