ISBN 978-84-15914-12-9

INTEGRACIÓN DE DISPOSITIVOS EN UN ROBOT QUIRÚRGICO TELEOPERADO MEDIANTE ROS

E. Bauzano[1], A. Fernández-Iribar[2], M.C. López-Casado[1], J. Klein[2], A. Rentería[2], V.F. Muñoz-Martínez[1]
[1]Universidad de Málaga, Andalucía Tech, Institutos Universitarios, Severo Ochoa, 4, 29590, Málaga, España, ebauzano@uma.es
[2]Tecnalia R&I, Mikeletegi Pasealekua 1, 20009, San Sebastián, España, asier.fernandez@tecnalia.com

Resumen

A lo largo de los años la robótica se ha ido convirtiendo en una especialidad cada vez más compleja, en la que interaccionan más y más dispositivos. El problema es que a menudo estos dispositivos tienen diferentes sistemas de comunicación y su integración no resulta sencilla. Para solucionar estos problemas, en este artículo se describe la metodología empleada para realizar las comunicaciones entre los distintos dispositivos de un sistema robótico para cirugía teleoperada mediante el software ROS. Todo ello ha sido estructurado y jerarquizado mediante una arquitectura UML en la que se diferencia, por un lado, una capa de abstracción de bajo nivel en la que se gestionan las entradas/salidas de cada dispositivo, y por otro una capa de comunicaciones de alto nivel que integra toda la información sobre el estado de los mismos. Los resultados de este desarrollo han sido puestos a prueba un prototipo de robot quirúrgico teleoperado dotado de tres brazos manipuladores.

Palabras Clave: robot quirúrgico, arquitectura UML, comunicaciones, sistema tolerante a fallos, ROS

1 INTRODUCCIÓN

La robótica es un campo de estudio en constante evolución, en el que se emplean tecnologías de última generación que permitan resolver problemas cada vez más avanzados. Esto implica que a menudo los desarrolladores deben lidiar con dispositivos que no suelen estar preparados para una fácil integración con el resto de su sistema. Del mismo modo, al aumentar la complejidad de estos problemas también suele el personal perteneciente a los equipos de desarrollo. No resulta extraño encontrar colaboraciones entre varios grupos de trabajo que ni siquiera tengan por qué ubicarse en las mismas infraestructuras o incluso provincia o país. En estos casos la tarea de integración de los distintos paquetes de trabajo puede volverse extremadamente compleja si no se sigue un procedimiento muy estricto.

Para resolver muchos de estos problemas la "Open Source Robotics Foundation" creó el *Robot Operating System* (ROS) [1]. El uso de este sistema de código abierto presenta muchas ventajas para los desarrolladores, como la independencia del lenguaje usado para programar cada dispositivo, herramientas para visualizar la información transmitida entre los dispositivos en tiempo de ejecución, escalabilidad en el código o una integración sencilla de los distintos dispositivos. En particular, ROS también incluye características que facilitan la simulación de dispositivos reales tales como brazos manipuladores, robots móviles o sensores láser [2].

Cada vez más grupos de investigación están empleando ROS para programar e integrar los distintos dispositivos y elementos de control en sus trabajos de desarrollo. Así, por ejemplo se ha diseñado un Vehículo Autónomo Submarino (AUV) cuya implantación ha sido posible gracias a ROS [3]. Otros trabajos han centrado su desarrollo en interfaces de usuario como el uso de tabletas en robots móviles [4] o la teleoperación bimanual de robots humanoides [5]. También hay trabajos que aprovechan el enorme potencial de las comunicaciones de ROS para desarrollar sistemas de comunicación entre múltiples robots basados en la nube [6], o bien aprovechan las herramientas de modelado para crear una arquitectura de diagnóstico y reparación de sistemas robóticos [7].

La robótica quirúrgica [8] es un campo de desarrollo que puede aprovechar las ventajas que ofrece ROS. En concreto, este artículo presenta una metodología que permite abstraer la gran diversidad en los protocolos de comunicaciones de cada dispositivo, a la par que facilita las tareas de integración de desarrollos realizados por distintos equipos de trabajo. Para ello se utilizará como modelo de estudio un robot quirúrgico teleoperado para laparoscopia con tres brazos manipuladores.

De este modo, en primer lugar en el apartado 2 se presenta una descripción del funcionamiento del sistema y de los elementos que lo integran. Cada dispositivo tiene una primera capa de abstracción de bajo nivel y que se describe a través de la

ISBN 978-84-15914-12-9

arquitectura UML propuesta en el apartado 3. Las comunicaciones de todos los dispositivos de este sistema han sido integradas según se describe en el apartado 4. Por último, en el apartado 5 se describe cómo ha sido el proceso de desarrollo e integración de todos los dispositivos introducidos en el capítulo 2 mediante el ROS.

2 DESCRIPCIÓN DEL SISTEMA

El número mínimo de instrumentos necesarios para realizar una operación laparoscópica son 3: la cámara laparoscópica y dos pinzas manejadas por ambas manos del cirujano, y elegidas según la fase de la intervención. La posición en la que ubicar el instrumental difiere en función del tipo de cirugía a realizar. Además, lo usual es que también aparezcan en el campo quirúrgico al menos el anestesista y un asistente al cirujano.

A partir de estos requisitos básicos, se ha diseñado un concepto de robot quirúrgico tal y como aparece en la Figura 1. Dicho robot consiste en tres brazos manipuladores, cada uno de ellos instalado sobre estructuras móviles con ruedas denominadas *Unidades Robotizadas* y que puede fijarse mediante unas patas actuadas manualmente en cualquier posición alrededor del paciente. Cada una de estas estructuras tendrá una misión concreta y actuará como *Cámara*, *Pinza Izquierda* o *Pinza Derecha*.

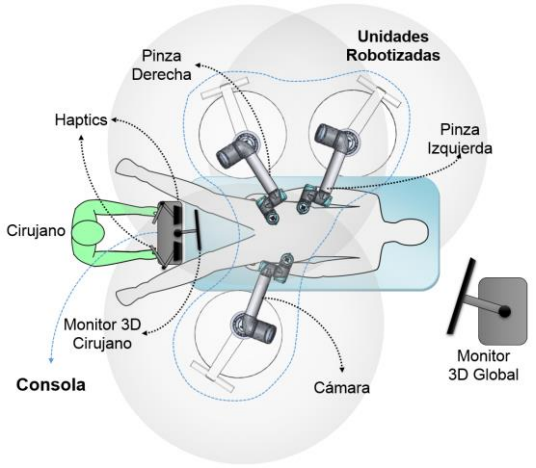


Figura 1: Posible disposición en quirófano del robot quirúrgico propuesto.

Este sistema de 3 brazos articulados es controlado por el *Cirujano* de forma teleoperada a través de una estructura denominada *Consola*, la cual puede ubicarse en cualquier lugar del quirófano. Dicha estructura dispone de dos dispositivos denominados *Haptics* con los que controlar el movimiento de los brazos, así como de dos pedales. Uno de ellos activa el modo de movimiento de la *Cámara*, mientras que el otro permite controlar los brazos que sostienen el instrumental quirúrgico. A su vez, la consola incluye un *Monitor 3D* que permite la visión en tres dimensiones con el uso de gafas polarizadas.
La metodología empleada para controlar todo el sistema puede resumirse en tres niveles:

1. **Planificación de movimientos.** En este primer nivel se asocian los movimientos de los haptics con los del extremo distal del instrumental laparoscópico, así como la apertura/cierre de las pinzas [9].
2. **Localización del fulcro.** En un segundo nivel el fulcro, o punto de inserción de las herramientas en el paciente alrededor del cual se realizan movimientos esféricos, debe estar bien localizado en todo momento para evitar daños en la pared abdominal. Este control se realiza mediante una realimentación de las fuerzas ejercidas por la herramienta [10].
3. **Realimentación de fuerzas háptica.** Por último se ha desarrollado un tercer nivel de control en el que se envían las fuerzas ejercidas por el extremo distal de las herramientas al haptic correspondiente. De este modo, el cirujano puede percibir sensaciones de contacto entre las herramientas y el interior del paciente [11-13].

Estos tres niveles de control han sido integrados en cada una de las unidades expuestas previamente para garantizar un movimiento seguro de los brazos manipuladores. A continuación se describen en detalle los dispositivos que integran cada una de estas estructuras y que serán gestionados por el método de control de tres niveles propuesto.

2.1 UNIDAD ROBOTIZADA

La unidad robotizada corresponde a una de las estructuras que soporta a un brazo manipulador. El último prototipo de esta unidad puede verse en la Figura 2. Cada una de estas estructuras dispone de 2 baterías de 100Ah para funcionar sin necesidad de conexión a la red eléctrica. El funcionamiento inalámbrico no es obligatorio, y para ello la unidad robotizada incluye una fuente de alimentación conectable a corriente eléctrica que permite tanto el funcionamiento cableado como la carga de las baterías. Del mismo modo, la comunicación entre cada unidad robotizada se realiza a través de un protocolo estándar Ethernet, que puede funcionar tanto con cables como de forma inalámbrica.

Todos los elementos descritos hasta ahora sirven para desempeñar como única tarea el suministro eléctrico y realizar las comunicaciones con el resto de unidades. Aparte de esta infraestructura básica, cada estructura dispone de un conjunto de dispositivos funcionales encargados de ejecutar los tres niveles de control presentados anteriormente:

ISBN 978-84-15914-12-9

Figura 2: Prototipo de unidad robotizada.

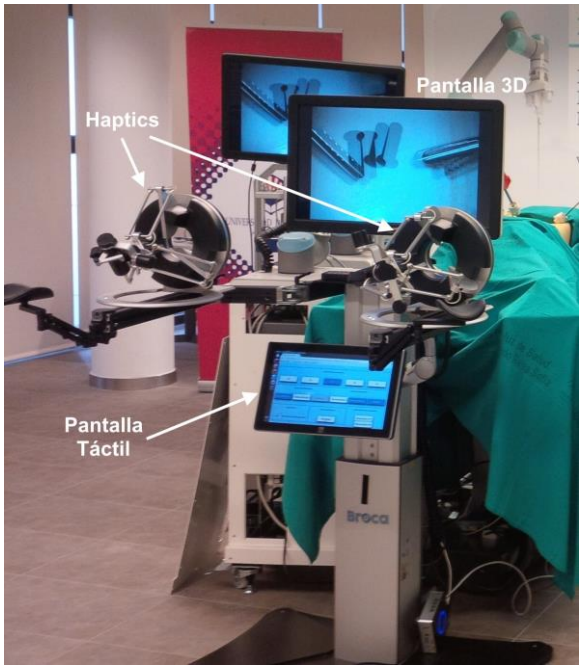


Figura 3: Prototipo de la consola.

- **Brazo manipulador**. Es el dispositivo principal que permite interactuar con el instrumental quirúrgico. Se controla a través de su caja de control, la cual gestiona todas las órdenes de movimiento y planificación de bajo nivel recibidas desde el PC Industrial.
- **PC Industrial**. Es el núcleo de computación de la unidad robotizada. Se encarga de ejecutar los algoritmos de control de movimientos y de fuerzas. También gestiona la comunicación entre los dispositivos que la integran, así como la comunicación con las otras unidades robotizadas y/o consola.
- **Sensor de fuerzas**. Recoge la medida de fuerzas y pares ejercidos por el instrumental manejado por el brazo manipulador. Estas mediciones resultan esenciales para el correcto funcionamiento del control de posicionamiento del fulcro y la realimentación háptica.
- **Actuador de la pinza**. Consiste en un motor integrado con un mecanismo capaz de abrir y/o cerrar la pinza de la herramienta quirúrgica. Dicho motor se actúa directamente a través de uno de los Grados de Libertad (GDL) de los haptics.
- **Sensor de agarre**. Posee un solo grado de libertad para medir y realimentar al haptic la fuerza con la que se está agarrando algún tejido u objeto.

2.2 CONSOLA

La consola (ver Figura 3) es una estructura bien diferenciada de las unidades robotizadas, ya que su misión es actuar de interfaz entre el cirujano y los brazos manipuladores.

Los principales dispositivos que integran la consola son los siguientes:

- **PC Industrial**. Al igual que ocurre con las unidades robotizadas, la consola debe disponer de un dispositivo de computación para ejecutar los algoritmos de control de fuerzas y los parámetros de operación deseados.
- **Dispositivos hápticos**. El cirujano dispone de dos de estos dispositivos para controlar hasta dos herramientas quirúrgicas. Los haptics miden las trayectorias deseadas para cada herramienta y se envían al brazo correspondiente. Además, estos haptics pueden aplicar fuerzas sobre la mano del cirujano. De este modo se pueden transmitir las fuerzas medidas por cada unidad robotizada para así crear una sensación de contacto similar a la que está ejerciendo el robot.
- **Pantalla táctil**. Desde este dispositivo el cirujano puede controlar los parámetros generales de operación, tales como la velocidad a la que se mueven los brazos manipuladores, parámetros generales de la imagen…
- **Pedales**. Con ellos se puede activar el movimiento de los brazos manipuladores. Según el pedal pulsado se activarán los que manejan el instrumental quirúrgico o la cámara laparoscópica, pero nunca ambos simultáneamente. Estos pedales evitan el movimiento accidental de alguno de los brazos mediante la filosofía dead-man (hombre muerto).

Una vez detallada cada una de las estructuras del sistema robótico, el siguiente paso consiste en

ISBN 978-84-15914-12-9

describir el proceso de integración de todos los elementos del sistema que sigue una jerarquía de dos capas: capa de abstracción y capa de comunicaciones.

3 CAPA DE ABSTRACCIÓN

Aunque habitualmente cada dispositivo incluye su propia controladora, la forma de acceder a su información así como el envío de comandos suele ser diferente para cada caso en particular. Por este motivo, el primer paso en la integración de un sistema consiste en crear una capa lógica que transforme esta información de manera que nuestro sistema pueda acceder mediante un mismo protocolo con independencia del dispositivo.

Para este propósito se ha diseñado una arquitectura UML en la que se describen las clases de cada dispositivo del sistema completo. Como puede verse en la Figura 4, todos los dispositivos heredan de una clase principal denominada *Device*. Esta clase describe los parámetros generales que todo dispositivo deberá incluir, como son la frecuencia de muestreo o el método de lectura de datos. A su vez, esta clase hereda de *DeviceError*, cuya finalidad es manejar posibles errores que aparezcan en el dispositivo.

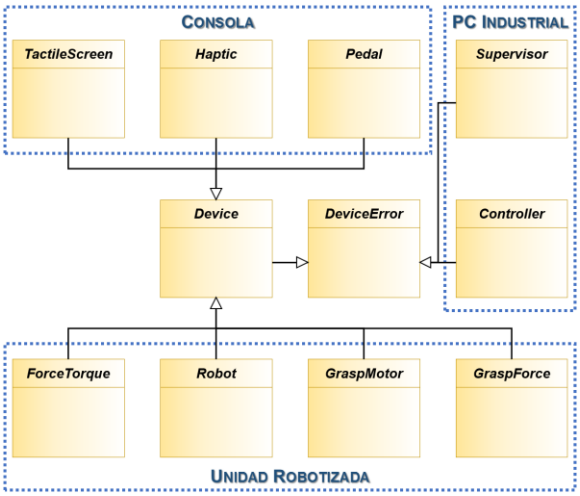


Figura 4: Arquitectura UML para la capa de abstracción.

La arquitectura UML de la Figura 4 muestra diferenciadas las clases de los dispositivos en función de la estructura a la que pertenecen: Unidad Robotizada o Consola. En el lado de la unidad robotizada aparecen las clases *ForceTorque* (sensor de fuerzas), *Robot* (brazo manipulador), *GraspMotor* (actuador de la pinza) y *GraspForce* (sensor de agarre). Por su parte, en la consola se encuentran las clases *TactileScreen* (pantalla táctil), *Haptic* (dispositivo háptico) y *Pedal* (pedales).

Existe un tercer grupo catalogado como PC Industrial con dos clases a priori no relacionadas con ningún dispositivo: *Supervisor* y *Controller*. En realidad estas clases se corresponden con la gestión y ejecución del flujo de trabajo del sistema robótico desde el PC Industrial, de ahí que reciban un trato sensiblemente diferente del resto de dispositivos. Por un lado, la clase *Supervisor* se encarga de comprobar el estado de todos los dispositivos y modificar la actuación del sistema robótico en caso de que se produzca algún error mediante control tolerante a faltas [14]. Por otro lado, la clase *Controller* es la encargada de aplicar los algoritmos de control de los tres niveles vistos en el apartado 2.

4 CAPA DE COMUNICACIONES

Una vez desarrollada la capa de abstracción del tipo de comunicaciones propietario de cada dispositivo, el siguiente paso consiste compartir esta información en todo el conjunto de unidades robotizadas y consola.

Para conseguir este propósito, a la arquitectura UML presentada en el apartado anterior se le añade una capa de comunicaciones, tal y como puede verse en la Figura 5. Por simplicidad se ha simplificado la Capa de Abstracción para que sólo incluya los dispositivos de una unidad robotizada, pero la extensión a los elementos que integran la consola se realizaría de forma análoga.

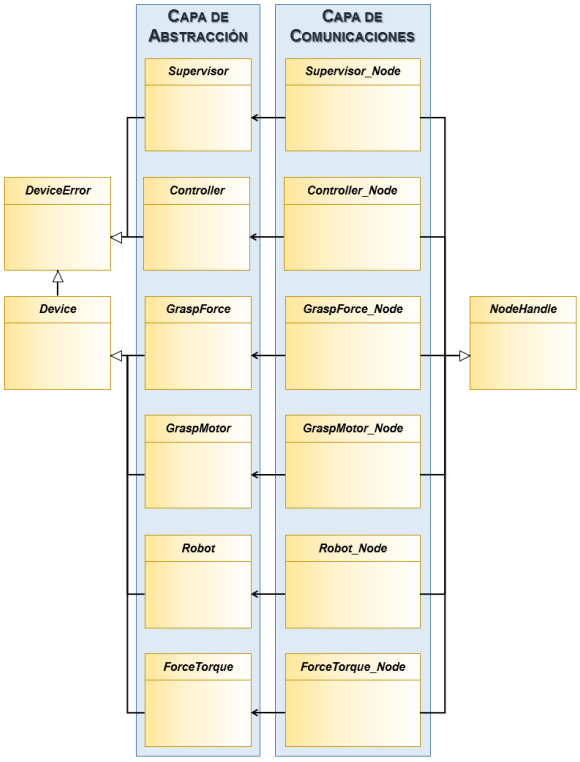


Figura 5: Capa de comunicaciones UML.

ISBN 978-84-15914-12-9

Como puede observarse, para la clase de cada dispositivo se ha extendido una clase asociada con la extensión "_Node". Estas nuevas clases cumplen dos objetivos muy importantes de cara al desarrollo del robot quirúrgico:

1. **Función de encapsulamiento**. La clase de cada dispositivo se ejecuta en un proceso independiente denominado *nodo*. Para ello, estas clases heredan los atributos y métodos que requiere un nodo de una clase padre denominada *NodeHandle*. La ventaja de este encapsulamiento es que permite aislar el trabajo realizado en cada dispositivo y reduce la integración a un problema de comunicación de los parámetros de entrada/salida necesarios para su funcionamiento con el resto del sistema.
2. **Función de comunicación**. Como se ha comentado, cada nodo asociado a un dispositivo debe comunicarse con el resto. De este modo, las clases "_Node" deben tener un protocolo de comunicaciones común que les permita intercambiar información entre ellas.

Aunque en este punto cada dispositivo tiene un nodo asociado capaz de comunicarse, falta un nodo que actúe como nexo de estas comunicaciones. La clase que define este nodo se ha marcado con el nombre de *NodeMessages* a la derecha de la Figura 5, y para que tenga carácter de nodo también debe heredar de la clase *NodeHandle*.

Así, cada vez que el nodo de un dispositivo deba transmitir información siempre pasará a través de *NodeMessages*. Esta forma de comunicación centraliza todos los datos en un único nodo que actúa a modo de "tablón publicitario" para el conjunto de dispositivos del sistema robótico, que consiste en un espacio en la memoria desde donde leer y escribir sus datos.

5 IMPLANTACIÓN DEL SISTEMA E INTEGRACIÓN ROS

El desarrollo de un proyecto de robot quirúrgico teleoperado requiere tomar una serie de decisiones que determinan las capacidades del prototipo final. Por ello, en la primera fase del proyecto el equipo técnico mantuvo reuniones con el grupo de cirujanos para definir las características técnicas que debía tener el sistema.

En base a estas decisiones, en una segunda fase se procedió a seleccionar los dispositivos más adecuados que cumplieran todos los objetivos marcados en estas reuniones y que ya han sido descritos en los apartados 2.1 y 2.2:

- **Brazo manipulador**: Universal Robots UR5 de 6 GDL.
- **Sensor de fuerzas**: Ati Gamma de 6 GDL.
- **Agarre de instrumental**: adaptador para herramientas de Storz gobernado por una placa Maxon EPOS2 y un sensor de fuerzas de 1 GDL OMD-20-FF-600N.
- **PC industrial**: Siemens IPC427D para cada unidad robotizada y consola.
- **Dispositivos hápticos**: Force Dimension Omega. 7 con 7 GDL en posición y fuerzas.
- **Pantallas**: 3D de Panasonic de 26" y táctil Elo 14".
- **Pedales**: conectados a través de una DAQ USB-DUX D.

5.1 INTEGRACIÓN ROS

Tras seleccionar todos los dispositivos que componen el sistema, la tercera fase consiste en la ejecución del desarrollo e implantación del sistema cuyas características se describieron en el apartado 2, de tal forma que la programación de todos los dispositivos ha seguido la metodología propuesta en los apartados 3 y 4.

Las clases relativas a la capa de comunicaciones se han diseñado mediante el software ROS, un sistema de comunicaciones *open source* enfocado a la integración de dispositivos empleados en robótica y que funciona en sistemas operativos Linux (principalmente centrado en las distribuciones de Ubuntu). Algunas ventajas del software ROS son:

- **Procesos independientes**. El nodo principal, denominado *roscore* (equivale a *NodeMessages* en la Figura 5), conecta al resto de nodos independientemente de si se ejecutan en la misma máquina o de forma distribuida. Lo único que debe saber roscore es la dirección IP de la máquina en la que se ejecuta cada nodo.
- **Gestión de mensajes**. El nodo *roscore* gestiona las comunicaciones entre el resto de nodos. La ventaja es que cada nodo puede enviar datos y ellos mismos deciden cuáles quieren recibir del resto de dispositivos. A su vez, los mensajes pueden tener casi cualquier formato, desde una magnitud escalar sencilla hasta complejas estructuras, vectores, matrices…
- **Interrupción del sistema**. El hecho de que un nodo se detenga por cualquier problema no impide que el resto de nodos pueda seguir ejecutándose. Esto permite un mayor control sobre el procedimiento a efectuar ante el fallo de un dispositivo.
- **Trabajo previo**. Con frecuencia, los desarrolladores ya habrán programado un paquete para comunicarse con multitud de dispositivos mediante ROS. Muchos de ellos publican estos

ISBN 978-84-15914-12-9

paquetes para poder ser utilizados en nuevos proyectos y ahorrar tiempo de desarrollo.

Las ventajas de ROS permiten desarrollar el sistema de forma escalonada y paralela entre los distintos equipos de trabajo. Con esto se consigue disminuir el tiempo de desarrollo y de integración, sin más que tomar la decisión del formato en que los mensajes son enviados entre los dispositivos. Así, cada vez que se realizó una integración del sistema completo, se completó en cuestión de horas en lugar de días o incluso semanas.

Una descripción general con todos los mensajes publicados por los dispositivos del robot quirúrgico a través de ROS aparece en la Tabla 1. La columna titulada como "Fuente" indica la procedencia de los mensajes en función de las clases de la capa de comunicaciones. En concreto, la letra X de las fuentes *unit_X* indica un identificador entre 1 y 3 según la unidad robotizada a la que referencia el mensaje. Análogamente, la letra Y de *Haptic_Y* representa si el haptic es el izquierdo o el derecho.

Tabla 1: Lista de mensajes de los dispositivos.

Fuente	**Mensajes**	**Tipo***
unit_X/Robot	Joints	L
	Pose	L
	Velocity	L
	Acceleration	L
	Force	L
	State	L
	Set_Target	E
	Set_Mode	E
unit_X/ForceTorque	Filtered	L
	Raw	L
	Reset	E
unit_X/GraspMotor	Position	L
	Target	E
	Reset	E
unit_X/GraspForce	Filtered	L
	Raw	L
	Reset	E
console/Haptic_Y	Pose	L
	Velocity	L
	Force	L
	Grip	L
	Set_Force	E
console/Pedals	Tools	L
	Camera	L
console/Tactile	Params	L

*Leyenda: E=Escritura ; L=Lectura.

En lo referente a la columna de mensajes, puede observarse que el robot ofrece toda la información a nivel articular con el mensaje *Joints* incluyendo posición, velocidad, corriente y temperatura. Además también se publican mensajes sobre la posición cartesiana (*Pose*), velocidad cartesiana (*Velocity*), aceleración cartesiana (*Acceleration*), fuerza medida en el efector final del brazo (*Force*) y estado del brazo (*State*). Para comandar órdenes de movimiento se emplea el canal de mensajes *Set_Target*, mientras que *Set_Mode* permite cambiar de modo de movimiento con haptics a manipulación manual del brazo y viceversa.

El resto de dispositivos de la unidad robotizada son el sensor de fuerzas (ForceTorque), que devuelve tanto la lectura bruta (*Raw*) como la filtrada de pares gravitatorios (*Filtered*), a la par que permite resetear el filtro (*Reset*). El sensor de agarre (GraspForce) también dispone de estos mismos mensajes, mientras que el actuador de la pinza (GraspMotor) devuelve su posición (*Position*), o bien recibe una orden para moverse (*Target*) o de reseteo de su offset (*Reset*).

Por su parte, en los dispositivos de la consola se encuentran los *Haptic_Y* de los que se puede recoger su estado de posición (*Pose*), velocidad (*Velocity*), fuerza ejercida por sus actuadores (*Force*), y posición del GDL de la pinza (*Grip*). Para poder aplicar una sensación háptica de fuerzas también se permite el envío de una señal de fuerzas a través de *Set_Force*. Por último, los mensajes de Pedals detectan si se desea mover la cámara (*Camera*) o las herramientas (*Tools*), y en el caso de la pantalla táctil se puede leer una serie de parámetros definidos por el usuario gracias al mensaje *Params*.

5.2 SEGURIDAD

El correcto funcionamiento del robot quirúrgico durante la intervención debe estar garantizado en todo momento. Para ello, se ha desarrollado un elemento denominado Supervisor que ya ha sido introducido en el apartado 3. Este elemento se encarga de verificar que todos los dispositivos hardware funcionan correctamente, incluyendo no sólo errores de bajo nivel como la lectura de sensores, comunicaciones o fallos en el hardware, sino también de alto nivel como realizar movimientos sólo cuando el instrumental esté dentro del campo de trabajo, o bien evitar ejercer fuerzas no deseadas debida a movimientos involuntarios del brazo manipulador [14].

El supervisor catalogará los errores que pudiera encontrar según su nivel de severidad. En los casos menos graves modificará el funcionamiento del sistema para seguir realizando la intervención. Sin embargo, si se detectara una situación bajo la cual el robot no pudiera seguir utilizándose, éste cambiará a un estado especial de emergencia. Al entrar en este estado la operación deberá seguir realizándose manualmente, y el robot retirado mediante el siguiente protocolo:

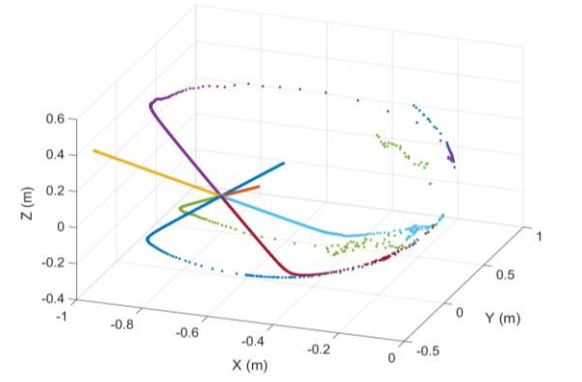
ISBN 978-84-15914-12-9

1. Desacoplar el instrumental quirúrgico del efector final de cada manipulador.
2. Desplazar los brazos para alejarlos del paciente.
3. Desactivar las fijaciones al suelo y mover cada unidad robotizada fuera del campo quirúrgico para facilitar las tareas de los cirujanos y asistentes.

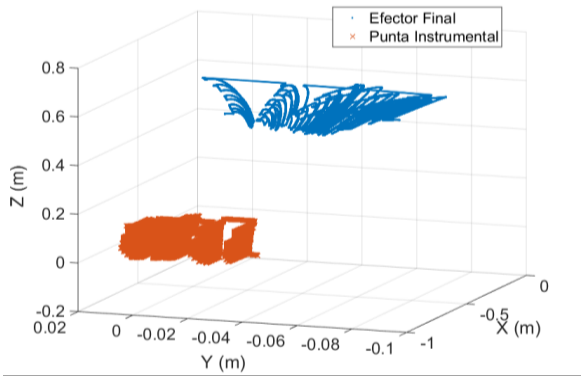
Como norma general, cada unidad robotizada podrá desplazarse en cualquier momento gracias a sus ruedas, pero deberá fijarse al suelo siempre que vaya a utilizarse. Además, todos los brazos manipuladores se encontrarán en un modo especial en el que pueden moverse de forma manual excepto cuando el cirujano tenga el control de los mismos a través de los dispositivos hápticos al pulsar el pedal dead-man.

5.3 EXPERIMENTOS

De entre la multitud de experimentos necesarios para verificar el funcionamiento de un robot quirúrgico, a modo ilustrativo este artículo muestra la validación de los movimientos del instrumental. El objetivo consiste en comprobar que cada herramienta quirúrgica puede posicionarse en cualquier punto del espacio de trabajo o campo visible. Se supondrá que dicho espacio consiste en un volumen cúbico de 15 cm de lado.



(a) Límites del espacio de trabajo



(b) Trayectorias de ambos extremos del instrumental

Figura 6: Captura de los movimientos del robot en el espacio de trabajo visible.

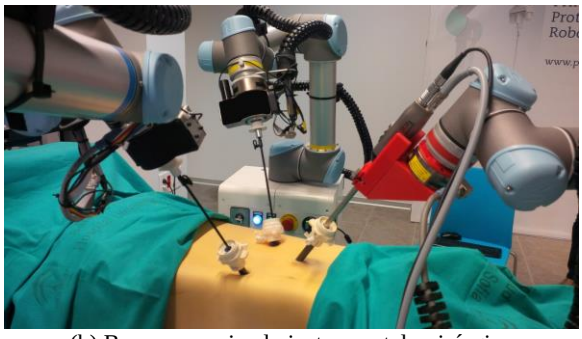
En primer lugar hay que averiguar cuáles son los límites de las coordenadas esféricas (α, β, ρ) respecto del fulcro. Para ello, en la Figura 6 (a) se muestra el movimiento de la punta del instrumental manejado por uno de los brazos manipuladores que va hacia cada esquina del cubo. Aquellos puntos que no siguen la línea están fuera de los límites esféricos (por ejemplo, un valor de ρ<0).

En segundo lugar hay que validar que todos los puntos dentro del espacio de trabajo son alcanzables por el robot. En la Figura 6 (b) puede verse el resultado de desplazar el robot por una secuencia de puntos calculada dentro del espacio de trabajo. En este caso todos los puntos son alcanzables por lo que el espacio de trabajo es válido.

6 CONCLUSIONES

La metodología propuesta en este artículo para el desarrollo de un robot quirúrgico de tres brazos robóticos ha permitido realizar, partiendo de cero, un prototipo funcional en un plazo de tiempo inferior a los tres años, el cual puede verse al completo en la Figura 7. En la imagen superior (a) se observa al usuario del sistema controlando dos de los brazos manipuladores a través de los dispositivos hápticos ubicados en la consola. Las herramientas de los 3 brazos pueden apreciarse con mayor detalle en la imagen inferior de la Figura 7 (b), incluyendo tanto el instrumental quirúrgico como la cámara estereoscópica.

(a) Manejo del sistema mediante la consola



(b) Brazos manejando instrumental quirúrgico

Figura 7: Integración del robot quirúrgico.

ISBN 978-84-15914-12-9

En este plazo de tiempo se incluye el período inicial de toma de decisiones sobre las capacidades técnicas que debía ofrecer el sistema robótico al cirujano, así como la selección de dispositivos que ya se ha comentado en el apartado 5. La toma de decisiones y reuniones conjuntas entre el equipo de cirujanos y el técnico ha resultado ser uno de los mayores aportes al proyecto, y sin el cual el proceso habría sido mucho más complejo.

El uso de una herramienta como ROS para la integración de dispositivos y sus comunicaciones ha facilitado las tareas de integración entre los distintos equipos técnicos, ubicados en distintas provincias del territorio nacional. Esto ha permitido paralelizar el trabajo, aumentando el rendimiento y disminuyendo los plazos de desarrollo al reducir las tareas de cada grupo.

En conclusión, la definición de una arquitectura UML apropiada para adaptar la programación de los dispositivos al software ROS ha hecho posible el desarrollo de un sistema tan complejo como es un robot quirúrgico teleoperado con varios grupos de trabajo ubicados en distintos puntos de la geografía española.

Agradecimientos

El grupo técnico formado por Tecnalia y la Universidad de Málaga desea agradecer a IMIBIC, el Hospital Reina Sofía y la Universidad de Córdoba tanto su confianza como el apoyo material y humano que ha permitido desarrollar este proyecto de compra pública precomercial soportado por el Ministerio de Ciencia e Innovación. También se quiere destacar la ayuda y asesoramiento prestados por Juan Cabello y Carlos J. Pérez para el diseño de la arquitectura UML

Referencias

[1] O'Kane, J.M., (2014) "A Gentle Introduction to ROS", *University of South Carolina*, ISBN 978-14-92143-23-9, Columbia.

[2] Qian, W. et al, (2014) "Manipulation Task Simulation using ROS and Gazebo", *IEEE Int. Conf. Robotics and Biomimetics*, pp- 2594-2598, Indonesia.

[3] DeMarco, K., West, M.E., Collins, T.R., (2011) "An implementation of ROS on the Yellowfin autonomous underwater vehicle", *IEEE Int. Conf. OCEANS*, pp- 1-7, Waikoloa.

[4] Speers, A., et al, (2013) "Lightweight tablet devices for command and control of ROS-enabled robots", *IEEE Int. Conf. Advanced Robotics (ICAR)*, pp- 1-6, Montevideo.

[5] Barros, J.O., Santos, V.M., Silva, F., (2015) "Bimanual Haptics for Humanoid Robot Teleoperation using ROS and V-REP", *IEEE Int. Conf. Autonomous Robot Systems and Competitions*, pp- 174-179, Vila-Real.

[6] Rosa, S., Russo, L.O., Bona, B., (2014) "Towards A ROS-Based Autonomous Cloud Robotics Platform for Data Center Monitoring", *IEEE Emerging Technology and Factory Automation (ETFA)*, pp- 1-18, Barcelona.

[7] Zaman, S., et al, (2013) "An Integrated Model-Based Diagnosis and Repair Architecture for ROS-Based Robot Systems", *IEEE Robotics and Automation*, pp- 482-489, Karlsruhe.

[8] Beasley, R.A., (2012) "Medical Robots: Current Systems and Research Directions", Journal of Robotics, Hindawi Publishing Corporation, Vol. 2012, doi:10.1155/2012/401613.

[9] Bauzano, E., Estebanez, B., Garcia-Morales, I., Muñoz, V.F., (2015) "Planning Automatic Surgical Tasks for a Robot Assistant", *Motion and Operation Planning of Robotic Systems*, Springer Int. Publishing Switzerland, DOI 10.1007/978-3-319-14705-5_7.

[10] Perez-del-Pulgar, C.J., Muñoz, V.F., (2014) "Control Scheme with Tissue Interaction Detection for a Single Port Access Surgery Robotic Platform", *Int. Conf. Biomedical Robotics and Biomechatronics (BIOROB)*, pp. 713-718, Sao Paulo, Brazil.

[11] Naerum, E., Hannaford, B., (2009) "Global Transparency Analysis of the Lawrence Teleoperator Architecture", *Int. Conf. Robotics and Automation*, pp. 4344-4349, Kobe, Japan.

[12] Christiansson, G.A., (2007) "Hard Master, Soft Slave Haptic Teleoperation", *Master Thesis on Physics Engineering, Chalmers University of Technology*,ISBN 978-90-8559-307-2, Sweden.

[13] Fjellin, J.E., (2013) "Design of a bilateral master-slave system with haptic feedback for ultrasound examinations", *Master Thesis, Faculty of Mathematics and Natural Sciences, University of Oslo, Norway*.

[14] Bauzano, E., Muñoz, V.F., Rivas-Blanco, I., (2011) "Sistema Tolerante a fallos para el control con realimentación de fuerzas en un robot asistente", *XXXII Jornadas de Automática*, Sevilla.