

# ESQUEMA DE INTEGRACIÓN PARA EXPERIMENTACIÓN REAL CON SISTEMAS ROBÓTICOS BI-BRAZO

Abiud Rojas-de-Silva, Carlos Rodríguez, Raúl Suárez. \*

## Resumen

*En este artículo se presenta un esquema para la integración de un sistema robótico bi-brazo, para transformar resultados obtenidos de simulaciones en ejecuciones en un entorno real. Este esquema utiliza una capa de comunicación basada en “The Robot Operating System (ROS)” y el software “The Kautham Project”, para calcular las trayectorias de movimiento, validar la existencia de soluciones de cinemáticas inversas, chequeo de colisiones y visualizar gráficamente la simulación del entorno. El sistema robótico bi-brazo utilizado es ADARS (Anthropomorphic Dual Arm Robotic System). Se han utilizado dos tipos de aplicaciones de cooperación bi-brazo para mostrar la validez del esquema durante la experimentación real.*

**Palabras clave:** Cooperación, sistemas bi-brazo, sistemas bi-manuales, ROS, Kautham.

## 1. INTRODUCCIÓN

La manipulación de objetos utilizando sistemas robóticos es una actividad que requiere resolver diferentes tipos de problemas, tales como calcular la prensión del objeto que será manipulado, planificar la trayectoria del robot para alcanzar la configuración de la prensión, calcular un camino para mover el objeto desde su configuración inicial a una nueva configuración, entre otros. En el caso particular de los sistemas bi-brazo, la complejidad de estos problemas aumenta debido a que es necesaria una distribución de tareas, así como la coordinación de cada conjunto mano-brazo (en adelante Robot) cuando éstos trabajan como sistemas independientes o de forma cooperativa para realizar la misma tarea.

\*Los autores pertenecen al Institut d’Organització i Control de Sistemes Industrials (IOC) - Universitat Politècnica de Catalunya (UPC), Barcelona, España ([francisco.abiud.rojas.de.silva](mailto:francisco.abiud.rojas.de.silva), [carlos.rodriguez.p](mailto:carlos.rodriguez.p), [raul.suarez@upc.edu](mailto:raul.suarez@upc.edu)). Este trabajo ha sido parcialmente financiado por el Gobierno de España a través de los proyectos DPI2013-40882-P y DPI2014-57757-R. A. Rojas fue parcialmente apoyado por la beca doctoral mexicana CONACyT 313768.

Estas cuestiones han motivado el desarrollo de una gran variedad de algoritmos para sistemas multi-robot para la coordinación del sistema y la distribución de tareas. No obstante, para traducir los resultados de estos algoritmos en movimientos reales de cada uno de los componentes del sistema robótico, es necesario contar con un esquema de integración que permita movimientos coordinados, y así poder evaluar el rendimiento de dichos enfoques en entornos reales.

En este artículo se presenta un esquema de integración para sincronizar y coordinar sistemas robóticos bi-brazo reales con la finalidad de utilizarlos como cadenas cinemáticas cerradas (e.g. prensión de objetos voluminosos) o como cadenas cinemáticas dobles (e.g. mover y manipular objetos en el mismo espacio de trabajo). El esquema presentado en este trabajo utiliza una capa de comunicación basada en *The Robot Operating System (ROS)* [1] para comandar y sincronizar cada componente del sistema bi-brazo. Se han utilizado dos aplicaciones robóticas para ilustrar la validez del enfoque propuesto. En la primera aplicación los Robots trabajan de forma desacoplada pero dentro del mismo entorno de trabajo con el objetivo de sujetar determinados objetos, quitando obstáculos potenciales y distribuyendo adecuadamente las tareas entre los Robots y fijando su secuencia. La segunda aplicación aborda el problema de sujetar objetos voluminosos usando dos manos antropomorfas de forma simultánea.

Después de esta introducción, el artículo está organizado de la siguiente forma. La Sección 2 presenta una revisión de trabajos previos relacionados, la Sección 3 presenta el hardware y el software utilizado y en la Sección 4 se presenta el esquema de integración. En la Sección 5 se detalla la capa de comunicación basada en ROS. Posteriormente, la Sección 6 explica las aplicaciones que se utilizan para mostrar la validez del esquema y la Sección 7 presenta dos experimentos reales utilizando el esquema propuesto. Finalmente, la Sección 8 resume el trabajo y presenta algunos temas que pueden ser tratados en el futuro.

## 2. TRABAJOS RELACIONADOS

El uso de sistemas robóticos bi-brazo para realizar tareas de manipulación ha generado una gran variedad de propuestas, principalmente enfocadas en tareas que involucren la manipulación de un solo objeto usando ambos brazos [2] [3], o manipulación de múltiples objetos usando cada brazo de forma independiente [4] [5].

La combinación de la planificación de tareas y movimientos cuando se manipulan objetos con dos brazos de forma independiente es aún un problema que no ha sido resuelto totalmente [6]. A pesar de ello, ya se han desarrollado diferentes algoritmos para prensión y manipulación de objetos en entornos con obstáculos [7] [8] y [9]. Una de las aplicaciones presentadas en este trabajo aborda este problema.

Por otro lado, el problema de encontrar prensiones alcanzables para objetos voluminosos utilizando dos manos simultáneamente se ha convertido también en una rama importante de investigación [2]. Este problema se ha abordado para objetos 2D [10] y 3D utilizando funciones de coste para encontrar prensiones adecuadas [11] y planificadores de movimientos con características especiales [12] [13] [3].

A pesar de que la mayoría de los algoritmos de planificación han sido aplicados en entornos reales para mostrar su robustez, usualmente no se detalla el esquema de comunicación utilizado para su ejecución en sistemas físicos reales. Ejemplos de trabajos que abordan este tema están relacionados con la teleoperación de sistemas robóticos [14] [15] o con el desarrollo de plataformas genéricas para la integración de componentes robóticos a bajo nivel [16] [17].

## 3. MONTAJE EXPERIMENTAL

El hardware utilizado en este trabajo es el “Anthropomorphic Dual Arm Robotic System” (ADARS) mostrado en la Figura 1, cuyos componentes son:

- **Brazos robóticos.** Dos manipuladores UR5 de Universal Robots de 6 grados de libertad, con una capacidad de carga de 5 kg.
- **Manos antropomorfas.** Dos manos antropomorfas Allegro de Simlab, de cuatro dedos y 16 grados de libertad (4 por dedo) y una capacidad de carga de 1.5kg.
- **Sensores táctiles.** Cada uno de los dedos de las manos está equipado con un sensor táctil que tiene una matriz de 4 x 8 celdas sensibles con una resolución de 3.8 mm y una frecuencia de muestreo de 400 medidas por segundo.



Figura 1: ADARS: Anthropomorphic Dual-Arm Robotic System.

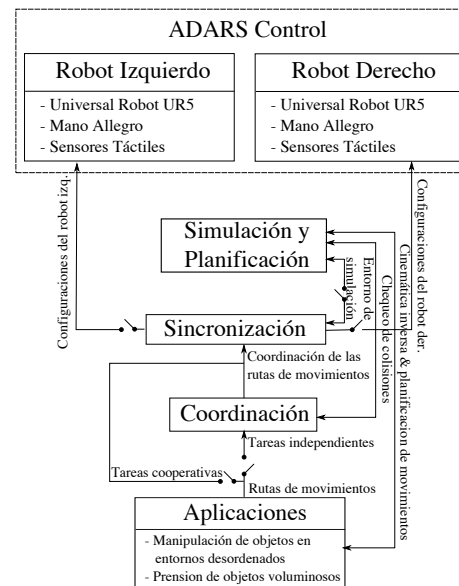


Figura 2: Estructura general del esquema de integración.

## 4. ESQUEMA DE INTEGRACIÓN

El objetivo principal del trabajo es facilitar la realización de experimentación real utilizando un sistema bi-brazo, incluyendo tareas independientes y cooperativas y, además, se considera la posibilidad de realizar coordinación de movimientos en línea, para evitar posibles colisiones entre los Robots.

La estructura general del esquema de integración se muestra en la Figura 2, siendo sus módulos principales:

- **Aplicaciones.** En este módulo se encuentran los algoritmos de planificación. Los datos de salida de estos algoritmos son un conjunto de configuraciones articulares que definen los caminos que recorrerá el sistema bi-brazo.
- **Coordinación.** Este módulo coordina las trayectorias de cada Robot con la finalidad de evitar posibles colisiones durante la ejecución de la tarea. Este módulo se utiliza cuando los brazos ejecutan tareas independientes, en caso contrario, se asume que las rutas de movimiento ya han sido coordinadas por el planificador.

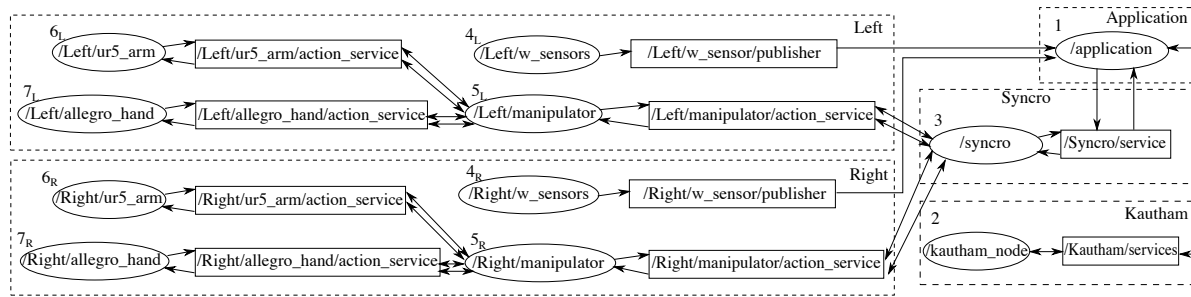


Figura 3: Diagrama de la capa de comunicación basada en ROS.

- **Sincronización.** Este módulo sincroniza los movimientos de los componentes del sistema bi-brazo; divide cada configuración global en las partes correspondientes a cada componente, i.e. la configuración correspondiente a cada brazo y mano.
- **Simulación y planificación.** *The Kautham Project* [18] es una herramienta de software de código abierto desarrollada en el Instituto de Organización y Control de Sistemas Industriales (IOC-UPC) utilizada para calcular cinemáticas inversas, realizar chequeo de colisiones, calcular las trayectorias de movimientos y visualizar gráficamente entornos de simulación.
- **ADARS Control.** Este módulo se comunica con las controladoras de cada elemento del sistema robótico bi-brazo. El módulo recibe los valores articulares, separados previamente en el módulo Sincronización y los envía a las controladoras correspondientes.

## 5. CAPA DE COMUNICACIÓN

### 5.1. COMUNICACIONES BASADAS EN ROS

El esquema de integración utiliza una capa basada en ROS que permite la comunicación entre diferentes procesos basándose en nodos de una red *peer-to-peer*. Las comunicaciones entre los nodos se puede clasificar en: a) Tópicos: comunicación basada en una estructura “publicador/subscriptor”. Un nodo puede publicar información enviando un mensaje a un tópico determinado, y cualquier otro nodo debe suscribirse al mismo tópico si necesita esta información; b) Servicios: comunicación basada en un sistema de “cliente/servidor”, en el cual un nodo cliente envía una solicitud de información a un nodo servidor y espera a que éste le responda; c) Acciones: comunicación basada en un sistema “cliente/servidor”, pero con la particularidad de que las acciones requieren una realimentación, que es positiva si la solicitud fue entregada correctamente

y negativa si no lo fue, en cuyo caso las acciones siguientes no se ejecutan.

La Figura 3 muestra el diagrama de la capa de comunicación desarrollada basada en ROS. Los óvalos representan los nodos (descritos más adelante), los rectángulos representan tópicos, servicios y acciones, y los rectángulos con líneas discontinuas corresponden a “namespaces”, que son contenedores que permiten usar los mismo nodos y tópicos para diferentes robots. En este trabajo se han definido dos “namespaces”, Left y Right, correspondiendo cada uno a un Robot.

### 5.2. NODOS DE LA CAPA DE COMUNICACIÓN

La capa de comunicación funciona de la siguiente forma. El nodo #1 es utilizado por las aplicaciones de cooperación (en la Sección 6 se presentan dos ejemplos). Cada aplicación genera los caminos que serán recorridos por el sistema bi-brazo y que son enviados a través del nodo cliente #1 al nodo #2 para visualizar gráficamente la tarea (simulando la sincronización) y al nodo #3 para realizar la sincronización en la ejecución real.

El nodo #2 está incluido dentro de *The Kautham Project* permitiendo utilizar muchas de sus herramientas a través de los siguientes servicios.

- **Collision\_check.** Detecta colisiones entre los Robots y el entorno, entre ambos Robots y autocolisiones de cada Robot.
- **Solve\_query.** Recibe la configuración inicial y final deseadas de un robot y genera un camino entre ellas. Este servicio puede utilizar una variedad de planificadores provenientes de la librería *Open Motion Planning Library* (OMPL).
- **Move\_robots.** Mueve el robot a una configuración dada.
- **Coordinate\_robots.** Recibe un camino para cada Robot y coordina sus movimientos para evitar colisiones entre ellos. Este servicio se utiliza cuando cada Robot realiza tareas independientes en un espacio de trabajo compartido.

- **Simulation.** Simula el espacio de trabajo real. Este servicio recibe el modelo del espacio de trabajo y las trayectorias del sistema bi-brazo.

El nodo #2 se utiliza en tres procesos diferentes. En el primero, las aplicaciones utilizan el nodo #1 para llamar funciones de *The Kautham Project* usando los servicios proporcionados por el nodo #2 para resolver cinemáticas inversas, encontrar caminos y realizar chequeo de colisiones. En el segundo proceso el nodo #2 se usa para realizar coordinación en línea, y en el último proceso el nodo #2 se usa para visualizar gráficamente la ejecución de la tarea en un entorno simulado.

El nodo #3 recibe del nodo #1 los caminos de ambos Robots en un único mensaje, posteriormente se encarga de dividir y asignar a cada Robot  $R_i$  la parte del mensaje que le corresponde, con  $i = L, R$  indicando *Left* y *Right* (en adelante se utilizará el mismo subíndice para indicar los nodos asignados a los componentes *Left* y *Right*). Después de esto, cada mensaje se envía al nodo #5 $_i$ , donde se divide en la configuración del brazo y la mano.

El nodo #4 $_i$  obtiene información sobre el estado de los sensores táctiles del Robot  $R_i$  a través de una solicitud periódica realizada a una librería interna que proporciona, en caso de que exista, el punto exacto de contacto, así como la fuerza aplicada. Posteriormente esta información se publica en un tópico para que pueda ser usada por otras aplicaciones.

El nodo #6 $_i$ , proporciona las acciones para controlar los brazos UR5. Las controladoras de cada brazo están conectadas a través de un protocolo Ethernet a una aplicación “cliente/servidor” a bajo nivel que permite enviar posiciones articulares, velocidades y aceleraciones, así como determinar la posición actual de cada articulación.

El nodo #7 $_i$  contiene las acciones para controlar las manos Allegro. Las manos están conectadas a través de una interfaz CAN (*Controller Area Network*), y se controlan con un controlador de posición que calcula el torque necesario para alcanzar las posiciones deseadas.

Las acciones de los nodos correspondientes a los Robots, están definidas con tres mensajes para intercambiar la información: solicitud, resultado y realimentación. Cuando se requiere un cambio en la configuración del Robot, los nodos #6 $_i$  y #7 $_i$  generan los mensajes de solicitud que contienen las nuevas configuraciones. Estos mensajes se envían a cada una de las controladoras de los Robots que internamente ejecutan las funciones que mueven cada componente; cuando los movimientos terminan, se genera un mensaje que informa si las nuevas configuraciones han sido alcanzadas o no. El

mensaje de realimentación informa si la solicitud de movimiento fue recibida correctamente, y de no ser así se rechaza cualquier otro mensaje de solicitud.

## 6. APLICACIONES

La manipulación coordinada utilizando sistemas bi-brazo puede ser clasificada en coordinación por metas y manipulación bimanual [2]. En la primera clase, los manipuladores resuelven la misma tarea pero sin interactuar directamente entre ellos (e.g. colocar diferentes objetos en una misma caja). En la segunda clase ambos brazos interactúan directamente (e.g. levantar un objeto usando los dos manipuladores). A continuación se presenta una aplicación de cada clase para mostrar como fueron implementadas con el esquema de integración propuesto. La primera aplicación consiste en manipular objetos en entornos con diversos obstáculos y la segunda consiste en realizar una prensión cooperativa de objetos voluminosos.

### 6.1. MANIPULACIÓN DE OBJETOS EN ENTORNOS CON OBSTÁCULOS

El objetivo de esta aplicación consiste en sujetar dos objetos específicos con el sistema bi-brazo. Los objetos pueden sujetarse con cualquiera de los dos Robots o bien pueden ser preasignados deliberadamente, dependiendo de la tarea a realizar. El acceso a los objetos de interés puede estar bloqueado por otros objetos que deberán ser quitados. Se asume que las posiciones de los objetos y las configuraciones iniciales y finales de los Robots son conocidas.

Una descripción detallada de esta aplicación puede consultarse en [4], donde se presenta una variación de un planificador probabilístico para calcular las trayectorias de los Robots usando un *grafo de precedencias*  $G$  para representar las tareas que serán ejecutadas por cada Robot. Los nodos de  $G$  representan los objetos de interés y los objetos amovibles  $O_j$  ( $j = 1, \dots, n$ ), y los arcos representan cada Robot  $R_i$  ( $i = 1, 2$ ).

Se utiliza un planificador de movimientos para calcular una ruta  $P_{i,j}$  desde la configuración inicial de  $R_i$ , hacia la configuración de prensión de  $O_j$ , pero, a diferencia de otros planificadores, no se descartan las configuraciones del Robot que colisionan con un objeto amovible  $O_k$ , y los objetos en colisión se incluyen en un conjunto de obstáculos amovibles  $SO_{i,j}$ . Al final, la ruta de movimiento del Robot  $P_{i,j}$  se genera incluyendo el conjunto de obstáculos amovibles  $SO_{i,j}$ . El planificador de movimientos se utiliza de forma recursiva para encon-

trar caminos que permitan quitar cada  $O_k \in SO_{i,j}$  siguiendo una estrategia de mínimo coste hasta encontrar un camino libre de colisiones que permita sujetar los objetos de interés. El chequeo de colisiones se realiza considerando el brazo y la mano del Robot cuando está recorriendo el camino hacia el objeto que será sujetado, y considerando también el objeto sujetado cuando éste es quitado del entorno. El cálculo de las trayectorias de movimiento y el chequeo de colisiones se realizan usando las funciones de *The Kautham Project* a través del nodo #2, que, después de recibir una solicitud del nodo #1, proporciona acceso a las funciones a través de los servicios **Collision\_check** y **Solve\_query**. Finalmente, las rutas deben ser ejecutadas por el sistema bi-brazo, pero debido a que éstas no se han coordinado y como las tareas asignadas a cada brazo se realizan de forma paralela, pueden existir colisiones entre los Robots. Para evitar dichas colisiones se ejecuta un proceso de coordinación en línea que prioriza las tareas haciendo que, ante riesgo de colisión, una tarea (es decir, un Robot) tenga prioridad sobre la otra [19]. Esta coordinación se realiza a través del servicio **Coordinate\_robot** también proporcionado por el nodo #2.

La figura 4(a) muestra un ejemplo hipotético en 2D, el entorno está compuesto por objetos cuadrados que representan los objetos amovibles  $O_j$  frente a un sistema robótico bi-brazo representado por  $R_1$  y  $R_2$ . Los cuadrados  $O_1$  y  $O_2$  representan los objetos de interés y los demás cuadrados representan los obstáculos. La figura 4(b) muestra el grafo resultante  $G$  para sujetar  $O_1$  y  $O_2$  con cada  $R_i$ : para sujetar  $O_1$  con  $R_1$  es necesario quitar el conjunto  $SO_{1,1} = \{O_4\}$  y para hacerlo con  $R_2$  es necesario quitar  $SO_{2,1} = \{O_5, O_7\}$ . La figura 4(c) muestra la secuencia de acciones resultante.

## 6.2. PRENSIÓN DE OBJETOS VOLUMINOSOS

Para esta aplicación las superficies de los objetos se representan mediante nubes de puntos 3D, se consideran contactos con fricción y solo se utilizan tres dedos por mano para sujetar el objeto (6 contactos en total).

El objetivo de la aplicación es buscar tres puntos de contacto alcanzables por cada mano. Primero, la nube de puntos que describe la superficie del objeto se divide en dos conjuntos de rebanadas  $SS_h$ , donde  $h = L, R$  representan la mano izquierda y derecha respectivamente. Esto se realiza desplazando iterativamente un plano ortogonal a uno de los ejes de inercia del objeto hasta una distancia  $d$  desde el extremo, siendo  $d$  la máxima distancia que la mano puede alcanzar sobre el objeto

sin tocarlo. El eje de inercia del objeto usado es aquel que forma el ángulo más pequeño con respecto al segmento definido por los orígenes de los sistemas de referencia de las bases de los brazos. Posteriormente, los puntos que se encuentran entre dos planos se almacenan como una rebanada  $S_i^h$  (See Fig. 5).

En cada rebanada  $S_i^h$  se calcula un conjunto de tripletas  $ST_i^h = \{T_{i,j}^h, j = 1, \dots, n\}$ . Cada tripleta está compuesta por tres puntos (cada uno con su respectivo vector normal)  $T_{i,j}^h = (p_1, p_2, p_3)$ .

La selección de las tripletas se basa en tres condiciones: a) El área del triángulo formado por los tres puntos de la tripleta debe ser menor que el área del triángulo formado por los puntos centrales de las yemas de los tres dedos usados; b) La distancia entre los puntos debe ser menor que la máxima distancia permitida entre las yemas de los dedos; c) Un índice que indica que tan cerca está el triángulo definido por los tres puntos de ser equilátero debe estar por encima de un umbral dado. El índice es igual a 1 para triángulos equiláteros e igual a 0 para triángulos degenerados de área nula.

Posteriormente cada tripleta de  $ST_i^R$  se evalúa junto con cada tripleta de  $ST_i^L$  con la finalidad de encontrar un par de tripletas  $G = (T_{i,j}^R, T_{i,j}^L)$  que satisfaga la condición de equilibrio de fuerzas (force-closure, FC) con una calidad superior a un umbral determinado.

La calidad de la prensión se evalúa usando como medida la máxima perturbación que una prensión puede resistir en cualquier dirección [20]. Finalmente, se realiza un análisis de alcanzabilidad de las tripletas encontradas calculando la cinemática inversa de todo el sistema. Si existe una solución cinemática, se calculan los caminos de cada Robot para alcanzar los puntos de contacto que forman la prensión. La planificación de movimientos y el chequeo de colisiones se realizan usando las funciones de *The Kautham Project* a través del nodo #2, que después de recibir una solicitud del nodo #1, proporciona acceso a las funciones a través de los servicios **Collision\_check** y **Solve\_query**. En este caso los movimientos de ambos robots están explícitamente coordinados ya que el planificador considera a los dos Robots como un solo sistema.

## 7. EXPERIMENTACIÓN

El primer experimento aborda la manipulación de objetos en entornos desordenados. Los objetos de interés son los cilindros verdes  $O_1$  y  $O_2$  y los cilindros azules representan los objetos amovibles  $O_j, j \geq 3$  que bloquean el acceso a los objetos de interés. El objetivo del experimento es que cada

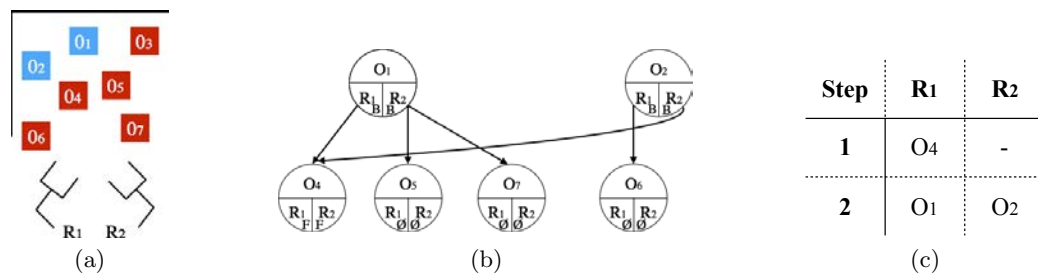


Figura 4: a) Ejemplo conceptual en 2D, b) Grafo de precedencias  $G$ , c) Secuencia de acciones resultante.

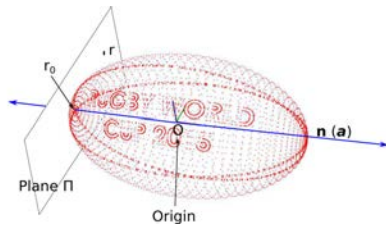


Figura 5: Balón de rugby con un plano ortogonal al eje principal para dividir el objeto en rebandas.

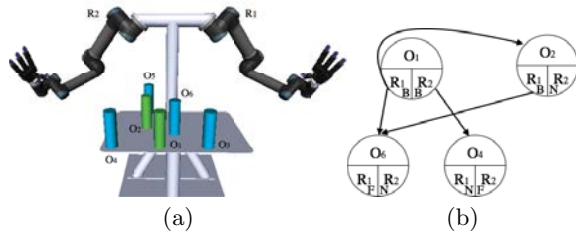


Figura 6: Experimento 1, (a) Setup del espacio de trabajo, (b) Gráfico de precedencias resultante.

Robot sujete un cilindro verde. La Figura 6(a) muestra el montaje del experimento, y la Figura 6(b) muestra el gráfico de precedencias resultante.

Como resultado se generaron las trayectorias para sujetar  $O_1$  con  $R_1$  quitando  $O_4$  con  $R_2$  y para sujetar  $O_2$  con  $R_2$  quitando  $O_6$ , que posteriormente se enviaron a los nodos #2 y #3 a través del nodo #1. En este caso el servicio **Coordinate\_robots** del nodo #2 fue activado para coordinar los planes de movimientos de cada robot. La Figura 7(a) muestra la secuencia de acciones resultante. Las Figuras 7(b-d) muestran algunas fotos de la ejecución real del experimento.

El segundo experimento aborda la presión de objetos voluminosos: un casco de bicicleta, un balón de rugby y un bote de detergente. Se asume que cada objeto está hecho de plástico con un coeficiente de fricción de 0.4 y se considera que una presión FC es aceptable si su calidad es superior a 0.03 usando la medida de calidad definida en [20].

Se realizaron 30 ensayos por objeto, obteniendo presiones con una calidad superior a 0.03 en el 92.22% de los casos, de los cuales el 86.5% tuvieron solución cinemática y de éstos se encontraron caminos libres de colisiones para un 88.4%. En la experimentación real se obtuvo un porcentaje de presiones exitosas del 65%. La incertidumbre en la posición y orientación del objeto y las fuentes de incertidumbre del sistema robótico fueron las causas principales de los experimentos fallidos. La Figura 8(a) muestra diferentes puntos de contacto de los tres objetos utilizados y las Figuras 8(b-d) muestran algunas fotografías de la experimentación real. Las trayectorias de cada Robot fueron enviadas a los nodos #2 y #3 a través del nodo #1. En este caso el servicio **Coordinate\_robots** del nodo #2 fue desactivado debido a que los caminos se coordinaron durante la planificación.

## 8. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se ha presentado un esquema de integración de componentes para la ejecución de acciones cooperativas usando sistemas bi-brazo, incluyendo una descripción de los módulos del esquema y de como ellos se interconectan usando ROS. La propuesta ha sido implementada en un sistema bi-brazo real compuesto por dos robots UR5 y dos manos antropomorfas Allegro equipadas con sensores táctiles en la yema de los dedos. Se utilizaron dos aplicaciones diferentes para mostrar la validez de la propuesta. Como trabajo futuro se pretende incluir nuevos módulos, como por ejemplo, un módulo de percepción usando cámaras y un módulo de búsqueda de presiones capaz de hacer frente a distintas fuentes de incertidumbre.

## Referencias

[1] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2009.



Step	R1	R2
1	O <sub>6</sub>	O <sub>4</sub>
2	O <sub>2</sub>	O <sub>1</sub>

(a)

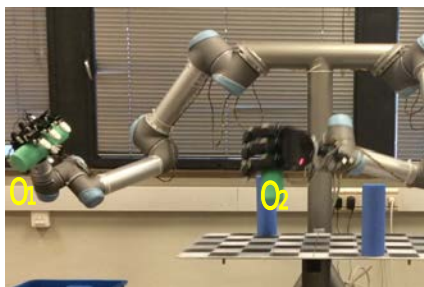
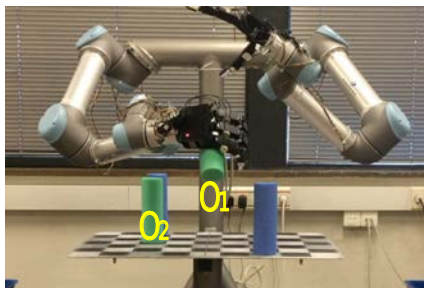
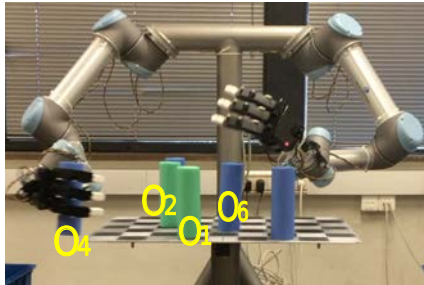


Figura 7: Experimento 1, a) Secuencias de acciones resultante, b)  $R_1$  yendo a sujetar  $O_6$  y  $R_2$  sujetando  $O_4$ , b)  $R_2$  moviendo  $O_1$  y  $R_1$  esperando para sujetar  $O_2$ , c)  $R_1$  moviendo  $O_2$  y  $R_2$  en su pose final.

- [2] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation - A survey," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1340–1353, 2012.
- [3] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Simultaneous Grasp and Motion Planning," *IEEE Robotics and Automation Magazine*, vol. 19, pp. 43–57, 2012.
- [4] C. Rodríguez, A. Montaña, and R. Suárez,

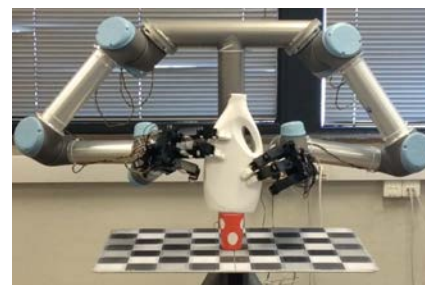
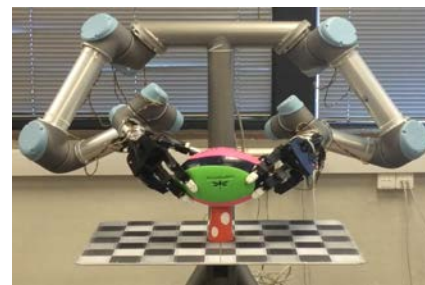
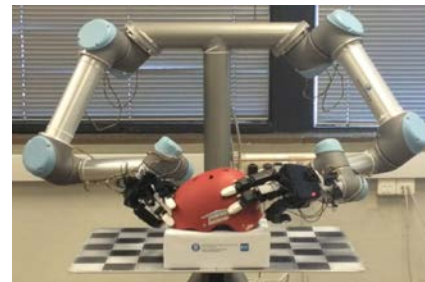
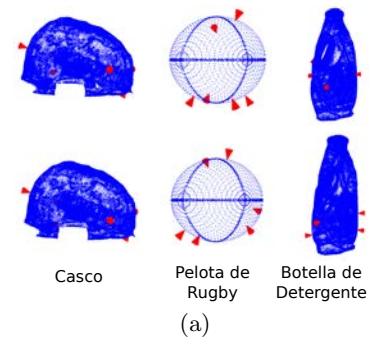


Figura 8: Experimento 2, a) Dos ejemplos de puntos de contacto alcanzables por ADARS en cada objeto. b) Prensión de un casco, c) Prensión de un balón de rugby, d) Prensión de un bote de detergente.

"Planning manipulation movements of a dual-arm system considering obstacle removing," *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1816 – 1826, 2014.

- [5] R. Suarez, J. Rosell, and N. Garcia, "Using synergies in dual-arm manipulation tasks," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 2015, pp. 5655–5661.

- [6] K. Hauser and J. C. Latombe, "Integrating task and PRM motion planning: Dealing with many infeasible motion planning queries," in *ICAPS Workshop on Bridging the Gap between Task and Motion Planning*, 2009.
- [7] M. Stilman, J.-u. Schamburek, J. Kuffner, and T. Asfour, "Manipulation planning among movable obstacles," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2007.
- [8] M. Dogar and S. Srinivasa., "A framework for push-grasping in clutter," in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.
- [9] L. Kaelbling and T. Lozano-Perez, "Hierarchical task and motion planning in the now," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 2011, pp. 1470–1477.
- [10] J. Caraza and X. Yun, "Two-handed grasping with two-fingered hands," in *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, June 1991, pp. 597–602 vol.1.
- [11] D. Berenson and S. Srinivasa, "Grasp synthesis in cluttered environments for dexterous hands," in *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, 2008.
- [12] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, "Humanoid motion planning for dual-arm manipulation and re-grasping tasks," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, oct. 2009, pp. 2464–2470.
- [13] N. Vahrenkamp, M. Przybylski, T. Asfour, and R. Dillmann, "Bimanual grasp planning," in *Humanoids*. IEEE, 2011, pp. 493–499.
- [14] J. Rosell, R. Suarez, and A. Pérez., "Safe teleoperation of a dual hand-arm robotic system," in *Robot2013: First Iberian Robotics Conference, Advances in Intelligent Systems and Computing*, Springer, Ed., 2014.
- [15] D. Kruse, J. T. Wen, and R. J. Radke, "A sensor-based dual-arm tele-robotic system," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, p. 4.18, 2015.
- [16] H. M. Do, T. Y. Choi, D. I. Park, G. J. Chung, and J. H. Kyung, "Design of software framework for system integration of dual-arm robot," in *Ubiquitous Robots and Ambient Intelligence (URAI), 2012 9th International Conference on*, 2012.
- [17] N. Vahrenkamp, M. Wächter, M. Kröhnert, K. Welke, and T. Asfour, "The robot software framework armarx," *Information Technology*, vol. 57, pp. 99–111, 2015.
- [18] J. Rosell, A. Pérez, A. Aliakbar, Muhayyuddin, L. Palomo, and N. García, "The kautham project: A teaching and research tool for robot motion planning," in *Proc. of the IEEE Int. Conf. on Emerging Technologies and Factory Automation, ETFA'14*, 2014. [Online]. Available: [sir.upc.edu/kautham](http://sir.upc.edu/kautham)
- [19] A. Montañó and R. Suárez, "An On-Line Coordination Algorithm for Multi-Robot Systems," in *18th Proc. IEEE Int. Conf. Emerging Technologies and Factory Automation, ETFA*, September 2013.
- [20] C. Ferrari and J. Cany, "Planning optimal grasps," in *IEEE International Conference on Robotics and Automation*, 1992.