Integración de cobots en ROS legado: Diseño de un sistema multi-robot interactivo

Miguel Burgh-Oliván, Rosario Aragüés, Gonzalo López-Nicolás

Afiliación: Robótica, Percepción y Tiempo Real (RoPeRT)
Instituto de Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.
Tel. +34-976762707, e-mail: *mburgh@unizar.es*

Resumen

Este estudio explora la implementación y uso de ROS Kinetic en sistemas multi-robot con cobots, centrándose en la integración y control interactivo. Aborda el diseño, configuración y ejecución, proporcionando ejemplos y documentación de modelos propuestos, sentando bases para futuros trabajos.

Sistemas multi-robot en ROS

Se emplea ROS (Robot Operating System) [1] para facilitar el desarrollo de software de robots. Este sistema de código abierto permite impulsar la innovación en robótica, gracias a que ofrece integración de hardware, simulación y visualización, y cuenta con una comunidad activa. ROS es ampliamente utilizado en la industria y la investigación. Nuestro desafío es configurar e integrar un sistema multi-robot con cobots en ROS Kinetic, una versión legada. Este sigue siendo ampliamente utilizado debido a diversas razones, como la dependencia de software o hardware específico, la dificultad de realizar una migración del sistema, y el coste y los riesgos que puede implicar dicha migración. En concreto, para nuestro caso, nos encontramos con la falta de documentación para la implementación e integración de dicho sistema utilizando cobots. Esto se debe a la complejidad de la configuración y al hecho de que el software utilizado (MoveIt!) para la planificación de movimientos no ofrece soporte para más de un cobot simultáneamente cuando se deben concretar las posiciones objetivo durante la ejecución. Por ello, se plantea solucionarlo aprovechando la ventaja de la arquitectura distribuida de ROS, que facilita la integración y modificación de los nodos del sistema al ser de código abierto. Esto nos permite adaptar y modificar la arquitectura del sistema para las soluciones propuestas.

Soluciones para la integración de múltiples cobots

Las soluciones propuestas (Ver Tabla 1), permiten la integración de N cobots en el sistema. Estos cobots pueden ser de diferentes modelos y fabricantes. También permite el uso de diferentes tipos de controladores, así como la interacción con el usuario. En el repositorio público de Github [2] bajo la licencia CC BY 4.0, se explica en detalle cada una de estas soluciones, así como en el artículo [3].

Tabla 1. Soluciones propuestas en donde los componentes clave son el planificador de movimientos y el URDF que representa el modelado de los robots. La primera solución es una replicación del planificador N veces, modelando un único cobot. La segunda opción replica un planificador de propósito específico que reduce el consumo de recursos y modelando M cobots.

Solución propuesta	Planificador	Réplicas	Cobots (URDF)
MoveIt-Nx1	MoveIt!	N	1
Custom-NxM	Custom	N	M

Resultados experimentales

Se han realizado varias pruebas en el simulador de Gazebo con uno, dos y cuatro cobots para realizar la tarea de agarrar y dejar un objeto. Se han evaluado ambas soluciones propuestas (con y sin el planificador MoveIt!). También se han realizado las pruebas en Gazebo para controlar dos cobots mediante Leap Motion [4]. Finalmente, se ha realizado la prueba en el robot físico (Ver Figura 1), pudiendo verificar el funcionamiento correcto del sistema así como la interacción correcta con el robot en un entorno real. Cabe añadir que la solución Custom-NxM proporciona una interacción más fluida que la solución MoveIt-Nx1, porque no tiene que esperar a que el planificador termine la ejecución de la trayectoria actual para ejecutar la siguiente.

Figura 1. Resultados realizando un *pick and place* con el robot físico, interactuando con el cobot mediante Leap Motion, que controla la pinza y la posición del robot.

Puesta en marcha del sistema

La documentación para la puesta en marcha también se encuentra en la página web [2]. Se explica paso a paso la configuración y los diseños implementados para uno, dos y cuatro cobots utilizando o sin utilizar el planificador de movimientos *MoveIt!*. Además también se presenta un ejemplo [2] para el control de uno o dos cobots mediante el dispositivo *Leap Motion*. Finalmente, se expone un ejemplo de cómo integrarlo y su funcionamiento en un robot físico. Se instala ROS Kinetic Kame en el sistema operativo Ubuntu 16.04. Es necesario tener Python 2.7 para garantizar el funcionamiento de los nodos implementados y opcionalmente el dispositivo Leap Motion. Se muestra un ejemplo en las Figuras 2 y 3.

- Clonar este repositorio:

```
git clone https://github.com/Serru/MultiCobot-UR10-Gripper
```

- Establecer el workspace de catkin:

```
cd ~/MultiCobot-UR10-Gripper/src
catkin_init_workspace
```

- Instalar todas las dependencias:

```
source /opt/ros/kinetic/setup.bash
cd ~/MultiCobot-UR10-Gripper
rosdep update
rosdep install -r --rosdistro kinetic --ignore-src --from-paths src
```

- Compilar el repositorio

```
catkin_make
rospack profile
source devel/setup.bash
```

Figura 2. Proceso de instalación del sistema y de las soluciones propuestas, *Custom-NxM y Moveit-Nx1* (Github).

Agradecimientos: Trabajo financiado por la Unión Europea - NextGeneration EU, por proyectos PID2021-124137OB-I00 y TED2021-130224B-I00 financiados por MCIN/AEI/10.13039/501100011033/ y por FEDER Una manera de hacer Europa.

Terminal 1:

```
roslaunch two_arm_no_moveit_gazebo ur10_joint_limited.launch
```

Terminal 2:

```
rosrun two_arm_no_moveit_manipulator ur10_1_robot_manipulator.py
```

Terminal 3:

```
rosrun two_arm_no_moveit_manipulator ur10_2_robot_manipulator.py
```

Figura 3. Preparación del entorno ROS y ejecución para el control de dos cobots utilizando la solución propuesta *Custom-NxM* (Github).

Conclusiones

En este trabajo se ha abordado la implementación y el diseño de un sistema multi-robot para cobots en ROS Kinetic que permite la manipulación de objetos mediante el dispositivo Leap Motion, el cual habilita la interacción entre el cobot y el usuario (posibilita la interacción simultáneamente con dos cobots) a través del reconocimiento de gestos y movimiento de las manos. De las diferentes alternativas se han seleccionado dos (MoveIt-Nx1 y Custom-NxM). Ambos son escalables, permiten movimientos simultáneos, integración heterogénea de cobots e interacción con el usuario, la diferencia principal es que Custom-NxM no tiene las limitaciones de MoveIt!, siendo más ligera y rápida para la realización de un trabajo concreto. Por el contrario, MoveIt! Es más lenta pero es más flexible porque provee más funcionalidades. Por esta razón se mantienen ambos modelos como solución. Finalmente ambos modelos se han validado tanto en simulación como en un entorno real.

REFERENCIAS

[1]. QUIGLEY, M., et al.: ROS: an open-source robot operating system. In: ICRA workshop on Open Source Software, vol. 3, p. 5. Kobe, Japan (2009)

[2]. BURGH-OLIVÁN, M., ARAGÜÉS, R. and LÓPEZ-NICOLÁS, G. MultiCobot-UR10-Gripper. [Computer software]. Retrieved from: https://github.com/Serru/MultiCobot-UR10-Gripper

[3]. BURGH-OLIVÁN, M., ARAGÜÉS, R. and LÓPEZ-NICOLÁS, G. ROS-Based Multirobot System for Collaborative Interaction. *Fifth Iberian Robotics Conference*. *ROBOT2022*. Lecture Notes in Networks and Systems, vol 589. Springer, Cham. https://doi.org/10.1007/978-3-031-21065-5_34.

[4]. LU, W., TONG, Z., & CHU, J. Dynamic hand gesture recognition with leap motion controller. IEEE Signal Processing Letters, 23(9), 2016, 1188-1192.