

EVALUACIÓN DE UN ENTORNO DE TELEOPERACIÓN CON ROS

David Vargas Frutos (david.vargas@upm.es), Juan Carlos Ramos Martínez (juancarlos.ramos@upm.es), José Luis Samper Escudero (joseluis.samper@upm.es), Miguel A. Sánchez-Urán González (miguelangel.sanchezuran@upm.es), Manuel Ferre Pérez (manuel.ferre@upm.es)

Centro de Automática y Robótica (CAR) UPM-CSIC, Universidad Politécnica de Madrid.

Resumen

En este artículo se resume el desarrollo de un sistema de teleoperación para el Grips de Kraft Tele-Robotics. La integración con ROS permite el uso de varias herramientas de análisis de datos, visualización, simulación, e interacción con múltiples sensores. Como instrumento maestro del sistema se propone el dispositivo háptico Phantom Omni de SensAble, para el que se ha desarrollado una reflexión de fuerzas.

Palabras clave: Robótica, teleoperación, háptico, realimentación de fuerzas.

1. INTRODUCCIÓN

El hombre a lo largo de la historia ha utilizado herramientas para aumentar el alcance de su capacidad de manipulación. Esto desembocó en lo que se conoce como sistemas de teleoperación maestro-esclavo, en los que el manipulador esclavo reproduce con fidelidad los movimientos que se ejecutan en el dispositivo maestro [?].

Dentro de la teleoperación, el sistema de control más básico es el unilateral, en el que el maestro simplemente genera las señales de referencia, y sea en posición o en velocidad, y el esclavo reproduce esas señales. El control bilateral, por otro lado, es una evolución de éste, en el que se genera una realimentación de información, normalmente en forma de fuerza aplicada sobre la mano del operador. Un esquema de este tipo de sistema puede observarse en la Figura 1. Los problemas asociados a este tipo de control son los debidos a los retardos del esclavo, tanto en su movimiento, como en la realimentación de fuerzas que éste debería generar.

La realimentación de información al operador se hace de diferentes formas. En primer lugar, de forma visual mediante el envío de imágenes de vídeo o simulaciones gráficas. El sentido de la vista permite conocer de qué elementos se compone la zona de trabajo, así como sus características de dimensión, posición, etc. Por otro lado, el sentido del tacto informa sobre los esfuerzos que se producen

en la manipulación de los objetos. Sistemas hápticos como el aquí descrito, ayudan a la inmersión en el entorno mediante una realimentación en fuerza. Por último, la realimentación auditiva no suele ser utilizada y se limita a sonidos de alarmas o a los producidos por las propias máquinas.

Cuando se diseña un sistema de teleoperación, una de las cosas que se debe tener en cuenta es el tipo de señales que se quieren transmitir entre el dispositivo maestro y el esclavo. Los dos esquemas de teleoperación bilateral más comunes son el esquema posición-posición y fuerza-posición [?]. El esquema de fuerza-posición, al igual que el de posición-posición, tiene un bucle cerrado de posición en el esclavo, tomando como referencia la posición en el maestro. Sin embargo, precisa de un sensor que sirva para obtener la fuerza/par que aplica el esclavo sobre su entorno o viceversa. Esta información es enviada al maestro y éste refleja esa lectura en bucle abierto de fuerza sobre el operador. Un amplio estudio sobre el control bilateral se realiza en [?], donde se analizan esquemas clásicos de control bilateral y esquemas con retardo temporal.

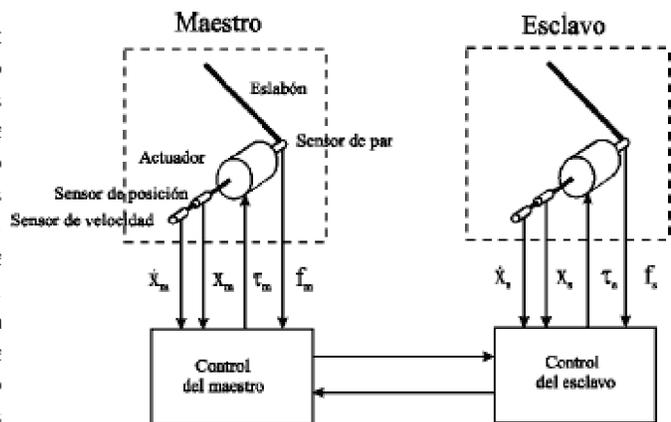


Figura 1: Esquema general de control bilateral. El maestro transmite al esclavo información acerca de distintos parámetros, como posición y velocidad. El esclavo realimenta al maestro con información que el operador percibe en forma de fuerzas.

También son posibles sistemas de control bilate-

ral en los que no existan sensores de fuerza. Para ello, es necesario realizar una estimación de los estados y fuerzas del sistema, y después usar estas estimaciones en los controladores. Un ejemplo de estos sistemas es desarrollado en [?]. En robots hidráulicos, como es este caso, puede ser calculada una aproximación del par de reacción que existe en cada articulación a partir de la diferencia de presión existente en los dos cilindros hidráulicos.

En relación a las interfaces de control, se puede hablar de interfaces directas, en las que el operador controla directamente el manipulador, o interfaces multimodales, que toman información de varios sensores, la combinan, y el operador la puede percibir a través de varios sentidos, lo que le permite un mejor control del manipulador. Por último, las interfaces para control supervisado incluyen todo tipo de dispositivos para aumentar el grado de inmersión en el entorno, dispositivos que pueden ser unas gafas de realidad virtual o un dispositivo háptico. El uso de estas interfaces crece cada vez más, y es en el aspecto háptico en lo que se centra este artículo.

El estudio aquí mostrado presenta la teleoperación de un brazo robótico de 6 grados de libertad mediante un dispositivo háptico de también 6 grados de libertad, con la posibilidad de realimentar al operador de forma proporcional a las fuerzas que actúen sobre el manipulador.

En la sección 2 se presentan algunas nociones sobre ROS y su uso en teleoperación. En la sección número 3 se procede a describir los equipos que conforman el entorno de teleoperación, en la sección número 4 se comentan aspectos sobre el tipo de las arquitecturas de teleoperación establecidas y sobre la realimentación de fuerzas generada en el dispositivo maestro, para finalmente, en la sección 5 concluir con un análisis de los resultados obtenidos.

2. ROS APLICADO A SISTEMAS TELEOPERADOS

2.1. ROS

ROS (Robotic Operating System) [?] es una plataforma de desarrollo *open source* para sistemas robóticos. Se compone por un conjunto de librerías, herramientas y estándares que ayudan al desarrollo de software robótico. Provee elementos tales como controladores y algoritmos, como por ejemplo algoritmos de navegación o de teleoperación. Es un sistema basado en un concepto modular y distribuido. Los distintos módulos se llaman nodos y se comunican entre sí mediante mensajes transmitidos a través de *topics* o servicios, si-

guiendo un modelo de publicador/suscriptor. Este modelo permite un envío de información ágil y dinámico.

2.2. ROS EN TELEOPERACIÓN

ROS tiene compatibilidad con un gran número de robots de forma nativa, al igual que con un gran número de sensores y cámaras. Para ello, proporciona varios paquetes que pueden facilitar tareas como la generación correcta de trayectorias, o planificación de agarres. También tiene integración con dos potentes herramientas de simulación y visualización como son Gazebo y RViz, lo que hace de ROS una plataforma muy completa para el desarrollo de tareas de teleoperación.

Los controladores utilizados para teleoperación deben estar diseñados de tal forma que sean robustos ante retardos, saturación en los actuadores y otras no linealidades. ROS es capaz de proporcionar el estado del maestro y del esclavo en cada ciclo de ejecución, además de monitorizar parámetros importantes del modelo. También permite calcular transformaciones de coordenadas, de forma que se puede obtener la posición en el espacio de cualquier componente referenciado a cualquiera de los sistemas de coordenadas definidos. Esto permite realizar una operación sobre el esclavo de forma instantánea a la que se realiza sobre el maestro. Además, el propio framework integra y gestiona eficazmente distintas interfaces con los principales protocolos de comunicación como CAN-BUS, TCP/IP y UDP.

3. DESCRIPCIÓN DE LOS EQUIPOS

El sistema real en el que se realizan los experimentos está formado por los siguientes componentes:

- Telemanipulador hidráulico Grips de Kraft TeleRobotics.
- Controlador en tiempo real de ©National Instruments NI-PXIe-8108.
- PC corriendo Labview 2011 conectado con el PXI.
- Sensor de Fuerza/Par ATI SI-130-10.
- Dispositivo háptico Phantom Omni de SensAble.

3.1. TELEMANIPULADOR GRIPS DE KRAFT TELERBOTICS

El brazo robótico utilizado es el modelo *Grips* del fabricante *Kraft TeleRobotics* mostrado en la Figura 2. Se trata de un manipulador hidráulico de

6 grados de libertad cuya operación se lleva a cabo bajo un sistema maestro-esclavo mediante el que se pueden realizar tareas con movimientos y velocidad humanos.



Figura 2: Brazo robótico hidráulico Grips de Kraft TeleRobotics utilizado como esclavo en el entorno de teleoperación bajo ROS.

Este sistema cuenta con la capacidad de reflejar las fuerzas que actúan sobre el brazo manipulador, de tal forma que el operador es realimentado con información de fuerzas y recibe así una mejor percepción del entorno de operación. Esta cualidad hace posible realizar tareas de un mayor grado de complejidad, como se ve en [?].

Como efectores finales se han usado dos pinzas diferentes. En primer lugar, una simple de dos dedos de *Kraft TeleRobotics*. Como segunda pinza, se usa una de tres dedos de *Robotiq*, que además dispone de sensores de fuerza en cada uno de los dedos. Las posibilidades de agarre de ésta última son variadas, ya que permite realizar agarre en tijera, con dos dedos, o adaptar los tres dedos a la forma del objeto a manipular. La decisión de usar como segunda pinza una de tres dedos es debido al equilibrio que tiene entre rendimiento y flexibilidad.

3.2. DISPOSITIVO HÁPTICO PHANTOM OMNI DE SENSABLE



Figura 3: Dispositivo háptico Phantom Omni de SensAble utilizado como maestro en el entorno de teleoperación bajo ROS.

Como recurso de control maestro se ha utilizado el *Phantom Omni* del fabricante *SensAble*, mostrado en la Figura 3.

Consiste en un dispositivo háptico de 6 grados de libertad, cuyas 3 primeras articulaciones son capaces de generar reflexión de fuerzas y que permite la interacción con entornos virtuales complejos. Puede medir de forma precisa la posición espacial tridimensional (X, Y, Z) y la orientación (giro, inclinación y dirección). Puede trabajar en ROS gracias al desarrollo de paquetes que hacen compatible su integración. En [?] se recoge un estudio sobre su análisis cinemático y manipulabilidad.

3.3. MODELO SIMULADO

Para el modelo simulado se ha utilizado Gazebo [?] como simulador debido a su potencia y a su facilidad de integración con ROS. Gracias al potente motor de físicas que posee Gazebo, se dispone de la capacidad de simular de forma precisa y totalmente realista el comportamiento de diversos entornos y escenarios bajo condiciones muy similares a la realidad. Debido a la complejidad de la puesta en marcha del sistema hidráulico del esclavo, así como a la peligrosidad de una operación imprecisa de un brazo robótico de tales dimensiones, el uso de Gazebo como base de testeo en simulación de los algoritmos desarrollados se torna fundamental.

Para dicha simulación se ha podido disponer de un modelo virtual del brazo robótico proporcionado por el fabricante. Se dispone también de los modelos de ambas pinzas facilitados igualmente por los fabricantes, con la necesidad de llevar a cabo la integración con el modelo global. Esto da lugar a un modelo virtual con un alto nivel de detalle y semejanza con el real. El modelo virtual utilizado se puede apreciar en la ilustración mostrada en la Figura 4.

Los modelos están formados por un conjunto de ficheros URDF. El acrónimo URDF proviene de *Unified Robot Description Format*, y es la forma que ROS tiene de representar modelos de robots. Lo que se hace generalmente es desglosar el modelo en ficheros XACRO primero, para después convertirlo a URDF. Cada fichero representa una parte del modelo general a través de un conjunto de *links* y de *joints*, por ejemplo el sensor de fuerza, o una de las pinzas.

Para las pruebas en simulación se ha diseñado un entorno similar al del laboratorio real, en el que se han añadido diferentes objetos para hacer pruebas de agarre, como son una esfera y un disco. Estos objetos servirán para comprobar las ventajas de cada una de las pinzas. La pinza de tres dedos, debido a su configuración, se adapta mejor

al agarre de la esfera, mientras que por otro lado, el disco es agarrado de forma más sencilla por el de dos dedos.

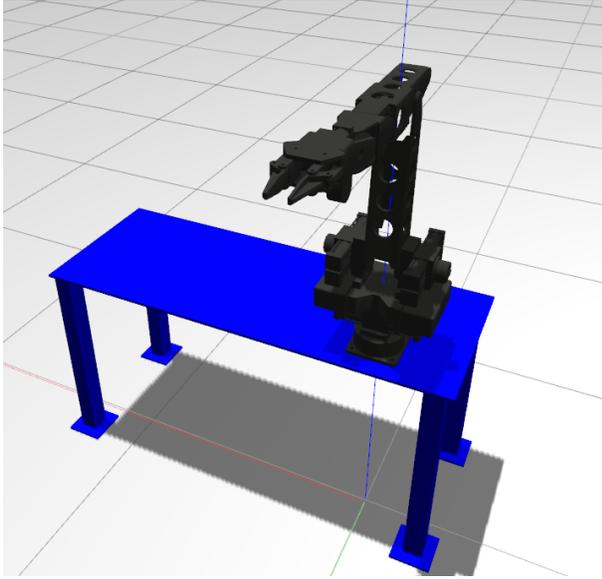


Figura 4: Simulación en Gazebo del entorno de manipulación. En el escenario real del laboratorio donde se ha llevado a cabo este trabajo, el brazo robótico Grips se encuentra situado sobre una estructura similar a una mesa, en la simulación se refleja fielmente dicha configuración.

En cuanto al dispositivo maestro, también se dispone de una interfaz gráfica para visualizar diferentes parámetros del mismo. Con los ficheros adecuados que especifican el modelo virtual del Phantom Omni, desde la herramienta RViz propia de ROS se puede monitorizar el dispositivo y la posición de cada una de sus articulaciones. Esto es útil para apreciar de una forma visual la información de posición y orientación de cada una de las articulaciones que se está transmitiendo por los *topics* correspondientes, en especial la referida al extremo final del dispositivo háptico. En la Figura 5 se puede apreciar la representación virtual del dispositivo háptico y el gizmo que indica de forma visual la posición y orientación de su extremo final.

4. ARQUITECTURAS DE TELEOPERACIÓN E INTERACCIÓN HÁPTICA

Uno de los factores que más influye a la hora de realizar este sistema teleoperado es el hecho de que el dispositivo háptico maestro no tiene la misma configuración cinemática que el brazo robótico esclavo, lo que hace necesario que se tenga que realizar una adaptación entre dispositivos.

Para ubicar el efector final del esclavo dentro del espacio de trabajo del brazo robótico en una posición semejante a la posición y orientación del efector final del maestro es necesario realizar una serie de transformaciones.

En cuanto a la posición, se realiza un proceso de escalado de las coordenadas cartesianas de la posición del efector final del esclavo aplicando un factor de conversión para cada eje. En este proceso de conversión se tienen en cuenta los alcances máximos de los efectores finales de maestro y esclavo en cada eje. Conociendo los límites de movimiento de cada caso se puede aplicar una conversión proporcional entre los distintos espacios de trabajo.

En cuanto a la rotación, se reformula la orientación de la ubicación del efector final del esclavo. Debido a que maestro y esclavo tienen configuraciones cinemáticas distintas, la posición natural o de reposo de ambos es distinta, y sus efectores finales tienen orientaciones respecto a sus sistemas de coordenadas locales prácticamente opuestas. La necesidad de realizar transformaciones entre las orientaciones de los efectores finales viene dada por la imposibilidad de alcanzar determinadas posiciones del esclavo manteniendo las orientaciones indicadas por el maestro.

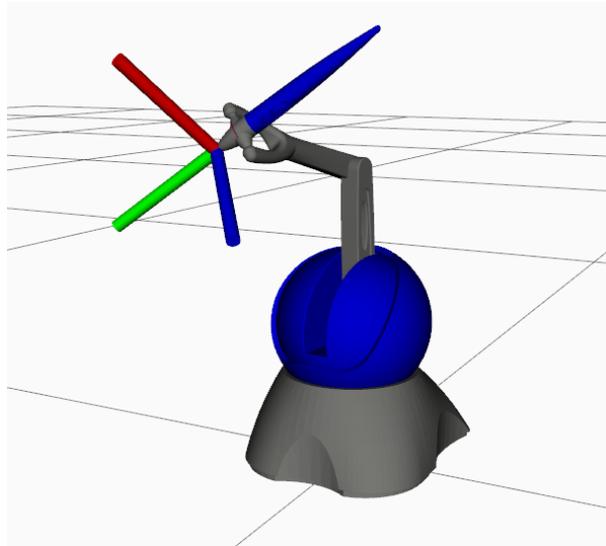


Figura 5: Modelo virtual del Phantom Omni

Con el objetivo de ubicar el efector final del esclavo en una posición determinada, se calcula la cinemática inversa del brazo robótico esclavo. Con los parámetros de la cinemática inversa es posible obtener la configuración angular de cada una de las articulaciones del brazo robótico para ubicar el efector final en tales posición y orientación.

En este punto cabe destacar que las posiciones que el esclavo es capaz de alcanzar tienen un determi-

nado número de orientaciones posibles. Por ello, si se indica al esclavo que se sitúe en una posición con una orientación que no es capaz de reproducir, los cálculos de cinemática inversa para esa posición en concreto devolverán un resultado negativo y se considerará que la posición pedida es inalcanzable. Como resultado, aunque las coordenadas cartesianas de la posición pedida al esclavo fueran posibles de alcanzar, si la orientación indicada no es la adecuada, el brazo robótico no se moverá a tal posición.

Una posible solución a este problema requiere realizar un cálculo exhaustivo de las rotaciones viables del efector final del esclavo para cada una de las incontables posiciones que el brazo robótico es capaz de alcanzar. Ante tal situación, se ha optado por una solución más inmediata, que consiste en relajar la tolerancia en términos de la orientación del efector final pedida al esclavo. De esta forma, se tendrán mayores probabilidades de que la cinemática inversa del brazo robótico pueda devolver resultados positivos para alcanzar la posición indicada por el maestro si se otorga menos prioridad a los requisitos de exactitud en la orientación.

Existen diversas librerías y métodos de cálculo de cinemáticas inversas [?]. Para este caso, se ha optado por calcular la cinemática inversa del esclavo usando la librería *TRAC-IK* [?]. Desarrollada recientemente, comprende un conjunto de paquetes integrados con ROS que proporcionan soluciones de cinemática inversa más eficientes que las calculadas por métodos habituales como KDL, basados en el cálculo de Jacobianas inversas. Tales algoritmos de convergencia aplican el método de Newton, que no siempre converge a una solución cuando se imponen límites articulares, algo común en muchas plataformas robóticas. *TRAC-IK* ejecuta de forma concurrente dos implementaciones de cinemática inversa. Una de ellas es una extensión del método de Newton que detecta y mitiga, mediante saltos aleatorios, los mínimos locales debido a los límites de las articulaciones. La otra consiste en un enfoque de optimización no lineal SQP (*Sequential Quadratic Programming*) que utiliza métodos cuasi-Newtonianos que manejan mejor los límites de las articulaciones. La búsqueda de la cinemática inversa devuelve una solución inmediatamente cuando uno de estos algoritmos converge.

En definitiva, esta implementación alternativa de cinemática inversa encuentra soluciones mucho más a menudo que KDL, es más rápida en promedio que KDL para cadenas de manipulación típicas y, si se desea, permite aplicar tolerancias en cada dimensión cartesiana, incrementando aún más la velocidad de convergencia. Por lo tanto, el uso de esta herramienta supone una solución perfecta pa-

ra el trabajo llevado a cabo.

Los procesos computacionales que se ejecutan son varios. Comprenden el cálculo de la posición y orientación del extremo del maestro en función del sistema de coordenadas definido como original; la adaptación de esta ubicación al espacio de trabajo del esclavo; el cálculo de la cinemática inversa con la que definir la configuración de las articulaciones del brazo robótico para alcanzar la ubicación pretendida y decidir si es viable o no; y el movimiento físico de las articulaciones a las posiciones decididas.

Todo este coste de procesamientos mencionados se ve reflejado en forma de retardos. La precisión no supone un problema por parte del maestro, pero el proceso de transformaciones que hay que realizar para generar coordenadas de ubicaciones adecuadas para el esclavo, así como los cálculos de cinemática inversa para llevar al esclavo a dichas ubicaciones, puede degradar esa precisión. Se han configurado distintas arquitecturas para experimentar el rendimiento de este sistema de teleoperación.

4.1. ARQUITECTURA CENTRALIZADA

La experimentación inicial consiste en realizar la teleoperación bajo una arquitectura centralizada, es decir, consiste en la ejecución de todo el software en un único ordenador. Tanto la adquisición de los datos de posiciones del maestro, la conversión a posiciones correspondientes en el espacio de trabajo del esclavo, el cálculo de las cinemáticas inversas, como el renderizado de la simulación del esclavo en el entorno virtual, se llevan a cabo en la misma máquina. El equipo utilizado para este experimento dispone de un procesador Intel Core i7-6700 a 3.40 GHz de frecuencia.

Los cálculos de cinemática inversa y el renderizado de la simulación son los procesos más costosos de todos los que se están llevando a cabo durante la ejecución de teleoperación. Ejecutando ambos en la misma máquina, se calcula el coste de cómputo de la cinemática inversa. El tiempo promedio de ejecución obtenido no supera los 5 milisegundos.

4.2. ARQUITECTURA DISTRIBUIDA

Se establece una arquitectura distribuida del sistema de teleoperación buscando una mayor fluidez de ejecución. En este caso se utilizan máquinas distintas para ejecutar la simulación por un lado y los cálculos de cinemática inversa y procesos de adquisición de datos por otro. Ambas máquinas pertenecen a la misma red de comunicaciones de área local. Se trabajo con el mismo ordenador uti-

lizado en el caso de la arquitectura centralizada, y se añade un ordenador portátil que dispone de un procesador Intel Core i7-7700HQ a 2.80 GHz de frecuencia

El hecho de que ROS esté basado en un funcionamiento distribuido hace sencillo la configuración de una arquitectura de este tipo. Es preciso definir convenientemente las variables de entorno *ROS_MASTER_URI* y *ROS_IP* en cada una de las máquinas de la arquitectura para posibilitar la comunicación entre ellas dentro de la red.

A pesar de separar bajo esta configuración los procesos de computación de la cinemática inversa y del renderizado de la simulación, el tiempo promedio del coste de cómputo de la cinemática inversa no varía de forma destacable respecto a la ejecución bajo arquitectura centralizada.

De esta forma, tanto para arquitectura centralizada como para distribuida la teleoperación se realiza de forma prácticamente similar. En ambos casos se efectúa de forma fluida. Tal y como se puede apreciar en la Figura 6, la coordenada *X* (indicadas en color rojo) e *Y* (indicadas en color verde) de las posiciones de los efectores finales del maestro (representadas como líneas continuas) y del esclavo (representadas como líneas discontinuas) se encuentran reescaladas en cada instante, de tal forma que la posición indicada por el maestro se transforma inmediatamente en una posición apropiada y acorde al espacio de trabajo del esclavo.

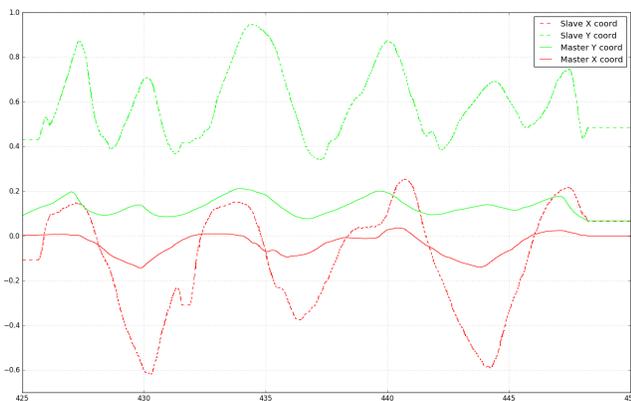


Figura 6: Gráfica de posicionamiento de los efectores finales de maestro y esclavo en un intervalo de 25 segundos. Se puede apreciar cómo cada ubicación del maestro es convertida de forma proporcional en una ubicación del esclavo. Los cálculos de cinemática inversa no introducen retardos apreciables en la operación del sistema maestro-esclavo, el esclavo se mueve de forma prácticamente inmediata a como lo hace el maestro.

4.3. ARQUITECTURA CON LÍNEAS DE TIEMPO REAL

Para realizar la teleoperación sobre hardware real y no sobre simulación se deben introducir elementos que proporcionen un funcionamiento en tiempo real. De esta forma, se sustituye el entorno virtual por el brazo robótico de *Kraft*, así como la comunicación mediante mensajes de ROS por un equipo de National Instruments orientado exclusivamente a trabajar bajo condiciones de tiempo real.

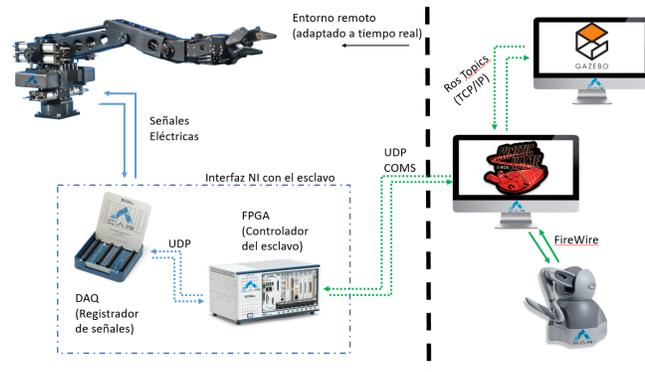


Figura 7: Arquitectura de teleoperación en tiempo real. Mientras la simulación del entorno virtual y la comunicación con el mismo se realiza bajo ROS, la operación del esclavo real requiere un equipamiento hardware adaptado a trabajar en tiempo real. En ambos casos, la adquisición de la información de las ubicaciones del efector final del maestro para realizar la tarea de control del esclavo se mantiene en ROS.

El sistema de teleoperación mencionado, mostrado en la Figura 7, funciona como prosigue: el programa que se ha desarrollado en ROS capta la posición final del *stylus* del *Phantom*, se calcula la cinemática inversa del esclavo para alcanzar la posición referida por el maestro, y el estado de cada una de las articulaciones que debería tener el *Kraft* se transmiten al PXI. Para este tipo de transmisión existen diversas soluciones, como es el caso del protocolo de comunicación BTP orientado a tareas de teleoperación desarrollado en [?]. Para este caso se utiliza UDP como protocolo de comunicaciones para transmitir las posiciones articulares. Por su parte, en el PXI está cargado un programa de *LabView* capaz de leer el estado de cada una de las articulaciones mandadas por UDP y enviárselas al *Kraft*, ejecutándose así el movimiento del esclavo.

El sistema de teleoperación queda totalmente completado con la reflexión al maestro u operador de las fuerzas ejecidas sobre el dispositivo esclavo mediante el entorno del manipulación.

4.4. REALIMENTACIÓN DE FUERZAS

La realimentación de fuerzas que se percibe en el dispositivo maestro permite detectar cuándo el brazo robótico colisiona o agarra un objeto, y cuándo actúan pares o fuerzas sobre sus articulaciones. Para la medición de los valores se dispone de un sensor de fuerza en el efector final del brazo robótico, al igual que de pequeños sensores en cada uno de los dedos de la pinza de *Robotiq*. Estos últimos valores de fuerza obtenidos al agarrar un objeto son enviados mediante terminal, al no disponer el dispositivo maestro de una interfaz sobre la que reproducirlos.

Como se recoge en [?], las fuerzas podrían ser estimadas partiendo de la ecuación que modela el comportamiento dinámico de un manipulador:

$$\tau_m = H(q)\ddot{q} + C(q, \dot{q}) + \tau_g(q) + \tau_f(\dot{q}) + \tau_{ext} \quad (1)$$

Donde:

- τ_m es el vector de los torques ejercidos en cada articulación.
- $H(q)$ es la matriz de inercia del robot.
- $C(q, \dot{q})$ es el vector de fuerzas centrípetas y coriolis.
- τ_g es el vector de fuerzas de gravedad.
- τ_f es el vector de las fuerzas de fricción.
- τ_{ext} es el vector de torques externos aplicados en cada articulación.

Reordenando y entrando en juego la Jacobiana, se puede hacer una estimación de las fuerzas de la siguiente forma:

$$T_{ext} = J^T (\tau_m - H(q)\ddot{q} + C(q, \dot{q})\dot{q} + \tau_g(q) + \tau_f(\dot{q})) \quad (2)$$

donde J es la matriz jacobiana del robot.

En cuanto al tiempo de muestreo, como se demuestra en [?], se tiene un valor de 1kHz, el recomendado para el tipo de teleoperación en la que es necesaria realimentación de fuerzas.

5. CONCLUSIONES Y TRABAJOS FUTUROS

Se ha logrado desarrollar una plataforma completa de teleoperación para el modelo *Grips* de *Kraft TeleRobotics* que permite simular el comportamiento de forma realista de todos los elementos. Gracias a esto, se puede ensayar antes de proceder con el robot real. Por otro lado, permite teleoperar de forma precisa, percibiendo además una realimentación de fuerzas a través del dispositivo háptico *Phantom Omni* de *SensAble*, lo que mejora el nivel de inmersión.

El software de ROS está orientado a funcionar como un sistema distribuido. Por este motivo, un entorno de nodos de ejecución con relativamente baja carga computacional que se ejecute en un equipo lo suficientemente moderno y potente no presenta diferencias de comportamiento entre ejecuciones bajo arquitectura centralizada y arquitectura distribuida.

A pesar de ello, a la hora de operar un esclavo hardware y no virtual, es recomendable prescindir de este framework, fundamentalmente porque está diseñado para funcionar bajo sistemas de Canónico y derivados, cuyos kernels no están comúnmente orientados a satisfacer los requisitos de tiempo real. Sustituir ROS por líneas de tiempo real configuradas mediante equipamiento orientado a trabajar en tiempo real permite establecer un entorno de teleoperación robótica con una adecuada reflexión de fuerzas.

En cuanto a los aspectos en los que seguir trabajando, una mejor realimentación de fuerzas es posible, al igual que una integración de todo el conjunto con un sistema de visión eficaz, por ejemplo formado por varias cámaras 3D como podrían ser los dispositivos *Kinect* de *Microsoft*.

Se propone también el desarrollo de un entorno bimanual en el que dos *kraft* pudieran colaborar para realizar tareas de manipulación más sofisticadas, como pudiera ser dar soporte con uno de ellos, y operar con el restante.

Agradecimientos

Este trabajo ha sido patrocinado por la Universidad Politécnica de Madrid. El material ha sido proporcionado por el Centro de Automática y Robótica, centro mixto de la Universidad Politécnica de Madrid y del Consejo Superior de Investigaciones Científicas.

El desarrollo de este trabajo ha sido parcialmente soportado por el proyecto ExoFlex (DPI2015-68842-R), financiado por MINECO.

Referencias

- [Ferre07] Ferre, M., Aracil, R., Balaguer, C., Buss, M., and Melchiorri, C. (Eds.). (2007). Advances in telerobotics, *Heidelberg: Springer*, Volume: 31.
- [Lawrence93] D.A. Lawrence, (1993) Stability and transparency in bilateral teleoperation, *IEEE Transactions on Robotics and Automation*, Volume: 9, Issue: 5, Oct 1993.
- [Torres07] Bogado Torres, Juan Manuel, (2007) Control Bilateral de Robots Teleoperados

por Convergencia de Estados, Tesis Doctoral Robótica e Informática Industrial, Universidad Politécnica de Madrid.

- [Daly09] John M. Daly, David W. L. Wang, (2009) Bilateral Teleoperation Using Unknown Input Observers for Force Estimation, *American Control Conference, ACC '09*.
- [Quigley09] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng., (2009) ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.
- [Qian14] Qian, W., Xia, Z., Xiong, J., Gan, Y., Guo, Y., Weng, S., and Zhang, J. (2014, December). Manipulation task simulation using ros and gazebo. In *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on* (pp. 2594-2598). IEEE.
- [Silva09] Alejandro Jarillo Silva, Omar A. Domínguez Ramirez, Vicente Parra Vega, Jesus P. Ordaz Oliver, (2009) PHANToM OMNI Haptic Device: Kinematic and Manipulability, *Electronics, Robotics and Automotive Mechanics Conference, CERMA '09*.
- [Acero16] Enrique del Sol Acero, (2016) Force Estimation for Teleoperating Industrial Robots, *Automation and Control Trends*, Capítulo 3.
- [Dalvand14b] Mohsen Moradi Dalvand, Saeid Nahavandi, (2014) Improvements in teleoperation of industrial robots without low-level access, *Systems, Man and Cybernetics (SMC)*.
- [Beeson15] Beeson, P., and Ames, B. (2015, November). TRAC-IK: An open-source library for improved solving of generic inverse kinematics. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on* (pp. 928-935). IEEE.
- [Wirz09] Raúl Wirz, Raul Marin, Manuel Ferre, Jorge Barrio, José M. Claver, and Javier Ortego, (2009) Bidirectional Transport Protocol for Teleoperated Robots, *IEEE Transactions on Industrial Electronics*, Volume. 56, NO. 9, Sep 2009.
- [Díaz06] Iñaki Díaz, Josune Hernantes, Ignacio Mansa, Alberto Lozano, Diego Borro, Jorge Juan Gil, Emilio Sánchez, (2006) Influence of multisensory feedback on haptic accessibility tasks, *Virtual Reality*, Volume 10, Issue: 1, May 2006.
- [Robkani14] Nizar Rokbani, Alicia Casals, Adel M. Alimi, (2014) IK-FA, a New Heuristic Inverse Kinematics Solver Using Firefly Algorithm, *Computational Intelligence Applications in Modeling and Control*, on (pp 369-395).