

CAPÍTULO X

INTERACCIÓN HUMANO-ROBOT A TRAVÉS DE INTERFACES EN LA NUBE

J. G. VICTORES^{1,2}, A. JARDON^{1,2}, S. MORANTE², M. F. STOELLEN^{1,2},
S. MARTINEZ¹, C. BALAGUER¹

¹RoboticsLab / ²Asociación de Robótica; Universidad Carlos III de Madrid
jcgvicto@ing.uc3m.es

Las tecnologías modernas se están incorporando progresivamente a nuestra vida cotidiana, encontrándonos rodeados de elementos que están compuestas por avanzados sistemas electrónicos embebidos: teléfonos móviles, ebooks, netbooks. Son ordenadores en miniatura que poco se parecen a sus antecesores, que ocupaban habitaciones enteras y sólo podían ser manejados por especialistas. Incluso niños pequeños son capaces de interactuar con sus pantallas táctiles o botones, navegando a través de pestañas, menús, e iconos. Lo que es más, incluyen la electrónica y el software necesarios para proporcionar las capacidades de comunicación inalámbrica que son explotadas por los autores, como se explica en el contenido de este capítulo. Todos estos avances han sido incorporados de forma natural en el ámbito de la robótica. Desde el régimen estacionario de las plantas de producción, ahora esta tecnología se puede encontrar en tiendas de consumo, y en última instancia, en nuestro entorno doméstico. En el amplio alcance de la robótica y la automatización, esto incluye desde persianas automáticas y aspiradoras motorizadas a manipuladores robóticos avanzados menos habituales. La investigación mundial actual se centra en cómo introducir elementos dinámicos y móviles para realizar “tareas domésticas” que requieren manipulación compleja y habilidades de razonamiento. Sin embargo, estas tecnologías empezarán a hacer nuestra vida más sencilla sólo con el desarrollo de interfaces humano-robot que proporcionen comodidad y satisfacción al usuario. En este capítulo, los autores proponen la fusión de la robótica con las capacidades de navegación web de los dispositivos modernos, y presentan implementaciones de prueba de concepto para tres plataformas robóticas diferentes: un robot móvil, un UAV, y un robot manipulador escalador.

1 Introducción

Es un hecho conocido que la tecnología robótica avanza gracias a desarrollos procedentes de otros campos tecnológicos. La mecánica de los robots es cada día más ligera y robusta debido a la utilización de materiales avanzados como pueden ser las estructuras de fibras de carbono. Nuevos materiales, más elásticos y a su vez más resistentes, están dando lugar a los llamados “actuadores suaves”, que permiten reducir riesgos para operarios y usuarios finales de robots. El hardware de la robótica avanza en gran parte gracias a logros en el tratamiento del silicio y materiales semiconductores, y de la tecnología de estado sólido y micro-electrónica en general. Estos son avances significativos que hacen que los robots de hoy en día (y los del futuro) sean mucho más fiables y veloces que los del pasado. Sin embargo, el valor de estos esfuerzos no está en la optimización de un solo parámetro mecánico o de procesamiento: significa llevar esta útil tecnología a un mayor grado de aceptación humana. Se ha comprobado que el grado de aceptación de un manipulador robótico puede verse incrementado si se disminuyen las dimensiones físicas de éste (Victores, 2010). Dispositivos con procesadores de mayor potencia computacional reducen esperas que pueden ser frustrantes y pueden acabar con el rechazo de la tecnología por el usuario final.

En este capítulo, los autores proponen la utilización de una solución software que ya ha alcanzado una amplia aceptación dentro de su contexto original para incrementar la facilidad de uso y la aceptación de dispositivos de manejo, acercando la robótica al usuario final. La tecnología empleada se basa en HTTP (el protocolo más utilizado para la transmisión de páginas web) y diversas implementaciones del concepto de “Cloud Computing” para la Interacción Humano-Robot (HRI, Human-Robot Interaction). El concepto, los beneficios, y el impacto del Cloud Computing se explicarán en la sección 1.1 de este capítulo. Un resumen de tecnologías empleadas en servidores de HTTP se verá en la sección 1.2. En la sección 1.3 se podrá ver una descripción de las plataformas robóticas para las cuales se han desarrollado implementaciones prácticas de HRI basadas en estas tecnologías. Una visión general de la arquitectura software utilizada durante el desarrollo de estas pruebas de concepto podrá verse en la sección 2. Una descripción más profunda de las diferentes implementaciones llevadas a cabo será dada a lo largo de la sección 3 de este capítulo. Finalmente, en la sección 4 se hará un breve repaso de los resultados más relevantes obtenidos, las contribuciones de los desarrollos que se consideran significativas, y las conclusiones a las cuales se ha llegado.

1.1 Computación en la nube

El término “Cloud Computing”, traducido en general como “computación en la nube”, fue acuñado por primera vez en 1997 por Ramnath Chellappa (Chellappa, 1997). El término tiene su origen en la forma de nube con la que se suele representar una red de computadores, véase la Figura 1, que a su vez se utilizaba anteriormente en el contexto de redes de telefonía. En estos diagramas, el perímetro de la nube delimita la barrera que linda entre las responsabilidades del cliente (en el modelo clásico cliente-servidor) y el proveedor de servicios de Internet (ISP, Internet Service Provider). El concepto de la “nube” extiende este perímetro, abarcando innumerables servidores, ISP, y toda la infraestructura de red que los conecta.

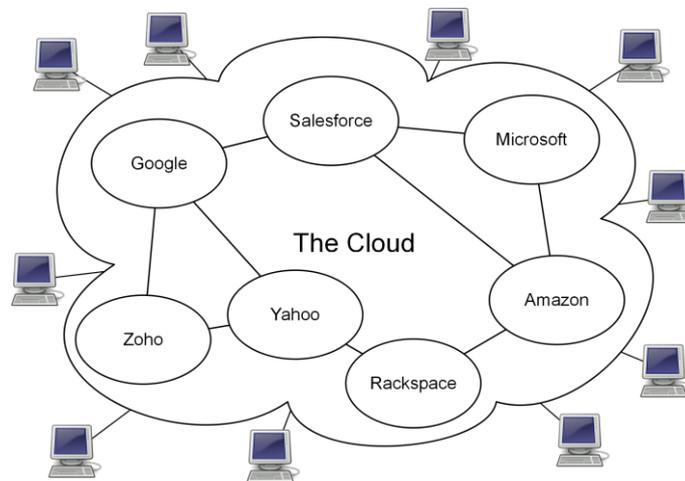


Fig. 1. La forma de nube engloba ISP e infraestructuras de red

La importancia de este paradigma reside en que la “nube” puede ofrecer un número virtualmente ilimitado de *servicios*, de forma descentralizada y a la vez integrada. Estos servicios son accesibles desde cualquier dispositivo con capacidades de navegación web – categoría que en la actualidad incluye móviles, smartphones, PDAs, ebooks, netbooks, y tablet PCs entre otros – sin necesidad de instalación de software adicional. Las funcionalidades de éstos pueden ser tales que lleguen a reemplazar por completo a los paquetes de software que se deben instalar. Un ejemplo de *servicio* ampliamente utilizado en la actualidad es Gmail, que puede utilizarse en lugar de aplicaciones como Thunderbird o Outlook. Existen también soluciones para empresas, como los servicios de planificación de recursos basados en web ofrecidos por Salesforce o SAP AG (Bingi, 1999).

Todavía existe hoy cierta controversia relacionada con los aspectos de seguridad de la nube. Por eso se han definido diferentes tipos de nubes (nubes privadas, nubes híbridas...) que pueden servir para la introducción de diferentes cortafuegos y grados de protección. A través de estos mecanismos las empresas logran cumplir con directivas de protección de datos (FISMA, HIPAA, SOX americanas, 95/46/EC europea). Algunos autores han rebautizado Internet (la “red de redes”) o ciertos componentes de la misma como Intercloud (la “nube de nubes”) en apología a esta tendencia.

1.2 Tecnologías en servidores de HTTP

El protocolo de transferencia de hipertexto (HTTP, Hypertext Transfer Protocol) es, por excelencia, el protocolo más empleado por los servidores de páginas web para transmitir éstas a sus clientes. Las arquitecturas y los sistemas operativos de estas máquinas puede variar, pero a efectos prácticos, todas ellas realizan esta misma funcionalidad (servir páginas web) y utilizan para ello una aplicación común: un servidor de HTTP. El protocolo HTTP permite la transmisión de páginas web escritas en HTML puro, además de la transmisión de imágenes, hojas de estilo, y de contenido más dinámico como pueden ser componentes de ActiveX, Javascript o Applets de Java (estos últimos tres deben ser computados por el lado cliente). Existen diversas aplicaciones y tecnologías en los que se puede basar un servidor de HTTP, y a continuación exponemos algunas de las soluciones de uso más extendido.

- Apache HTTP Server (“httpd”): Se trata de la aplicación servidor de HTTP más utilizada en el mundo, con un 62,71% de cuota de mercado a Mayo de 2011 (Netcraft, 2011). En principio, se trata de una simple aplicación que se ejecuta y que funciona de servidor de HTTP a medida que llegan peticiones por parte de clientes (dispositivos con una conexión TCP/IP que los enlaza con la máquina). Sin embargo, su arquitectura modular permite la ampliación de las funcionalidades de su núcleo. Módulos como `mod_ssl` le aportan posibilidades de servir HTTP bajo conexión segura (HTTPS); `mod_python` permite integrar un intérprete del lenguaje Python dentro del servidor.
- Alternativas semejantes a Apache HTTP Server: Existe en el mercado un gran número de alternativas a Apache HTTP Server, de licencia tanto libre como propietaria. Entre ellos se encuentran los servidores IIS de Microsoft (con una cuota de mercado del 18,37% según las mismas estadísticas), nginx de Igor Sysoev (7.35%), GWS de Google (5.00%), o lighttpd (0.58%).

- Infraestructuras para la creación de aplicaciones web: Se trata de librerías e infraestructuras software para desarrolladores, realizados para diferentes lenguajes de programación (C++, Java, Pearl, Python, Ruby...), y orientadas al desarrollo de webs dinámicas y aplicaciones basadas en web. Su grado de complejidad es variado, siendo en general mayor a medida que el desarrollador utiliza una infraestructura que le permite mayor flexibilidad en cuanto al formato de las páginas ofrecidas y la potencia a la hora de integrar la interfaz con componentes de más bajo nivel.

1.3 Plataformas robóticas utilizadas

En esta sección se describirán los tres robots que se han utilizado como plataformas para las cuales se han desarrollado las interfaces web descritas en este capítulo. Se trata de robots de diferente naturaleza para demostrar el potencial de estas pruebas de concepto: un robot móvil, un robot autónomo volador, y un manipulador escalador avanzado.

1.3.1 *Ecro*

El proyecto *Ecro* (Earth Civil Robot) forma parte de la línea de investigación en robótica terrestre civil de la Asociación de Robótica UC3M – ASROB (González-Fierro, 2010). Consiste en el diseño y desarrollo de un robot móvil (figura 2) para aplicaciones de ayuda a la población civil.



Fig. 2. Robot *Ecro*

Su función principal será la navegación autónoma a través de terrenos desconocidos, utilizando los sensores y elementos que le proporcionan información del exterior. De entre estos dispositivos de recogida de información, destacar el uso de la visión 3D proporcionada por la cámara Kinect de Microsoft. El proyecto engloba tanto diseño mecatrónico como desarrollo software de control y algoritmos de alto nivel. Todo el proceso de diseño, así como el montaje y la programación, han sido desarrollados dentro de ASROB.

1.3.2 UAV

El robot UAV (Unmanned Aerial Vehicle) es un desarrollo de otra línea de investigación paralela de ASROB, la robótica civil de aire. El UAV es una aeronave impulsada con cuatro rotores (véase en la figura 3). A este tipo de vehículos se les conoce también como Quadcopter.



Fig. 3. Robot UAV

Este desarrollo, a diferencia del proyecto Ecro o del robot ASIBOT que se tratará en la siguiente subsección, tuvo una base de mecánica y electrónica con un punto de partida comercial: el kit Xaircraft X600D Quad. Sin embargo, los componentes se han ido modificando a medida que surgían necesidades para adaptarse a los objetivos que se pretendían. Estos objetivos abarcan desde la teleoperación hasta la navegación autónoma.

1.3.3 ASIBOT

El robot asistencial ASIBOT, véase la figura 4, es un manipulador escalador portable de cinco grados de libertad (Jardón, 2006). Fue desarrollado por el grupo de investigación Robotics Lab, inicialmente financiado a través del proyecto europeo IST bajo el nombre de MATS.

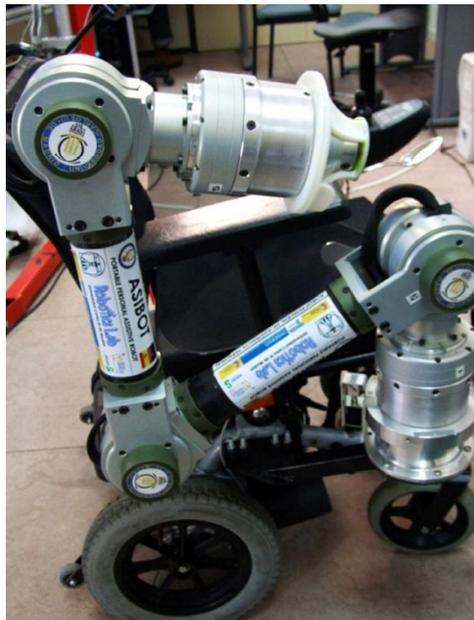


Fig. 4. Robot ASIBOT acoplado a silla de ruedas

El robot posee una serie de exclusivas características que resultan muy ventajosas a la hora de ser utilizado como robot asistencial.

- Estructura ligera y simétrica escaladora.
- Sistemas de control y comunicación embebidos.
- Espacio de trabajo teóricamente ilimitado a través de conectores para escalar suministradores de 24 V.
- Sistema de intercambio de herramientas para pinzas, utensilios, u otras herramientas pasivas o activas adaptadas a tareas.
- Arquitectura de control apta para la integración con el entorno.
- Capacidad de conexión con diferentes dispositivos para la HRI, pensado para diferentes niveles de posibilidades y preferencias de usuarios.

2 Arquitectura software

Como se ha descrito en la sección 1.2, la plataforma más utilizada para servir contenido web es el Apache HTTP Server. Los clientes realizan peticiones al servidor, y éste transmite los contenidos web a través del protocolo HTTP. Este modo de funcionamiento, como previamente se ha descrito, ostenta un amplio cupo de mercado. Es ampliamente adoptado y reconocido. Sin embargo, esta solución impone un límite en el diseño y en alcance de una implementación: lo que se realiza dentro de la aplicación (Apache) queda “dentro” de la misma, quedando “encapsulada” de alguna forma. Esto está representado en la figura 5 (izquierda de la línea a trazos).

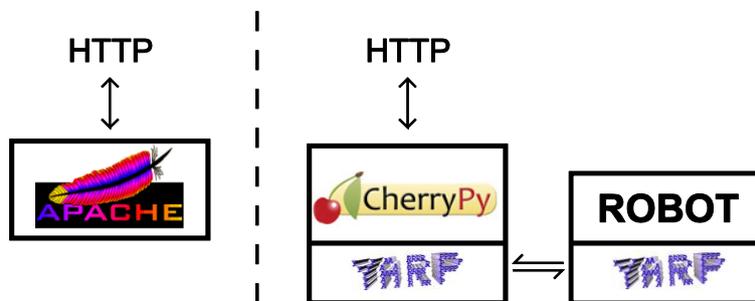


Fig. 5. Arquitectura software habitual (izquierda) y adoptada (derecha)

Para superar esta limitación, existe la posibilidad de utilizar un módulo de extensión de Apache como puede ser `mod_python`. Éste integra un intérprete de Python dentro de Apache, permitiendo la llamada a funciones escritas en este lenguaje desde la interfaz web. Sin embargo, se quería llegar – además de la necesidad de poder llamar a funciones – a un control total del contenido web suministrado. Esto sólo es posible mediante la utilización de infraestructuras para la creación de aplicaciones web (existiendo también la posibilidad de desarrollar una desde cero). Esto es así porque la propia aplicación que se desarrolla es la que interpreta los datos que suministra como contenido web, pudiendo manipularlos en el proceso al tratarse de un elemento intermedio hecho a medida.

Existen infraestructuras para la creación de aplicaciones web para multitud de lenguajes (véase sección 1.2). Para el desarrollo descrito en este capítulo, se ha seleccionado Python como lenguaje de programación. Se trata de un lenguaje que combina los beneficios de un lenguaje interpretado (rapidez en desarrollo y depuración) y una alta eficiencia computacional. Se ha seleccionado CherryPy como la librería utilizada para servir HTTP debido su sencillez y claridad que aporta su API (Hellegouarch, 2007).

La arquitectura software que se utiliza para las comunicaciones con los robots es YARP (Fitzpatrick, 2009). Su adopción se debe a su naturaleza ligera y multiplataforma, su API sencilla, y envoltorios para varios lenguajes (Python entre ellos). Muchas de las librerías que se autodenominan “infraestructuras” imponen una estructura de programa, incluso eliminando la posibilidad de programar una función `main()` con contenido propio. YARP sin embargo, se utiliza mediante sencillas llamadas a funciones dentro del código fuente, en este caso principalmente orientado a la infraestructura dada por CherryPy. Este mecanismo de integración entre la interfaz web y el robot a controlar está representado de forma esquemática en la figura 5 (derecha de la línea a trazos).

3 Implementación técnica

En esta sección se describen las diferentes interfaces que se han desarrollado para las distintas plataformas robóticas. Cada interfaz se ha implementado en dos versiones: completa (para dispositivos de pantalla amplia) y reducida (para dispositivos móviles y similares). La interfaz reducida conserva la funcionalidad de la versión completa, pero prescinde de elementos pesados como las imágenes de alta resolución, que pueden ralentizar o incluso detener la carga de páginas en dispositivos de características muy limitadas.

3.1 Interfaz desarrollada para Ecro

La interfaz desarrollada para Ecro consta de cinco pestañas: Home, PWM Sender, Joypad, Speech y Kinect. A continuación se describe la funcionalidad de cada una de ellas.

- Home: Se trata de la página de entrada inicial de la interfaz. Se proporciona una descripción del robot y una serie de enlaces que pueden resultar útiles o interesantes para el usuario.
- PWM Sender: Esta pestaña permite enviar de forma numérica los anchos de pulso (en microsegundos) de las señales PWM que serán inyectadas a las unidades de control de los motores de las ruedas del robot.
- Joypad: Consta de unos botones que mueven el robot como si de un videojuego se tratase, véase la figura 6. Estos botones envían valores para las señales PWM a las ruedas del robot adecuados al tipo de movimiento solicitado.

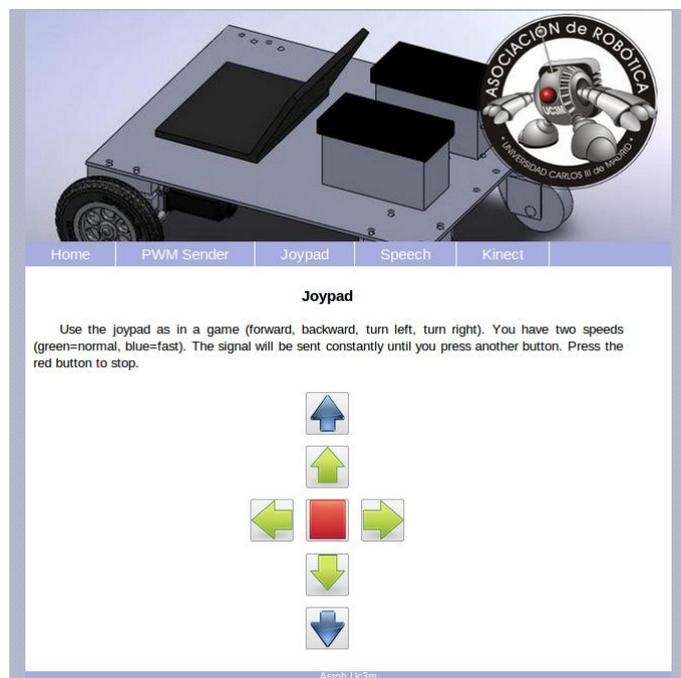


Fig. 6. Pestaña Joypad de la interfaz del robot móvil Ecro (versión completa)

- **Speech:** La interfaz muestra una caja de reconocimiento de voz, la cual tiene unas expresiones guardadas que se corresponden con las señales enviadas a través de la pestaña Joypad. Frases como "adelante" o "izquierda" producen la misma ejecución que pinchar en estos iconos. Está implementado utilizando la API de Google, que explota las posibilidades de HTML5. Es la propia nube de servidores de Google la que realiza el servicio de reconocimiento de voz.
- **Kinect:** Esta pestaña está diseñada para mostrar las imágenes RGBD recibidas a través del dispositivo Kinect para Xbox 360.

3.2 Interfaz desarrollada para el robot UAV

Esta interfaz web, que puede verse en la figura 7, es una adaptación de la interfaz desarrollada para el robot Ecro. Se han implementado sólo dos pestañas: Home (página de inicio) y PWM Sender. Se trata de una prueba de concepto de la adaptabilidad de una misma interfaz (se reutiliza incluso la misma hoja de estilos) y posibilidades de reutilización de código (el código fuente Python que integra CherryPy y YARP es casi el mismo utilizado para la interfaz desarrollada para el robot Ecro) de un robot a otro.



Fig. 7. Pestaña PWM Sender de la interfaz del robot UAV (versión completa)

La reutilización de código es especialmente notable en este caso debido a que la naturaleza de las señales que se envían a las unidades de control de los motores de los robots es la misma, señales PWM. Donde para las ruedas del Ecro se utilizan dos señales PWM, para el control de las hélices del robot UAV se debe disponer de cuatro.

3.3 Interfaz desarrollada para el robot asistencial ASIBOT

La apariencia de la interfaz desarrollada para el robot asistencial ASIBOT es diferente debido a la utilización de una hoja de estilos (CSS, Cascading Style Sheets) diferente. Además, al tratarse de un brazo manipulador, se han incorporado otras características, como el movimiento en ejes cartesianos además de los movimientos del espacio articular. También se han incorporado pestañas relacionadas con la ejecución y grabación de tareas pregrabadas, puesto que en un robot de este tipo cobran gran importancia. Las nueve pestañas que se han implementado son las siguientes: Home, Articular, Cartesian, Command, Teach, Predefined, Video, Docking, Mode. A continuación se explica cada una.

- Home: Es la pestaña de inicio de la interfaz web.
- Articular: Permite movimientos del brazo en el espacio articular. Incluye modo incremental (continuo), relativo, y absoluto referente a la posición inicial de referencia.

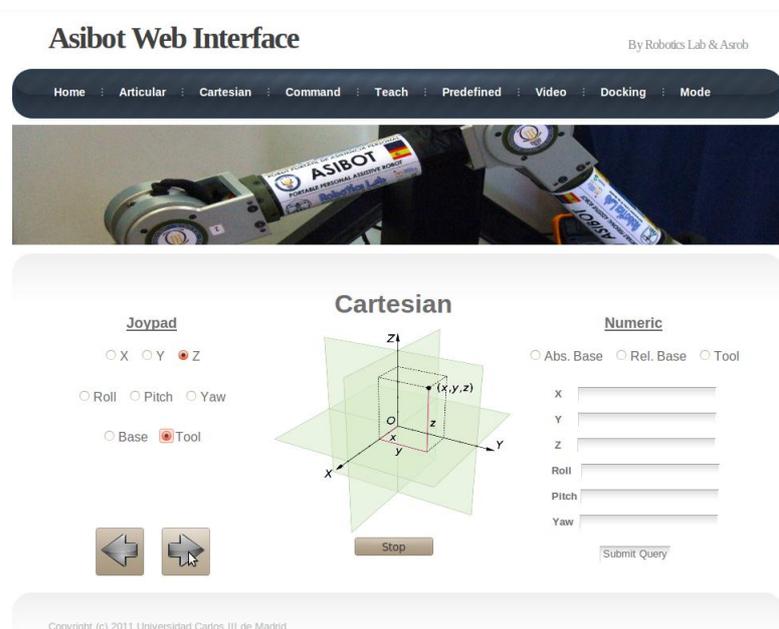


Fig. 8. Interfaz ASIBOT (versión completa): movimientos cartesianos

- Cartesian: Permite movimientos del brazo en el espacio cartesiano, de forma incremental, absoluto, relativo a la base, y relativo a la herramienta. Esta pestaña puede verse en la figura 8.
- Command: A través de esta pestaña pueden enviarse comandos directamente a las unidades de control de los motores del robot. También se implementa reconocimiento de voz en esta pestaña.
- Teach: Permite la creación de ficheros de tareas pregrabadas. Lee la posición en la que se encuentran los ejes y anexa los datos de los encoders (“grabando” la posición) al fichero programa que se selecciona para editar.
- Predefined: Permite la ejecución de tareas pregrabadas. Como puede verse en la figura 9, en lugar de botones con texto, han sido implementados iconos e imágenes representativas de acciones. Estas imágenes pueden manipularse de forma sencilla.
- Video: En esta pestaña se muestran los flujos de imágenes que publican las cámaras IP que se encuentran situadas dentro del entorno donde trabaja el robot asistencial ASIBOT.
- Docking: Permite controlar los solenoides de seguro para el anclaje y des-anclaje del robot de los conectores que utiliza para escalar.

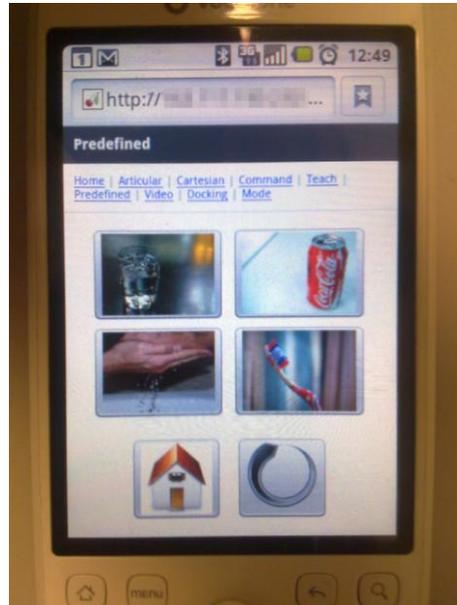


Fig. 9. Interfaz ASIBOT (versión reducida): ejecución de tareas pregrabadas

- Mode: Mediante esta pestaña se determina si se está controlando el robot real, la simulación, o ambas a la vez.

4 Resultados y conclusiones

Se han desarrollado un total de seis interfaces basadas en tecnología web moderna (nótese por ejemplo la utilización de HTML5 validada por W3C). Pese a haber implementado controles útiles para la robótica más avanzada, se ha cuidado el diseño orientado a niños pequeños y a personas discapacitadas o ancianas en todas las interfaces. Podemos destacar los siguientes beneficios.

- No es necesaria la instalación de software. Cualquier navegador se conecta directamente a través de la URL de la interfaz.
- Independencia de resolución de pantalla. Los navegadores web se caracterizan por renderizar el contenido ajustado a la resolución de la pantalla de la que se disponga.
- Multiplataforma desde el punto de vista de sistema operativo. Probado en Windows, Linux, Mac OS X, Android y Symbian.

- Multiarquitectura desde el punto de vista de procesador. Probado en Intel x86, Qualcomm, y ARM.

Esto se resume en que las interfaces son independientes, por tanto, del tipo de dispositivo empleado. Basta con que éste pueda proporcionar una conexión de internet y un navegador. Esto implica además las posibilidades probadas de interacción táctil, multitáctil, y con lápiz de tipo stylus.

4.1 Contribuciones

Cabe destacar que las interfaces desarrolladas no son las primeras interfaces web del mundo utilizadas para controlar robots. En (Saucy, 2000) se describe el uso de una página web para el control de un robot desde laboratorios diferentes. En (Marín, 2005) se propone el uso de la web como alternativa para la utilización de robots entre estudiantes e investigadores. Debemos hacer énfasis en las aportaciones de los desarrollos y la filosofía llevada a cabo.

- Población objetivo. Al contrario que desarrollos anteriores, en este capítulo se busca que el alcance de la población se maximice a toda la sociedad. Buscamos explotar los conocimientos que las personas tienen sobre los dispositivos que utilizan habitualmente, e incrementar las posibilidades de éstos añadiendo capacidades de control de robots sin necesidad de instalación de software.
- Computación en la nube. Se propone e implementa la utilización de servicios ofrecidos por nubes, no sólo de un servidor puntual. A efectos prácticos, esto se puede interpretar como la suma del factor de computación distribuida a la capacidad de proporcionar una interfaz basada en web.

Como ejemplo funcional de este último punto destaca el servicio de reconocimiento de voz incorporado a las interfaces desarrolladas. Se trata de un servicio proporcionado por la nube de un tercero, que además se actualizará de forma gratuita y automática, sin necesidad de modificar la interfaz ni las aplicaciones servidor CherryPy/YARP desarrolladas.

Referencias

Bingi, P. and Sharma, M.K., Godla, J.K. 1999. Critical issues affecting an ERP implementation. *Information Systems Management*—16(3): 7–14.

Chellappa, R. 1997. Intermediaries in Cloud-Computing: A New Computing Paradigm. In *Intermediaries in Electronic Markets*, INFORMS Dallas '97 Meeting.

Fitzpatrick, P., Metta, G., Natale, L. 2008. Towards long-lived robot genes. *Robotics and Autonomous Systems*–56(1): 29–45.

González-Fierro, M., Jardón, A., Martínez, S., Stoelen, M.F., Victores, J.G., Balaguer, C. 2010. Educational initiatives related with the CEABOT contest. In *Proc. of 2nd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPACT 2010)*, Darmstadt, Germany, pp. 649–658.

Hellegouarch, S. 2007. *CherryPy Essentials: Rapid Python Web Application Development*. Packt Publishing Ltd.

Jardón, A., Giménez, A., Correal, R., Cabas, R., Martínez, S., Balaguer, C. 2006. A portable light-weight climbing robot for personal assistance applications. *Industrial Robot: An International Journal*–33(4): 303–307.

Marín, R., Sanz, P.J., Nebot, P., Wirz, R. 2005. A multimodal interface to control a robot arm via the web: A case study on remote programming. *IEEE Trans. Industrial Electronics*–52(6): 1506–1520.

Netcraft. 2011. May 2011 Web Survey. URL (último acceso 9 de Mayo 2011): <http://news.netcraft.com/archives/category/web-server-survey/>

Saucy, P., Mondada, F. 2000. KhepOnTheWeb: Open access to a mobile robot on the Internet. *IEEE Robotics & Automation Magazine*–7(1): 41–47.

Victores, J.G., Jardón, A., Bonsignorio, F., Stoelen, M.F., Balaguer, C. 2010. Benchmarking Usability of Assistive Robotic Systems: Methodology and Application. *Workshop on the Role of Experiments in Robotic Research*, IEEE Int. Conf. on Robotics and Automation (ICRA 2010), Anchorage, Alaska.