

INTERFAZ DE CONTROL PARA UN ROBOT MANIPULADOR MEDIANTE REALIDAD VIRTUAL

Elena Peña-Tapia, Juan Jesús Roldán, Mario Garzón, Andrés Martín-Barrio y Antonio Barrientos
Centro de Automática y Robótica (UPM-CSIC), Universidad Politécnica de Madrid
c/ José Gutiérrez Abascal, nº2, 28006 Madrid

elena.ptapia@alumnos.upm.es, jj.rolan@upm.es, ma.garzon@upm.es, andres.mb@upm.es, antonio.barrientos@upm.es

Resumen

Las tecnologías inmersivas como la Realidad Virtual tienen potencial para mejorar la teleoperación de los robots. Estas tecnologías transportan al operador al espacio de trabajo, le proporcionan información sobre la misión y le permiten comandar los robots de manera intuitiva. En este trabajo se ha integrado un robot manipulador Kinova Jaco² en un entorno de realidad virtual visualizado con las gafas de realidad virtual HTC Vive. La interfaz permite tanto la monitorización del brazo robótico (estado de las articulaciones, localización del efector, acciones sobre su entorno...) como el comando del mismo (mediante el envío de metas al efector). Además, el trabajo aborda con éxito la conexión entre el mundo de los robots, dominado por Robot Operating System (ROS), y el mundo de la realidad virtual, recreado mediante Unity y Steam VR. Las primeras pruebas con operadores muestran el potencial de esta interfaz: la transmisión de información espacial, el sistema de comando intuitivo y la operación en condiciones de seguridad. En trabajos futuros se abordará la integración de información sobre el entorno obtenida mediante cámaras 3D, así como la visualización de las trayectorias del robot manipulador hacia sus metas.

Palabras clave: Robótica, Robot Manipulador, Interfaz, Realidad Virtual.

1 INTRODUCCIÓN Y OBJETIVOS

En los últimos años las misiones robóticas han experimentado un gran desarrollo y se han aplicado en múltiples campos como la protección de infraestructuras [4], monitorización ambiental [11], etc. Estas misiones presentan dificultades ligadas a la utilización de múltiples robots, la realización de tareas complejas o el despliegue en escenarios desestructurados.

Algunos de los problemas de factores humanos en las misiones robóticas son el exceso de carga de trabajo, la falta de conciencia de situación, un in-

adecuado nivel de estrés que pueda dar lugar a estados de ansiedad o aburrimiento y una falta o exceso de confianza en la automatización. Estos problemas pueden provocar que las decisiones del operador sean incorrectas y sus acciones de control provoquen desde pérdidas de eficiencia hasta accidentes con los robots.

Estos retos pueden ser abordados mediante el desarrollo de interfaces que asuman algunas funciones del operador, transmitan la información adecuada de la forma adecuada, mantengan su atención durante la misión y favorezcan la toma de decisiones. En particular, estas interfaces exploran posibilidades como las interacciones multimodales, las tecnologías inmersivas, los componentes predictivos y los paneles adaptativos.

En este trabajo se ha desarrollado una interfaz mediante realidad virtual para la monitorización y el control de un robot manipulador. Esta interfaz emplea tecnologías inmersivas (realidad virtual) para introducir al operador en el entorno de trabajo y mejorar su percepción sobre las acciones del robot, así como interacciones multimodales (mandos con vibración) para reforzar la información visual y facilitar el comando del robot.

El resto de esta comunicación está organizada de la siguiente manera: La sección 2 recopila algunos desarrollos actuales de interfaces inmersivas para misiones robóticas. La sección 3 presenta el sistema formado por el robot manipulador y la interfaz en realidad virtual. La sección 4 describe el robot manipulador, así como su sistema de planificación y control de trayectorias. La sección 5 describe la interfaz de realidad virtual y aborda tanto la visualización de la información como la generación de los comandos. Por último, la sección 6 resume los principales resultados del trabajo y las posibles continuaciones del mismo.

2 ESTADO DEL ARTE

Las interfaces de operador para robots deben integrar estas informaciones [1]: las medidas de los sensores de los robots, los comandos enviados a los robots, el estado actual de los robots (por ejemplo, el nivel de batería), el estado actual de las tareas

(por ejemplo, el porcentaje de realización) y los mapas necesarios para la navegación.

El rendimiento de estas interfaces puede verse afectado por los problemas que se presentan en la tabla 1. Factores como la limitación del campo visual, la pérdida de la orientación o la degradación de la percepción de la profundidad han de tenerse en cuenta en su diseño. El desarrollo de interfaces inmersivas es una de las estrategias más prometedoras para abordar estos problemas.

Tabla 1: Problemas en la teleoperación.

Problema	Posible solución
Limitación del campo visual	Inmersión Múltiples cámaras
Desorientación	Mapas
Pérdida del contexto	Inmersión
Degradación de la profundidad	Inmersión
Retardo en las comunicaciones	Predicción Adaptación

A día de hoy existen tres tipos de tecnologías inmersivas: la realidad aumentada (RA), la realidad virtual (RV) y la realidad mixta (RM).

La realidad aumentada superpone elementos virtuales sobre los vídeos de los robots. Estos elementos virtuales pueden incluir mapas, elevación del terreno, obstáculos, rutas de los robots, posiciones de los objetivos y otros datos relevantes. La literatura contiene ejemplos de interfaces de realidad aumentada para robots manipuladores [5], robots quirúrgicos [15] y flotas con múltiples robots [6].

La realidad virtual recrea escenarios combinando representaciones de elementos reales (por ejemplo, robots y obstáculos) y elementos virtuales (por ejemplo, rutas y objetivos). En la literatura se pueden encontrar desarrollos de interfaces de realidad virtual para robots móviles [9] y manipuladores industriales [10]. Algunos estudios empiezan a demostrar el potencial de estas interfaces para mejorar la conciencia de la situación de los operadores sin incrementar sustancialmente su carga de trabajo [12].

Por último, la realidad mixta combina aumentada y virtual, generando escenarios en los que los operadores pueden interactuar con objetos tanto reales como virtuales. Esta tecnología es reciente y apenas se ha implementado en interfaces, pero existe algún desarrollo para robots industriales [3] y móviles [7].



Figura 2: Brazo robótico Jaco² de Kinova.

3 DESCRIPCIÓN DEL SISTEMA

La integración del robot manipulador en la realidad virtual requiere la conexión entre el mundo real y el mundo virtual, tal y como se muestra en la figura 1. El mundo real (descrito en la sección 4) está formado por el robot manipulador y su espacio de trabajo y está dominado por *Robot Operating System (ROS)*, dado que este sistema se emplea tanto para la planificación como para el control de este robot. Por su parte, el mundo virtual (descrito en la sección 5) contiene la interfaz de operador y está controlado por *Unity* y *Steam VR*, ya que estas herramientas son utilizadas para la reproducción y la visualización de la realidad virtual.

4 ROBOT MANIPULADOR

Para el desarrollo de este trabajo se ha utilizado un robot manipulador Jaco² fabricado por *Kinova Robotics*. Se trata de un brazo robótico de seis grados de libertad que incluye un efector final formado por tres dedos plásticos flexibles. Soporta una carga de peso de 1,6 kg y consigue un alcance de 900 mm. Este brazo permite también realizar control de fuerza, haciendo que sea seguro para trabajar de forma colaborativa con personas. El robot manipulador formado por el brazo y su efector pueden observarse en la figura 2.

El control del robot se realiza a través de una conexión USB o Ethernet entre un ordenador, en el que se ejecuta la planificación de trayectorias y demás tareas de alto nivel, y un controlador embebido en el brazo, en el que se ejecuta el control de las articulaciones y, en caso de ser necesaria, la cinemática inversa.

La planificación de los movimientos del brazo utiliza como herramienta principal la *Open Motion Planning Library (OMPL)* [13], que es una librería

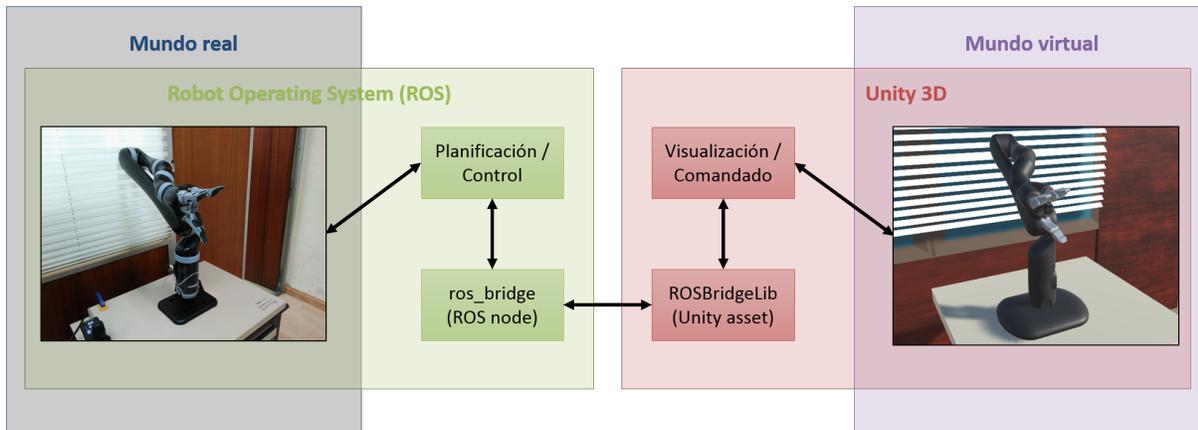


Figura 1: Esquema del sistema formado por el robot manipulador y la realidad virtual.

de código abierto que implementa varios algoritmos de planificación basados en muestreo: algunos ejemplos son PRM, RRT, EST, SBL, KPIECE y SyCLOP. Esta librería está integrada en el software de manipulación móvil *MoveIt!*, que provee funcionalidades cinemáticas como la comprobación de colisiones, la percepción 3D y otras herramientas.

Para los experimentos realizados, se utilizó el planificador *RRT-connect* [8]. Primero, este planificador obtiene un estado aleatorio q_r en el espacio articular. Luego busca entre los estados previamente observados el más cercano a q_r y lo denota como q_c . Finalmente expande el árbol, teniendo en cuenta las capacidades cinemáticas del robot, desde el estado q_c en dirección a q_r hasta llegar a un estado válido q_m , el cual es añadido al árbol de exploración. La figura 3 muestra un caso de uso de esta herramienta en el que el robot manipulador alcanza las metas enviadas desde la realidad virtual.

La configuración del robot en *MoveIt!* se realiza mediante un fichero Unified Robot Description Format (URDF), que utiliza un formato XML para representar el modelo del robot, sus sensores y la escena en la que se encuentra. El modelo utilizado parte de un fichero básico proporcionado por el fabricante, al cual se agrega la descripción de la escena y los sensores adicionales que se utilizan. A partir de este modelo URDF, *MoveIt!* obtiene la información necesaria para el verificador de colisiones y los demás elementos necesarios para la planificación del movimiento.

Una vez realizada la configuración inicial, *MoveIt!* se comunica con el driver del robot para obtener la posición de cada articulación y demás parámetros necesarios para conocer su estado actual del robot. La configuración final o la posición del robot puede ser enviada a *MoveIt!* mediante una interfaz de

control o a través de un mensaje de ROS. Una vez recibida la meta, se realiza la planificación y se obtiene la trayectoria de cada articulación, que es enviada de vuelta al driver del robot, que se encarga de comunicarse con el brazo y controlar los movimientos del mismo.

Para poder controlar el manipulador desde la interfaz de realidad virtual es necesario establecer una comunicación bidireccional entre el entorno *Unity* y los componentes de software del robot, los cuales trabajan bajo ROS. Para esto se utiliza el componente *RosBridge*¹, que permite comunicar nodos de ROS con programas externos, convirtiendo los mensajes estándar de ROS en mensajes JSON, que pueden ser enviados a través de una interfaz web. *RosBridge* establece un protocolo para el establecimiento de las comunicaciones, es decir, para suscribirse o publicar en un tópico de ROS. Además, este paquete posee una implementación que permite el envío de mensajes utilizando web-sockets.

Como se mencionó anteriormente, la comunicación entre *RosBridge* y *Unity* es bidireccional. *RosBridge* recibe de *Unity* los mensajes de control, formados por comandos de movimiento y otras señales que controlan la ejecución de la misión. Por otra parte, *RosBridge* envía de vuelta el estado del robot, que consiste en la posición de cada una de las articulaciones, y en caso de estar disponibles, las medidas de los sensores y cualquier otra información que permita al operador tener un conocimiento adecuado de la situación.

La figura 4 muestra un esquema general de las comunicaciones entre el robot y *Unity*. Como se puede observar, una vez recibidos los comandos desde *RosBridge*, un nodo intermedio se encarga de procesarlos y convertirlos en mensajes que pueden ser interpretados como órdenes de con-

¹http://wiki.ros.org/rosbridge_suite

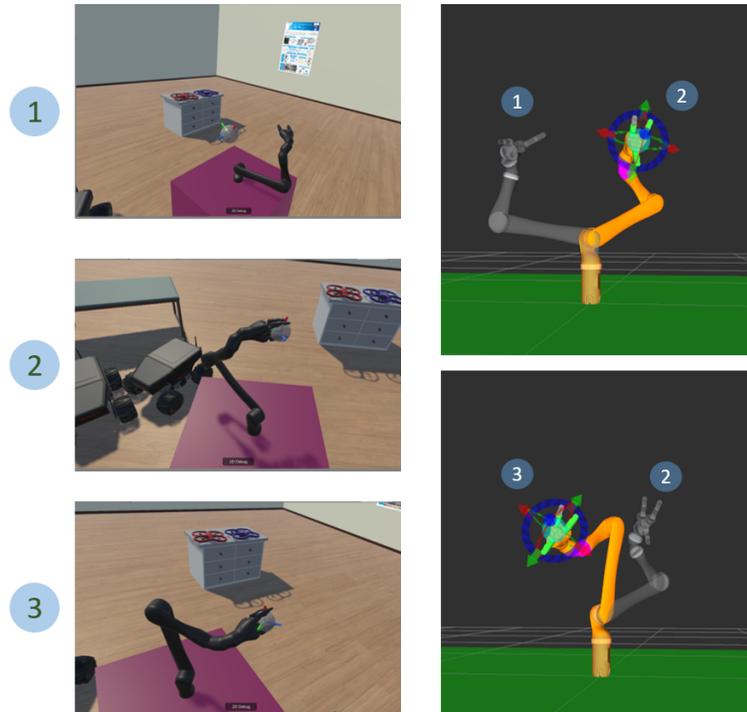


Figura 3: Planificación de los movimientos del robot manipulador.

trol válidas para *MoveIt!*, este nodo envía también las señales necesarias para iniciar, detener o reiniciar el movimiento del brazo. *MoveIt!* procesa estos mensajes, realiza la planificación de movimientos y envía las trayectorias articulares al controlador del robot. Este último se encarga de la comunicación con el controlador embebido del robot y envía de vuelta mensajes con el estado de las articulaciones. *RosBridge* envía este mensaje directamente a *Unity*, donde se procesa para ser presentado al operador.

5 REALIDAD VIRTUAL

Para el desarrollo de la interfaz por realidad virtual se ha utilizado el entorno *Unity* y para su visualización se han empleado las gafas de realidad virtual *HTC Vive* y el programa *Steam VR*. *Unity* es un entorno de desarrollo de videojuegos que proporciona multitud de recursos como escenarios realistas, modelos dinámicos y cinemáticos, personajes con inteligencia artificial, etc. El conjunto *HTC Vive* incluye un casco con gafas de realidad virtual y auriculares, dos controladores con botones y vibración y dos estaciones base para posicionar estos elementos en la habitación. Por último, *Steam VR* es un complemento para *Unity* necesario para compilar y ejecutar los programas en la realidad virtual.

El modelo del robot se ha introducido en *Unity* descompuesto en ocho eslabones: base, hombro,

brazo, antebrazo, muñeca 1, muñeca 2, mano y dedos. Cada eslabón se ha implementado mediante dos objetos: uno para ensamblarlo con el anterior con la posición y la orientación adecuadas y otro para permitir el movimiento de su articulación. El resultado es un robot virtual con posiciones y movimientos que coinciden con los del robot real.

Además, se han diseñado dos escenarios para el robot manipulador: uno para pruebas en el que el robot virtual se encuentra en un escenario vacío y otro para demostraciones que reproduce el laboratorio del robot real. La herramienta *Unity* y, sobre todo, la amplia gama de recursos disponibles en internet permite el modelado de todo tipo de escenarios interiores o exteriores: por ejemplo, casas, oficinas, industrias, cultivos, bosques...

Para el comando del manipulador se plantearon varios sistemas intuitivos y finalmente se optó por una esfera con tres ejes. La esfera representa la posición que tiene que alcanzar el efector del robot, mientras que los tres ejes definen la orientación que debe adquirir el mismo. Cuando la esfera está en el efector, el robot permanece en reposo. En cualquier momento, el operador puede agarrar la esfera, desplazarla por la escena y soltarla en otra localización. Cuando el operador suelta la esfera, el robot se desplaza hasta esta localización.

Además, se ha introducido un sistema de teletransporte que permite al operador escoger el

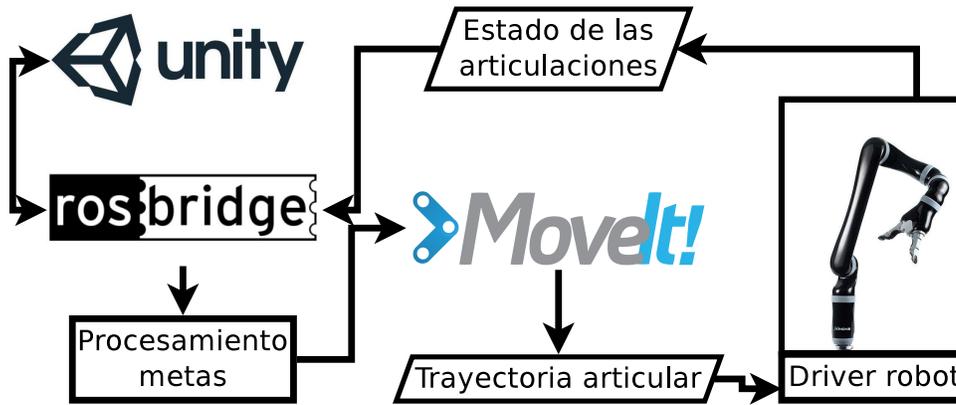


Figura 4: Esquema de comunicaciones del robot con el entorno *Unity*.

punto de vista desde el que observar la escena y comandar el robot. Para teletransportarse, el usuario sólo tiene que presionar un botón en el mando y dirigir la parábola hacia el punto deseado del suelo (ver figura 5). Este sistema es especialmente útil para tener una visión adecuada de las tareas de manipulación y enviar las metas del efector de forma precisa.

En el comandado del manipulador intervienen varios scripts. El primero de ellos (*Interactable*) se desarrolló tomando como base un ejemplo de *Steam VR* y controla los movimientos de la esfera de comandado, permitiendo al usuario cogerla, desplazarla y soltarla. El resto de los scripts pertenecen a la librería *ROSBridgeLib* [2] y han sido adaptados para trabajar con los mensajes que emplea el robot manipulador.

El script *RealSenseViewer* establece la comunicación por socket con el paquete *RosBridge* y crea los suscriptores y publicadores pertinentes. En este caso, el programa desarrollado en *Unity* se suscribe a un mensaje con las posiciones de las articulaciones (*sensor_msgs/JointState*) y publica otro mensaje con las metas para el efector (*geometry_msgs/Pose*).

Cada mensaje que se desee intercambiar entre *Unity* y *ROS* se debe implementar mediante dos scripts. El primero tiene que definir la estructura del mensaje, proporcionar constructores a partir de un *JSONNode* (mensaje recibido por el socket) y de cada una de las variables del mensaje y proporcionar métodos para acceder a dichas variables. El segundo se ocupa de establecer el tipo de mensaje y el tópicos en el que se debe recibir.

6 RESULTADOS Y CONCLUSIONES

Los resultados de las primeras pruebas con operadores indican que la interfaz aporta beneficios

relacionados con el conocimiento de la situación y el comandado intuitivo del robot. Los operadores valoran positivamente la libertad para elegir el punto de vista desde el que observan el robot, así como la facilidad de comandar el robot moviendo la meta de su efector.

En trabajos futuros se reforzará la información visual del robot y su entorno, mostrando el espacio de trabajo, la manipulabilidad y las trayectorias planificadas. Además, se prevé el desarrollo de un método de comandado directo que permita modificar una a una las orientaciones de las articulaciones. Por último, se estudiará el modelado automático del entorno del robot, empleando cámaras 3D y mostrando las nubes de puntos.

Agradecimientos

La investigación que ha llevado a estos resultados ha recibido fondos de los proyectos RoboCity2030-III-CM (Robótica aplicada a la mejora de la calidad de vida de los ciudadanos. fase III; S2013/MIT-2748), financiado por los Programas de Actividades I+D en la Comunidad de Madrid y cofinanciado por los Fondos Estructurales de la Unión Europea, DPI2014-56985-R (Protección robotizada de infraestructuras críticas), financiado por el Ministerio de Economía y Competitividad del Gobierno de España, y SAVIER (Situational Awareness Virtual Environment), dirigido y financiado por Airbus Defence & Space.

Referencias

- [1] Chen, J.Y.C., Haas, E.C., Barnes, M.J., (2007) Human performance issues and user interface design for teleoperated robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1231-1245.
- [2] Codd-Downey, R., Forooshani, P. M., Speers, A., Wang, H., Jenkin, M. (2014). From ROS

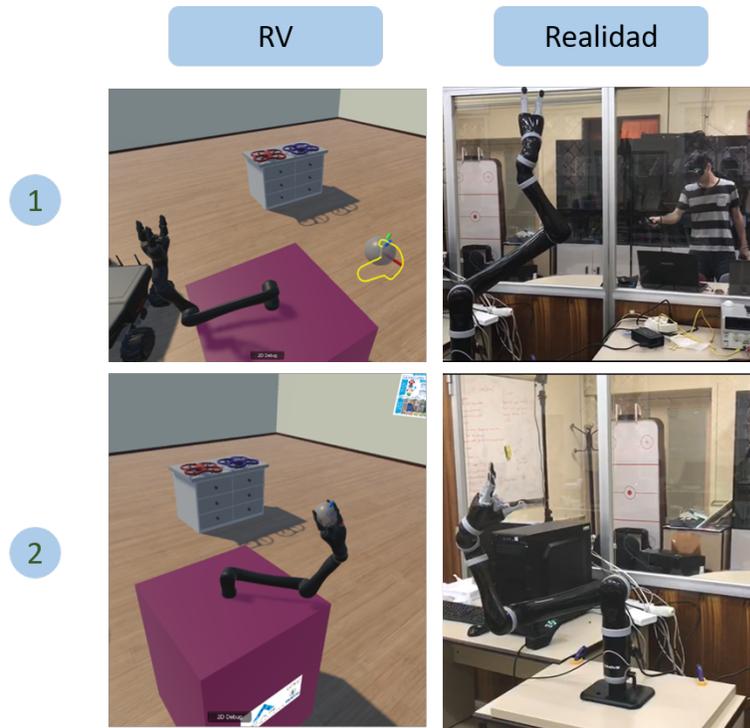


Figura 5: Comandado del robot manipulador.

- to unity: Leveraging robot and virtual environment middleware for immersive teleoperation. In *Information and Automation (ICIA), 2014 IEEE International Conference on* (pp. 932-936). IEEE.
- [3] Foit, K. (2014). Mixed reality as a tool supporting programming of the robot. In *Advanced Materials Research* (Vol. 1036, pp. 737-742). Trans Tech Publications.
- [4] Garzón, M., Valente, J., Roldán, J. J., Cancar, L., Barrientos, A., Del Cerro, J. (2015). A multirobot system for distributed area coverage and signal searching in large outdoor scenarios. *Journal of Field Robotics*.
- [5] Gaschler, A., Springer, M., Rickert, M., Knoll, A. (2014). Intuitive robot tasks with augmented reality and virtual obstacles. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (pp. 6026-6031). IEEE.
- [6] Ghiringhelli, F., Guzzi, J., Di Caro, G. A., Caglioti, V., Gambardella, L. M., Giusti, A. (2014). Interactive augmented reality for understanding and analyzing multi-robot systems. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on* (pp. 1195-1201). IEEE.
- [7] Hoenig, W., Milanés, C., Scaria, L., Phan, T., Bolas, M., Ayanian, N. (2015). Mixed reality for robotics. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (pp. 5382-5387). IEEE.
- [8] Kuffner, J. J., LaValle, S. M. (2000). RRT-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on* (Vol. 2, pp. 995-1001). IEEE.
- [9] Lin, A., Milshteyn, A., Herman, G., Garcia, M., Liu, C., Rad, K., Boussalis, H. (2014). Virtual reality head-tracking observation system for mobile robot. In *Embedded Computing (MECO), 2014 3rd Mediterranean Conference on* (pp. 152-157). IEEE.
- [10] Matsas, E., Vosniakos, G. C. (2017). Design of a virtual reality training system for human-robot collaboration in manufacturing tasks. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 11(2), 139-153.
- [11] Roldán, J. J., Garcia-Aunon, P., Garzón, M., de León, J., del Cerro, J., Barrientos, A. (2016). Heterogeneous Multi-Robot System for Mapping Environmental Variables of Greenhouses. *Sensors*, 16(7), 1018.
- [12] Ruiz, J. J., Viguria, A., Martínez-de-Dios, J. R., Ollero, A. (2015). Immersive displays

for building spatial knowledge in multi-UAV operations. In Unmanned Aircraft Systems (ICUAS), 2015 International Conference on (pp. 1043-1048). IEEE.

- [13] Sucas, I. A., Moll, M., Kavraki, L. E. (2012). The open motion planning library. IEEE Robotics & Automation Magazine, 19(4), 72-82.
- [14] Sucas, I. A., Chitta, S. (2013). Moveit!. Online at <http://moveit.ros.org>.
- [15] Wen, R., Tay, W. L., Nguyen, B. P., Chng, C. B., Chui, C. K. (2014). Hand gesture guided robot-assisted surgery based on a direct augmented reality interface. Computer methods and programs in biomedicine, 116(2), 68-80.