

Integración del funcionamiento del UR3e con el Turtlebot4 utilizando Matlab

Illán Linos Souto, Malena Potesta González, Sara Rodríguez Fernández
Máster en Ingeniería Industrial, especialidad en Sistemas Robóticos

Curso 2023-2024

Este trabajo busca la integración de los robots UR3e y Turtlebot4. Se propone la creación de un programa en Matlab, en el que se emplean técnicas de Machine Learning para distinguir entre piezas correctas y defectuosas, además, de poder distinguir la forma de las piezas. Se busca que el UR3e le entregue la pieza al Turtlebot4 y que, en función de si la pieza presenta defectos o no, el Turtlebot4 lleve la pieza a un punto u otro. En este trabajo se utiliza como software principal Matlab, con las toolbox de RoboDK, Deep Learning y ROS. Otros software utilizados son Motive, RoboDK, ROS2 y RViz2. Finalmente, se consigue que todo el sistema funcione de forma coordinada en tiempo real.

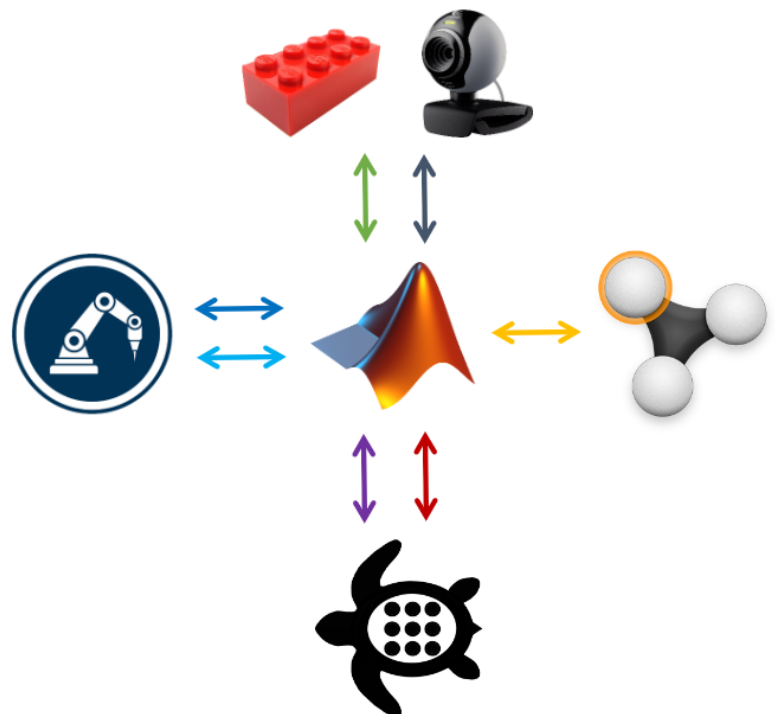
OBJETIVOS

El objetivo principal de este trabajo es la integración de los diferentes sensores, robots y métodos de aprendizaje con los que se han trabajado a lo largo del curso.

Como objetivos secundarios se tienen la búsqueda de un mayor conocimiento del programa RoboDK y de su funcionamiento conjunto con el UR3e y Matlab. Del mismo modo, se pretende profundizar en las técnicas de Machine Learning mediante redes neuronales creadas en Matlab y la utilización de técnicas SLAM para generar un mapa del laboratorio con Turtlebot4 y poder navegar a distintos puntos del laboratorio enviándole las posiciones deseadas desde Matlab.

ARQUITECTURA

- 1 Obtener trayectorias del Motive
- 2 Generar programa en RoboDK
- 3 Tomar foto del lego
- 4 Enviar el turtlebot al punto de recogida
- 5 Se clasifica la pieza
- 6 Se ejecuta el programa de RoboDK
- 7 El turtlebot recoge la pieza y se mueve al destino según la clasificación



PROGRAMACIÓN EN MATLAB

Previo a la programación en Matlab se realiza una captura del movimiento del UR3e (el cual se mueve con la touch pendant), mediante el software Motive en el que se prepara la trayectoria que debe realizar el UR3e para pasarle la pieza al Turtlebot4. Las cámaras deben ser calibradas, primeramente. A continuación, se colocan sobre el extremo del brazo robot tres marcadores que se identifican con un sólido rígido, para obtener tanto posiciones como orientaciones. A su vez, se crea la base de datos que utiliza la red neuronal y se realiza su entrenamiento en Matlab, con unos resultados de precisión en torno al 95%. Debe realizarse, también, previamente a la programación del script conjunto, el mapeo del laboratorio con el Turtlebot4 para lo que se emplean técnicas SLAM del paquete turtlebot4navigation y RViz2.

Una vez realizados los trabajos previos se comienza con la creación del script de Matlab. En primer lugar se establece la conexión entre Matlab y RoboDK y generan en Matlab los comandos que permiten la creación de un programa en RoboDK. Para que este programa se ejecute en el robot real se debe establecer la conexión en control remoto (el ordenador y robot deben estar conectados a la misma red) y desde RoboDK se debe marcar la opción *Ejecutar en el robot*. La siguiente parte consiste en la captura de una imagen de la pieza a clasificar y guardarla en la misma carpeta donde está el fichero. A continuación, se crea un nodo desde Matlab que permite comunicarse con el Turtlebot4 usando Ros2. Mediante el topic `goal_pose` se le indica al Turtlebot la posición a la que debe acudir para recoger la pieza y las posiciones a las que debe ir en caso de que la pieza sea válida o defectuosa (para que el robot pueda navegar se deben abrir en otro ordenador el mapa del laboratorio generado). Con el topic `cmd_vel` se ajusta la posición del robot en el lugar de recogida de la pieza. Hecho esto, se realiza la clasificación de la pieza con la red entrenada, se ejecuta el programa creado para que el UR3e le pase la pieza al Turtlebot4 y, una vez que el Turtlebot tiene la pieza, se procede a alejar el robot y se envía a un punto u otro del laboratorio en función de si la pieza presenta defectos o no. Para que la conexión Matlab-Turtlebot4 funcione, el ordenador en el que se encuentra el script de Matlab y el Turtlebot4 deben estar conectados a la misma red.

RESULTADOS

Una vez creado el programa se comprueba que se ejecuta correctamente y que ambos robots actúan de forma coordinada en tiempo real comunicándose mediante flags (Figura 1). Surgen algunos problemas a la hora de realizar la programación del sistema. En primer lugar se debe tener en cuenta que Motive no es especialmente preciso y esto hace que la trayectoria grabada no coincida perfectamente con la real. Además, se deben adaptar las orientaciones de Motive al sistema de ejes de RoboDK (Figura 2). En cuanto a la red neuronal, la mayor dificultad está en tener en el laboratorio una iluminación correcta para la captura de imágenes, a la vez que obtener el número adecuado de imágenes y evitar sobre entrenamientos. El Turtlebot4 presenta numerosos problemas en su funcionamiento, muchos ellos dados por una batería que se descarga a gran velocidad (Figura 3) y un software que debe reiniciarse frecuentemente. La simulación del robot presenta un bufer ficticio, lo que unido a que el lidar del robot tampoco posee gran precisión, dificulta que se acerque de forma exacta el UR3e, al detectar los obstáculos más cerca de lo que están. También, surgen algunos problemas en relación a las versiones del software que se utilicen. En la versión de 2022 no se le pueden enviar acciones al Turtlebot4, solo topic, pero sí permite enviarle al UR3e las órdenes de conectarse con el robot real, lo que no permite la versión 2023. A su vez, la conexión de RoboDK con softwares de programación está pensada especialmente para Python, donde sí se puede dar la orden de ejecutar el programa en el robot real directamente (sin marcar la opción desde RoboDK), lo que no lo permite Matlab.



Figura 1.— UR3e y Turtlebot4

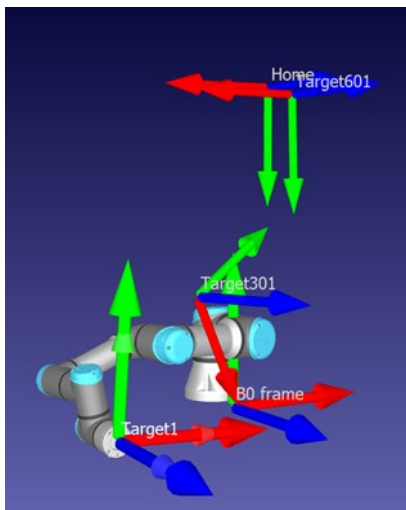


Figura 2.— Programa en RoboDK



Figura 3.— Batería e IP del Turtlebot4

CONCLUSIONES

A pesar de las dificultades encontradas al trabajar con dos robots distintos y diferentes software y lenguajes de programación, se ha logrado crear un programa en Matlab que cumple con los objetivos establecidos y consigue que ambos robots funcionen coordinadamente. Como posibles ampliaciones del trabajo se propone la integración de la pinza del UR3e para la manipulación de las piezas, así como la utilización de un robot distinto al Turtlebot4, que pueda funcionar con mayor precisión.