Revista Iberoamericana de Automática e Informática Industrial

CEA comité español de automática

ISSN: 1697-7912. Vol. 5, Núm. 2, Abril 2008, pp. 25-37

http://riai.isa.upv.es

MANIPULACIÓN AUTÓNOMA MULTIPROPÓSITO EN EL ROBOT DE SERVICIOS JAUME-2

Mario Prats, Pedro J. Sanz, Ester Martínez, Raúl Marín, Angel P. del Pobil

Departamento de Ingeniería y Ciencia de los Computadores,
Universitat Jaume I, Av. Sos Baynat s/n, 12071 Castellón, España
(e-mail: {mprats,sanzp,emartine,rmarin,pobil}@icc.uji.es)

Resumen: Este artículo presenta una arquitectura de control modular que permite realizar tareas de manipulación en entornos cotidianos a un robot diseñado para ello. Dicho robot integra una plataforma móvil, un brazo redundante articulado, y diferentes tipos de sensores incluyendo visión. Un control de impedancia velocidad/fuerza, que permite la ejecución de una gran variedad de tareas mediante el formalismo denominado TFF (i.e. *Task Frame Formalism*) ha sido implementado de modo satisfactorio. Las distintas tareas son representadas como una red de habilidades básicas que son ejecutadas por el robot usando el control de impedancia. Diferentes mecanismos para la transición entre habilidades de acuerdo al estado perceptual del robot han sido definidos. La validez experimental de nuestro enfoque es demostrada mediante el Robot de Servicios de la UJI, *Jaume-2*, realizando una tarea cotidiana como es abrir una puerta. Finalmente, este nuevo marco es aplicado para progresar en la nueva versión del Robot Bibliotecario de la UJI, demostrando una gran habilidad en el modo en que el robot manipula libros.

Palabras Clave: manipulación; arquitecturas de control; robótica de servicios

1. INTRODUCCIÓN

La transición del paradigma de la robótica de una tecnología industrial específica a los mercados de consumo, cotidianos y de servicios lleva a la aparición de nuevos retos. Los nuevos sistemas robóticos deben ser más inteligentes, flexibles, modulares, fiables y robustos. En particular, la interacción física en nuestros entornos cotidianos requiere sistemas robóticos flexibles dotados de habilidades sensomotoras y con realimentación multisensorial basados en arquitecturas *software* para la percepción y el control que les permitan mostrar niveles de adaptación mayores a través del aprendizaje y la adquisición multimodal de habilidades con el fin de alcanzar objetivos en entornos que no se conocen completamente.

La investigación presentada en este artículo está dirigida a algunas de estas cuestiones en el contexto de interacción física a través de manipulación en determinados entornos cotidianos, donde existe a priori solo un conocimiento parcial del escenario donde se desenvuelve el robot. La manipulación es un problema bien conocido en robótica (Bicchi and Kumar, 2000; Allison *et al.*, 2000). Uno de los principales objetivos de esta comunidad científica ha sido el agarre (Ferrari and Canny, 1992; Borst *et al.,* 1999; Miller *et al.*, 2003), sin embargo, la interacción física presente en nuestra vida diaria cuando se realiza una manipulación, incluso en tareas simples y comunes, va más allá de "coger y dejar" algún objeto. Ejemplos de estas tareas son: encender la luz, extraer un libro de una estantería, abrir una puerta o un cajón, empujar objetos, abrir el grifo, etc. Estas son habilidades básicas que necesitan incorporarse en futuros robots de servicios de una forma robusta y flexible.

Algunas de esas tareas ya han sido tratadas en la literatura exhibiendo una ejecución razonable para las tareas particulares para las que se diseñaron los sistemas. Sin embargo, típicamente pierden adaptabilidad y fallan irremediablemente cuando se introducen pequeños cambios en la tarea. Entre las características que normalmente se tienen en cuenta para este hecho, pueden mencionarse, entre otras, la imposibilidad de garantizar la velocidad de una

dirección controlada por fuerza, el uso de un valor fijo para la ganancia del controlador en una tarea dada o la pérdida de capacidad de modificación o adición de nuevos comportamientos *online*.
En este artículo se presenta una arquitectura *software* que permite la definición y ejecución de tareas complejas de manipulación dentro del Formalismo del Sistema de Referencia de la Tarea (i.e. TFF) (Bruyninckx, 1996; Mason, 1981). La arquitectura incluye las tres formas distintas de especificar el sistema de referencia de la tarea, sugeridas en (Kröger *et al.*, 2004) y permite elegir entre ellas en tiempo de ejecución. Junto con la arquitectura, se presentan varios módulos de acción y percepción para que el robot perciba el entorno y actúe en consecuencia. Por razones de seguridad y ejecución, el robot está dotado de un control de impedancia velocidad/fuerza que considera las fuerzas externas, haciéndolo más robusto a cambios del entorno o modelando los errores. La ganancia del controlador se calcula *online* permitiendo al robot ejecutar varias tareas que no se pueden ejecutar con otro tipo de controladores velocidad/fuerza. También se introduce la noción de "habilidad", como un módulo encargado de definir y supervisar recursivamente tareas. Los módulos de percepción son independientes de las acciones, resultando así en pocas acciones motoras que pueden usarse de forma modular en muchas tareas de manipulación, después de ejecutar las acciones adecuadas de percepción. Además, debido a la definición recursiva de habilidades, es posible construir habilidades más complejas de forma incremental. Nuestro enfoque permite la ejecución paralela de comportamientos del robot y la modificación o adición de nuevos comportamientos *online*, permitiendo metodologías de aprendizaje incremental.

La arquitectura ha sido implementada sobre *Jaume-2*, el Robot de Servicios de la UJI (Prats *et al.*, 2005) (véase Figura 1). El sistema completo proporciona un marco de trabajo general para definir y ejecutar tareas de manipulación cotidianas, de una forma flexible y modular. En particular, este artículo describe dos tareas: girar la manivela de una puerta y extraer un libro de una estantería usando una mano de tres dedos, que es la base de la nueva versión de nuestro Robot Bibliotecario UJI (Prats *et al.*, 2005).

En la Sección 2 se introduce brevemente el Robot de Servicios UJI. La Sección 3 presenta el control de impedancia velocidad/fuerza. La tarea de abrir una puerta se describe en la Sección 5, donde también se presentan los resultados de la ejecución. Después, en la Sección 6, se resume la aplicación del robot bibliotecario, y se informa del progreso alcanzado con respecto a prototipos anteriores, gracias al uso de la nueva arquitectura. En particular, se demuestra la tarea de extracción de un libro imitando los movimientos humanos. Finalmente, algunas conclusiones y direcciones de futuro se esbozan en la Sección 7.

1.1 Trabajo relacionado

La robótica de servicios es un área que está consiguiendo más y más atención a medida que avanzamos en el tiempo. Sin embargo, hasta donde alcanza nuestro conocimiento, hay pocos robots de servicios que pueden ejecutar más de una tarea de manipulación distinta en un entorno cotidiano. Los robots normalmente están programados para ejecutar una tarea especializada concreta (Marrone *et al.*, 2002), y tienen dificultades cuando hacen otras cosas.

Figura 1. *Jaume-2*, el Robot de Servicios de la UJI

Un problema fundamental en el diseño de robots multitarea es que se necesita tener varios controladores y seleccionar cuál de ellos usar en función de la tarea a realizar. Este problema se ha tratado en algunos trabajos relevantes como (Austin *et al.*, 2000), o más recientemente en (Thomas *et al.*, 2003; Kröger *et al.*, 2004). Estos trabajos persiguen objetivos similares a los nuestros. El primero usa su propia arquitectura para abrir una puerta con un manipulador móvil, una tarea que requiere capacidades complejas de manipulación y coordinación. Los otros dos también definen una arquitectura para la ejecución de tareas basada en "*skill primitives*" (Hasegawa *et al.*, 1992), aunque se centran en el ensamblaje industrial más que en robots de servicios.

Un inconveniente importante de los enfoques existentes es que necesitan especificar la tarea a un nivel muy bajo para que el robot la ejecute con éxito. En el trabajo de Thomas et al. (Thomas *et al.*, 2003), el robot ejecuta la tarea usando un control híbrido velocidad/fuerza. Con este controlador, el espacio de posibles movimientos del efector se divide en direcciones controladas por velocidad y direcciones controladas por fuerza, de acuerdo al sistema de referencia de la tarea (*Task Frame*, *TF*). Esto significa que no se puede garantizar la velocidad de una dirección controlada por fuerza, lo cual puede ser necesario en ciertas tareas de servicios, como se explicará en la sección 3.

Por otra parte, Petersson & Kragic (Austin *et al.*, 2000) usan un control de impedancia que es más apropiado para tareas de servicios. Sin embargo,

establecen la ganancia del controlador a un valor fijo para una tarea dada. Cuando se ejecuta otra tarea, se tiene que encontrar otra ganancia experimentalmente, lo que puede presentar un problema en robots que necesiten aprender nuevas tareas de forma autónoma.

Se ha mejorado en esta dirección usando un controlador velocidad/fuerza cuya ganancia se calcula automáticamente *online*, a partir de la velocidad actual del robot y la máxima fuerza que el robot puede aplicar antes de dañar el entorno o romper sus dedos. Estos parámetros constituyen magnitudes naturales para definir una tarea. Además, nuestra arquitectura permite la ejecución paralela de comportamientos del robot y, aún más importante, permite la modificación o adición de nuevos comportamientos *online*. Esta última característica es necesaria, por ejemplo, para el aprendizaje por demostración, cuando se quiere que el robot aprenda más comportamientos sin necesidad de programarlos a mano.

Además, todas las tareas de manipulación se han definido mediante el formalismo TFF (Bruyninckx, 1996; Mason, 1981), que ha demostrado ser un concepto poderoso para la ejecución de tareas de manipulación (Kröger *et al.*, 2004; Baeten *et al.*, 2003). Kröger et al. muestran en (Kröger *et al.*, 2004) tres formas distintas de especificar el sistema de referencia de la tarea para que el robot pueda ejecutar una tarea guiado por sensores. Nuestra arquitectura implementa estos tres casos y, además, permite elegir entre ellos en tiempo de ejecución.

Con respecto a la aplicación del robot bibliotecario, es notable que sólo dos laboratorios distintos hayan investigado en esta área, ambos con el objetivo de hojear libros para el usuario. Así pues, una solución teleoperada fue presentada por Tomizawa (Tomizawa *et al.*, 2003), de la Universidad de Tsukuba, en Japón. Dos tareas importantes que necesitan mejorarse, según la opinión de los autores (Tomizawa *et al.*, 2003), son: la optimización de la planificación de la manipulación y la mejora del reconocimiento del libro, que constituyen precisamente las dos contribuciones importantes de nuestra investigación. El otro sistema está controlado de forma autónoma y fue presentado por un laboratorio de la Universidad John Hopkins. Este trabajo se conoce como CAPM (i.e. *Comprehensive Access to Printed Materials*) (Suthakorn *et al.*, 2002). Su principal objetivo, actualmente en progreso, es poder hojear materiales impresos en tiempo real a través de una interfaz Web.

La contribución principal de nuestro trabajo con respecto a estos sistemas es que el entorno no ha sido modificado para reducir la complejidad del problema. En lugar de localizar los libros en cajas especiales que son más fáciles de coger, como realizan en la John Hopkins, se extraerán los libros tal y como se encuentran en una biblioteca real. La idea subyacente en nuestra investigación se resume en el principio de adaptar el robot al mundo en lugar del mundo al robot.

2. DESCRIPCIÓN DEL SISTEMA

El Robot de Servicios de la UJI (Figura 1) es un manipulador móvil que consiste en un brazo redundante, *Mitsubishi* PA-10, ensamblado en un robot móvil, *PowerBot* de *ActivMedia*. El manipulador está dotado con una Mano de *Barrett* de tres dedos articulados y un sensor de fuerza/aceleración JR3 que está montado en la muñeca del robot. También se dispone de una cámara montada en la muñeca del robot. El manipulador, la mano y los controladores de la cámara, junto con el ordenador de control están montados en la plataforma móvil, y conectados a las baterías del *PowerBot*. El ordenador es un Pentium 4 a 3GHz con 512 Mb de RAM.

La arquitectura descrita aquí se ha programado en C++ haciendo un uso extensivo de la abstracción de clases para proporcionar una forma intuitiva para añadir nuevas partes. Se ha diseñado para usarse conjuntamente con el brazo y la plataforma móvil. Aunque este artículo solo trata la manipulación, los sensores y actuadores del robot móvil también están integrados en la arquitectura.

3. CONTROL DE FUERZA

El control de fuerza de robots para su interacción con el entorno es un tema que ha sido estudiado ampliamente en la literatura (Mason, 1981; Raibert and Craig, 1981; Hogan, 1984; De Schutter and Brussels, 1988a; De Schutter and Brussels, 1988b). Existen varios esquemas para controlar simultáneamente velocidad y fuerza en un manipulador, pero se pueden clasificar principalmente en dos métodos: control híbrido velocidad/fuerza (Raibert and Craig, 1981) y control de impedancia (Hogan, 1984). Aunque existen varios trabajos más recientes sobre control de fuerza, podemos decir que prácticamente todos los esquemas están basados en las ideas originales desarrolladas durante los años 80.

El control híbrido en velocidad/fuerza o posición/fuerza se utiliza cuando el robot debe ejecutar algún tipo de tarea de interacción, porque encaja muy bien con el TFF (Bruyninckx, 1996; Mason, 1981). En este esquema, el espacio de posibles direcciones de la tarea se divide en dos subespacios ortogonales: el subespacio de las direcciones controladas en fuerza (restringidas en posición) y el de la direcciones controladas en velocidad (restringidas en fuerza). Un gran número de tareas pueden ser definidas mediante este enfoque. La única condición es que debe ser posible descomponer el espacio de la tarea en direcciones controladas en fuerza y direcciones controladas en velocidad (Marrone *et al.*, 2002; Baeten *et al.*, 2003).

Ésta condición normalmente se cumple para tareas de interacción en entornos industriales, como limpiar o pulir una superficie. En este tipo de tareas, el control híbrido permite controlar la fuerza explícitamente en las direcciones restringidas en posición. Por ejemplo, para pulir una superficie, se puede controlar el robot para que aplique una fuerza constante de 20N sobre la superficie, a la vez que éste se mueve libremente en el plano tangente al vector fuerza. El control híbrido sería muy adecuado en estas circunstancias.

Sin embargo, al considerar una tarea diaria, como abrir un cajón, es difícil especificarla en términos del control híbrido. En este caso no queremos aplicar explícitamente una fuerza determinada, sino producir un movimiento en el objeto. Puede ocurrir que el cajón sea pesado y la fuerza que el robot aplique no sea suficiente. Por el contrario, el cajón puede ser muy ligero para la fuerza determinada, y entonces el robot ejecutaría la tarea muy rápidamente, quizá dañando el entorno.
Para este tipo de tareas, no se necesita controlar la fuerza explícitamente, sino que la velocidad es más importante. No es natural decir "abre el cajón con 20N de fuerza, no importa la velocidad", sino que es preferible decir "abre el cajón lentamente". El control de fuerza sigue siendo necesario, pero desde un enfoque diferente. El robot debe intentar mantener una velocidad durante la tarea, adaptando su fuerza de forma que ésta no supere el máximo que la mano o brazo pueden soportar. Si el cajón está cerrado, el robot intentaría abrirlo aplicando cada vez más fuerza, hasta que se alcanzaran los límites establecidos.

El control de impedancia permite controlar explícitamente la velocidad en todas direcciones (Hogan, 1984). Consiste en un bucle interno de fuerza que modifica la velocidad que se envía al robot de acuerdo a las fuerzas externas. Para el ejemplo del cajón, este controlador es válido siempre que el bucle de fuerza cancele la velocidad cuando la fuerza alcance el máximo que el robot puede soportar. En el control de impedancia, teniendo en cuenta solamente la matriz de rigidez K (*stiffness matrix*), la fuerza está relacionada con el desplazamiento contra la superficie de contacto, según $F = K(X_d - X) = K \cdot X_e$, siendo X_d la posición de contacto deseada para el robot, y X la posición actual. De aquí se deduce que $X_e = K^{-1}F$. K^{-1} puede verse como una ganancia, α, de forma que, en el caso diferencial, la velocidad de referencia del robot se modifica según la fuerza, de acuerdo a la siguiente ecuación:

$$v' = v + \alpha F \qquad (1)$$

donde v es la velocidad de referencia, v' es la velocidad modificada, y F es la fuerza externa. De la ecuación (1) se puede deducir que v' se anula cuando $\alpha = -v/F$. Por tanto, para anular la velocidad final v' cuando la fuerza F alcanza un valor máximo, independientemente de la velocidad original v, α debe ser actualizada en cada iteración según $\alpha = -v/F_{\max}$, siendo $F_{\max}$ la fuerza máxima que el robot puede soportar. De esta forma, el robot detiene su movimiento siempre que $F = F_{\max}$, independientemente de la velocidad original.

La Figura 2 muestra el esquema del control de impedancia. Se toma como entrada la velocidad deseada, dada en el sistema de referencia de la tarea. Esta velocidad puede ser el resultado de un proceso previo de fusión sensorial, lo cual permite controlar cada dirección con una modalidad sensorial distinta. Las direcciones asignadas a cada controlador se pueden configurar mediante las matrices de selección $S_1, S_2, ..., S_n$, (matrices 6x6 diagonales donde cada elemento de la diagonal puede ser 1 o 0, dependiendo de si el grado de libertad correspondiente es gestionado por ese controlador o no). El resultado es un vector velocidad, dado en el sistema de referencia de la tarea, el cual se utiliza para calcular la nueva ganancia α, y después se modifica según el bucle de fuerza. Otra matriz de selección, S_f, permite seleccionar las direcciones que serán también controladas por fuerza. La velocidad modificada v', se transforma al sistema de referencia del efector y, posteriormente, en velocidad articular según la inversa de la matriz jacobiana del manipulador.

Tal y como se puede deducir de la ecuación (1), la velocidad final v' depende linealmente de la fuerza externa F. Podría ser interesante tener una mejor respuesta del robot en casos extremos, cuando la fuerza actual supera la máxima fuerza que la mano o brazo robot pueden soportar. En este caso, se puede imponer un crecimiento exponencial de la velocidad del robot en los casos en que $|F| > |F_{\max}|$. El funcionamiento del controlador se muestra en la Figura 3, donde se representa la velocidad resultante (para una dimensión) con respecto a la fuerza actual, para tres velocidades originales distintas, y tomando $F_{\max} = 10N$. La figura muestra como, independientemente de la velocidad original, la velocidad controlada es siempre cero cuando $F = 10N$. Si la fuerza sobrepasa el valor máximo permitido, la velocidad crece exponencialmente en sentido contrario.

Figura 2. Esquema del controlador velocidad/fuerza

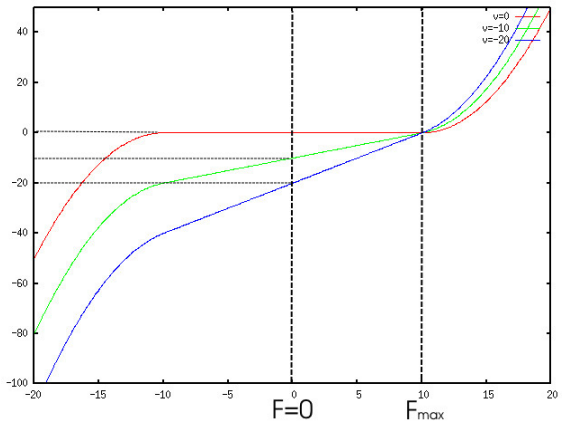

Figura 3. Velocidad resultante con respecto a la fuerza externa actualmente ejercida (en una dimensión)

4. ARQUITECTURA SOFTWARE

Varias arquitecturas software para robots móviles han sido propuestas (Beckey, 2005). Sin embargo, hay pocas contribuciones en el área de manipuladores (Kröger *et al.*, 2004; Petersson *et al.*, 1999). Tal y como Sciavicco y Siciliano destacan en (Sciavicco and Siciliano, 2000), una arquitectura de control para robots manipuladores debe constar de:

- La capacidad de manipulación, para actuar sobre el entorno.
- La capacidad sensorial, para obtener información del mundo.
- La capacidad de procesamiento de datos, para procesar la actividad del sistema.
- La capacidad de comportamiento inteligente, capaz de modificar el comportamiento del sistema según la información externa.

Muchas arquitecturas implementan las primeras características, pero la última de ellas, normalmente, no se tiene en cuenta. Este es el caso para la mayoría de arquitecturas basadas en comportamientos (Arkin, 1998), en las cuales se programa de antemano un conjunto de comportamientos para una determinada tarea. Los robots ejecutando estas arquitecturas funcionan bien con las tareas para las que han sido diseñados, pero son incapaces de crear nuevos comportamientos y expandir sus capacidades.

Desde nuestro punto de vista, éste es un aspecto fundamental para el desarrollo futuro de los robots. Un robot debe ser capaz de ejecutar las tareas para las que ha sido diseñado, pero también de modificar o incluso crear nuevos comportamientos según su experiencia. Con este objetivo en mente, nuestra arquitectura está estructurada en tres grandes módulos: percepciones, habilidades y acciones (véase la Figura 4).

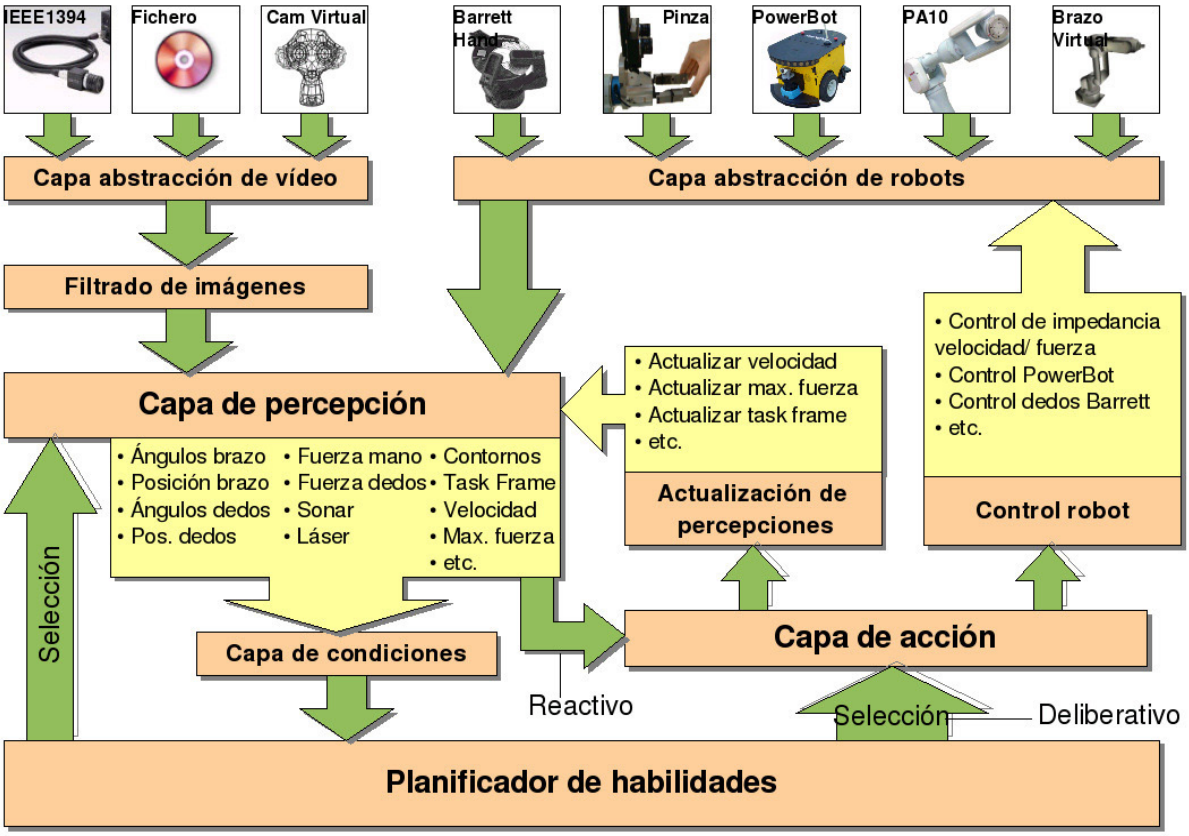

Figura 4. Arquitectura diseñada para el robot manipulador móvil

Como percepción entendemos cualquier información que pueda ser usada para guiar nuestras acciones, como, por ejemplo, la percepción de la fuerza actual en la muñeca del robot, o la percepción de la posición de la mano en el espacio. También consideramos percepciones internas como las referencias, condiciones y objetivos inmediatos del robot. Por ejemplo: la fuerza máxima que los dedos del robot pueden soportar, la condición que compara la fuerza actual en los dedos con respecto al máximo permitido, o la posición del espacio donde el robot ha de mover inmediatamente la mano.

La arquitectura permite evaluar condiciones sobre la mayoría de percepciones, como por ejemplo, evaluar si la fuerza en los dedos es mayor que un valor dado. Muchas de las percepciones dependen de otras. Por ejemplo, la percepción que atiende la fuerza en la muñeca hace uso de la percepción que almacena la posición actual del efector con tal de compensar la gravedad en las lecturas de fuerza. La arquitectura proporciona mecanismos para definir este tipo de relaciones y para integrar fácilmente nuevos sensores. La abstracción de hardware y escalabilidad son características importantes de toda arquitectura (Orebäck and Chistensen, 2003).

La información proveniente de las percepciones se usa para ejecutar acciones de una forma reactiva (Arkin, 1998). El controlador presentado en la sección 3 ha sido integrado en el módulo de acción de la arquitectura, junto con otros algoritmos de control para mover los dedos y la plataforma móvil.
Una acción toma como entrada un subconjunto de percepciones y genera un vector de control que se envía al robot. En particular, la acción que se encarga del control de fuerza del brazo robot toma como entrada las siguientes percepciones:

- La velocidad de referencia.
- El sistema de referencia de la tarea, donde se expresa la velocidad.
- La fuerza máxima que el robot puede soportar.
- La fuerza actual.

Tomando estas entradas, la acción usa el controlador presentado en la sección 3 para calcular el vector de control que finalmente se envía al robot. Esta conexión directa entre percepción y acción proporciona al robot un comportamiento reactivo.

La información proveniente de los módulos de percepción no necesita visitar módulos "superiores" para ser utilizada en la acción. Cambios perceptuales, incluyendo modificaciones en el sistema de referencia de la tarea o la fuerza máxima permitida, son inmediatamente reflejados en el comportamiento del robot.

La arquitectura también permite la ejecución de acciones que no generan movimiento del robot. Estas acciones influyen en el estado perceptual del robot, modificando la velocidad de referencia, por ejemplo. Una acción, en nuestra arquitectura, representa un proceso que puede afectar tanto al estado perceptual como al motor.

Aunque las percepciones se usan para guiar las acciones del robot al más bajo nivel, también se usan para generar una representación del entorno y planificar acciones futuras. Por tanto, la información perceptual también fluye hacia los niveles superiores donde se generan y supervisan planes.

El módulo de habilidades se encarga de seleccionar qué acciones hay que ejecutar, y que percepciones hay que activar en cada momento. Se encarga de definir y supervisar tareas de manipulación. En el marco de nuestra arquitectura, definimos una habilidad como un conjunto de acciones u otras habilidades conectadas por condiciones. Una habilidad puede verse como un autómata, donde los nodos son acciones u otras habilidades, y los arcos son condiciones, construidas sobre percepciones. La principal diferencia con respecto a otros enfoques, como las "*skill primitive nets*" (Thomas *et al.*, 2003; Hasegawa *et al.*, 1992) reside en que nuestras redes no sólo están compuestas de "skill primitives" (Thomas *et al.*, 2003; Kröger *et al.*, 2004), sino que también contienen nodos que actúan sobre el estado perceptual del robot. La ventaja es que la información de la tarea no está codificada junto al código de acción. Por el contrario, ésta se almacena en módulos perceptuales que son independientes de la acción, lo cual permite tener unas pocas acciones motoras generales que pueden ser usadas en cualquier tarea de manipulación, después de haber ejecutado las correspondientes acciones perceptuales.

La Figura 5 muestra un ejemplo de una habilidad con todas las posibles conexiones que la arquitectura permite. Los nodos representan acciones, mientras que los arcos representan condiciones. La arquitectura permite la ejecución paralela de acciones. En el ejemplo de la figura, hay dos nodos iniciales, A1 y A5. Por tanto, estas dos acciones se ejecutarán concurrentemente. Si la condición C1 se hace verdadera mientras se ejecuta A1, el robot comenzará a ejecutar A2 a la vez que seguirá ejecutando A5. Si, en este punto, la condición C2 se activa, la acción A2 terminará, pero solamente la acción A5 estará activa, porque C2 estará esperando a que se active C5. Si esto ocurre, A3 comenzará inmediatamente su ejecución. Si, por el contrario, la condición que se activa mientras se ejecuta A2 es C4, entonces A4 comenzará su ejecución, sin esperar a C5 ni C3. Por tanto, a parte de las conexiones entre habilidades, la arquitectura también proporciona facilidades para sincronización.

Además, debido a la definición recursiva de habilidad como la conexión entre acciones u otras habilidades, es posible integrar habilidades existentes dentro de otras. Con este mecanismo, es posible construir habilidades más complejas incrementalmente. Por ejemplo, el robot podría conocer una habilidad sencilla como moverse en una dirección hasta

detectar una fuerza. Esta habilidad podría ser integrada dentro de otra más compleja, como pulsar un botón, la cual, a su vez, podría formar parte de otra habilidad, y así sucesivamente. Esta característica ofrece un marco para el aprendizaje que nos gustaría abordar en trabajo futuro: el robot podría aprender nuevas habilidades incrementalmente basándose en aquellas que ya conoce.

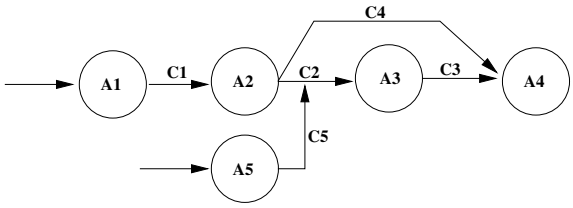

Figura 5. Una habilidad es la conexión de acciones mediante condiciones

5. EJEMPLO DE UNA TAREA

En esta sección se presenta una tarea que el robot ejecuta en base a la arquitectura de la sección anterior. El objetivo de la tarea es abrir una puerta. Un posible método para abrir una puerta fue presentado en (Austin *et al.*, 2000), enfocándose en la última parte de la tarea: seguir el arco de forma adecuada al tirar del pomo de la puerta. Nosotros nos centramos en la primera parte, cuyo objetivo es girar el pomo mediante control de fuerza.

Asumimos que los dedos del robot están colocados sobre el pomo de la puerta, de forma que el sistema de referencia de la tarea se corresponde con el que aparece en la Figura 6. La estrategia del robot radica en mover la mano en la dirección -Y mientras que el valor absoluto de la fuerza sea menor que 15N, valor que fue elegido experimentalmente y debería ser suficiente para girar cualquier pomo. Al mismo tiempo, el robot debe girar lentamente su muñeca mientras que el momento de fuerza en la dirección Z sea menor que 1Nm. Si la velocidad de rotación no es adecuada para el caso particular, aparecerá un momento de fuerza en la dirección Z, y el robot actualizará la rotación de su muñeca en consecuencia, intentando mantener el momento por debajo de 1Nm. Es importante destacar que la rotación de la muñeca también afecta al sistema de referencia de la tarea (Figura 6), modificando la dirección en la cual el robot aplica la fuerza. El resto de grados de libertad también se controlan en fuerza con tal de compensar los errores en el posicionamiento.

Esta estrategia ha sido implementada como una habilidad en la arquitectura. El esquema se muestra en la Figura 7. Cada nodo representa una acción atómica. Estas acciones ya están implementadas en la arquitectura como habilidades básicas. La tarea se define seleccionando las acciones a utilizar, y configurando los valores perceptuales de entrada a estas acciones. El primer nodo se encarga de definir el sistema de referencia de la tarea: se define en el extremo del dedo central de la mano de Barrett. A continuación se define la velocidad de referencia y el valor máximo para la fuerza. Finalmente, el robot ejecuta la acción '*Mover Brazo*', que se corresponde con el control de fuerza presentado en la sección 3. Cuando la fuerza en la dirección Y sobrepasa el valor de 15N, consideramos que el pomo ha sido completamente girado, y la puerta puede ser abierta. El valor de 15N ha sido seleccionado suficientemente alto para asegurar que el robot sea capaz de girar cualquier pomo.

Figura 6. Sistema de referencia asociado a la tarea.

Figura 7. Habilidad para girar una manivela de una puerta

La secuencia completa de la tarea puede observarse en la Figura 8, donde el sistema de referencia de la tarea aparece en rojo. La Figura 9 muestra las fuerzas que aparecen durante la ejecución de la tarea, y cómo el robot reacciona a ellas. Durante la primera parte, la fuerza en la dirección Y se incrementa hasta un valor aproximado de 10N que se corresponde con la resistencia que ofrece el pomo al movimiento. En consecuencia, la velocidad de la mano a lo largo de esta dirección disminuye, ya que la fuerza actual está próxima al máximo permitido. Durante esta fase, y para este caso particular, no existe momento de fuerza en el eje Z, y la muñeca gira a 0.04 rad/s. Durante la segunda fase, el pomo empieza a ofrecer mayor resistencia, y la fuerza finalmente alcanza el valor de 15N, que es el máximo permitido. En este momento, como es de esperar, el robot disminuye la velocidad hasta cero, y la tarea se considera terminada. Es importante destacar que, durante esta segunda fase, el momento de fuerza en Z comienza a incrementarse (en dirección negativa). El robot, por tanto, modifica su velocidad en este eje y detiene el movimiento cuando el valor absoluto del momento de fuerza alcanza el valor máximo permitido de 1Nm.

La combinación de los dos comportamientos permite al robot ejecutar la tarea teniendo en cuenta las fuerzas que se generan en cada caso particular, sin tener conocimiento sobre el modelo geométrico del pomo.

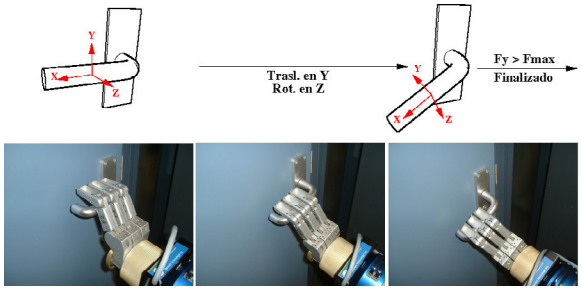

Figura 8. El robot en acción girando la manivela de la puerta

Figura 9. Evolución de las fuerzas y velocidades en la tarea de girar la manivela

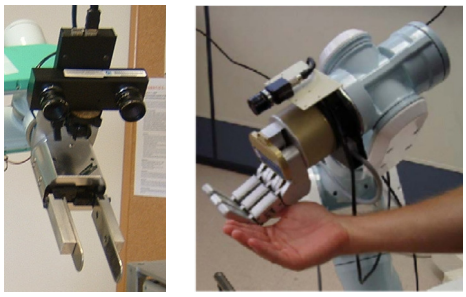

Figura 10. Garra inicial en *Jaume* (izq.) y mano diestra actual en *Jaume-2* (dcha.)

6. APLICACIÓN DEL ROBOT BIBLIOTECARIO

El controlador introducido en la sección 3, y la arquitectura de la sección 4, han sido integrados en una aplicación robótica real: el robot bibliotecario.

En nuestro laboratorio de investigación, se trabaja en un prototipo de robot bibliotecario diseñado para trabajar en este entorno parcialmente conocido. Se pretende que el sistema sea capaz de localizar y extraer cada libro requerido por un usuario. La operación se inicia cuando el usuario pide un libro mediante sus datos (i.e. autor o titulo) o su código asociado. El robot se encarga entonces de localizarlo en una biblioteca convencional, lo extrae y lo entrega al usuario. La única información inicial es el código, que aparece escrito en una etiqueta localizada en el lomo de cada libro.

Después de varios años utilizando el robot denominado *Jaume* para desarrollar esta aplicación (Ramos-Garijo *et al.*, 2003; Prats *et al.*, 2005), ahora se dispone de una versión más avanzada: *Jaume-2*, introducido en la sección 2, con intención de conseguir un sistema general, capaz de realizar diferentes tareas, sin necesidad de haber sido programado específicamente para una habilidad particular. La principal diferencia de este nuevo robot respecto al anterior (Ramos-Garijo *et al.*, 2003) radica en que ahora se dispone de una mano diestra (*Barrett Hand*) de tres dedos articulados, véase la Figura 10, que permite mayor versatilidad y abre nuevos horizontes en la manipulación de todo tipo de objetos.

6.1. El entorno de trabajo

El escenario elegido ha sido la biblioteca de la Universidad Jaume I, en Castellón. Se ha diseñado un modelo completo de esta biblioteca, permitiendo al robot conocer de forma aproximada su estructura. Aunque el modelo de dicha biblioteca ha sido definido con información específica, resulta fácilmente adaptable a cualquier otra biblioteca mediante los siguientes parámetros:

- El número de edificios
- El número de plantas en cada edificio
- El número de estanterías en cada planta, con su respectiva localización y contenido
- El número de módulos y estantes que compone cada estantería

Se asume en nuestro caso que los libros están clasificados mediante el sistema de clasificación de la Biblioteca del Congreso norteamericana, que asocia un código, también denominado signatura, a cada libro.

En las bibliotecas que siguen dicho sistema, los libros se agrupan normalmente por su área de conocimiento, de forma que, dado un código de libro, se sabe aproximadamente su localización espacial.

Este conocimiento es almacenado en la representación de la estantería: además de la localización del estante, la signatura del primer libro que contiene, también se guarda. Conociendo el primer libro de cada estante y asumiendo que los libros están correctamente ordenados, es sencillo detectar que estante contiene el libro que se está buscando.

6.2. Descripción de la aplicación

Para llevar a cabo las tareas descritas se han implementado los siguientes módulos software:

- **Interfaz de usuario**, encargada de transmitir las ordenes del usuario al robot. Debe permitir a dicho usuario controlar de forma remota el escenario robotizado, así como monitorizar información a través de sensores y cámaras. La interfaz de usuario puede usarse tanto en línea como fuera de línea, lo que significa que el simulador 3D permite la especificación de tareas del robot en un entorno virtual para enviar las órdenes a continuación al escenario real (i.e. interfaz predictiva).
- **Navegación**, para el guiado del robot a través de la biblioteca hasta alcanzar la localización deseada. Esta localización podría ser indicada por el usuario remoto, o procesada automáticamente en el modo de operación autónomo.
- **Visión**, encargado de capturar y procesar las imágenes con vistas a localizar y reconocer las signaturas de los libros.
- **Agarre**, responsable de controlar el brazo y la mano robótica para la extracción de los libros.

Un ejemplo de esta interfaz se observa en la Figura 11. Se muestra un modelo 3D virtual de una porción de la biblioteca real. El robot es continuamente modelado con su configuración actual. Si cambia su posición (o los ángulos de sus articulaciones), la nueva configuración es automáticamente reflejada en la vista ofrecida al usuario en la interfaz. Con esta utilidad, el usuario conoce en todo momento la posición del robot. La interfaz también permite al usuario elegir entre diferentes vistas predefinidas, o mover la vista actual en cualquier dirección.
Para el control remoto del robot, se dispone del panel derecho de la aplicación. El usuario remoto puede recibir también, si lo requiere, video en vivo proveniente de la cámara del robot, que se muestra dentro de una pequeña pantalla de la aplicación.

El robot localiza en modo autónomo el estante donde el libro se encuentra, y planifica un camino hasta su posición objetivo. Detalles al respecto están disponibles en (Martinez *et al*., 2005). Cuando el robot está buscando un libro en modo autónomo, navega a través de la biblioteca evitando colisionar con personas u obstáculos. Para ello, se usan algoritmos disponibles incluidos en las librerías de programación del robot.

Cuando el robot alcanza la posición frontal al estante que debiera contener el libro, no conoce la posición exacta de dicho libro. Utiliza la visión para localizar, leer y comparar las etiquetas de los libros. En resumen, el módulo de visión incluye algoritmos para corregir errores de posicionamiento de la cámara, segmentar las imágenes, realizar el seguimiento de las etiquetas y leerlas mediante reconocimiento óptico de caracteres (i.e. OCR en inglés). Estos algoritmos se detallan en (Prats *et al.*, 2005). El robot comienza buscando el libro en los estantes y módulos de la estantería teóricamente esperada. En lugar de buscar el libro uno por uno en el estante completo, se ha implementado un algoritmo de búsqueda inteligente que toma ventaja de la existencia de un orden preestablecido en la disposición de los libros y/o etiquetas asociadas (Martinez *et al*., 2005).

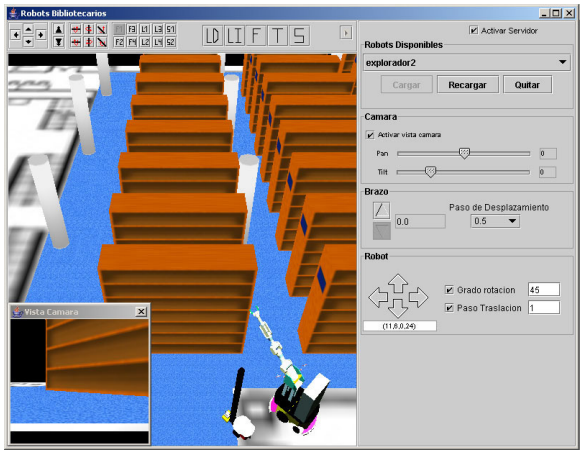

Figura 11. La interfaz de usuario

Cuando el libro deseado es encontrado en la imagen, comienza el proceso de agarre o extracción. Con la pinza de dos dedos plano paralelos, el enfoque era proceder al agarre desde una posición frontal (Prats *et al.*, 2005). Usando realimentación de fuerza, ambos dedos eran abiertos hasta que la distancia entre ellos se correspondía con la anchura del libro. Con la mano de *Barrett*, ha sido posible utilizar un enfoque más natural adaptado de la imitación del mismo movimiento realizado por personas. Este enfoque se explica en la sección siguiente, donde se demuestra que la arquitectura propuesta es suficientemente general para llevar a cabo tareas muy diferentes.

Además, como se introdujo anteriormente, la interfaz de usuario puede utilizarse en modo *offline* para aquellas situaciones donde se pretenda implementar una tarea sofisticada y los robots no estén disponibles en ese momento. De hecho, la interfaz de usuario provee un modo por el que aplicaciones externas pueden enviar instrucciones al entorno de simulación.

Como esta interfaz funciona sobre el protocolo HTTP, mover un robot a una posición es tan simple como abrir una URL (i.e. Uniform Resource Locator). A continuación se muestra un ejemplo de carga de un libro dado en el escenario de simulación 3D desde MATLAB:

```
urlread('http://localhost/command','GET',[{'
name''loadBook'}{'param1'positionBook}]);url
read()
```

Por tanto, usando una herramienta externa como MATLAB para programar algoritmos sobre el robot, pueden enviarse los resultados de dichos programas al simulador 3D de manera simple y directa.

Así por ejemplo, en las figuras 12 y 13, se muestran resultados de un programa en MATLAB que conecta al simulador 3D para la búsqueda de un libro específico en la biblioteca, utilizando técnicas simples de comparación visual por color. Dicha búsqueda es realizada en paralelo por dos robots móviles, y una vez que el libro es localizado se llama al robot manipulador móvil para proceder a su extracción.

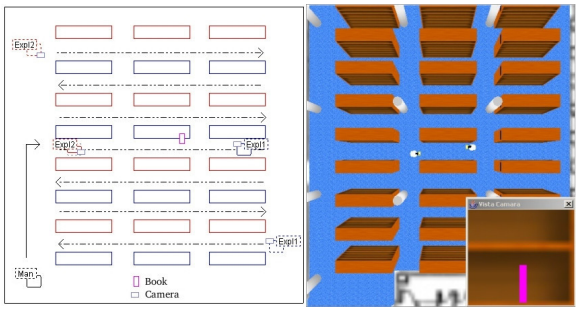

Figura 12. Simulación de robots móviles buscando un libro

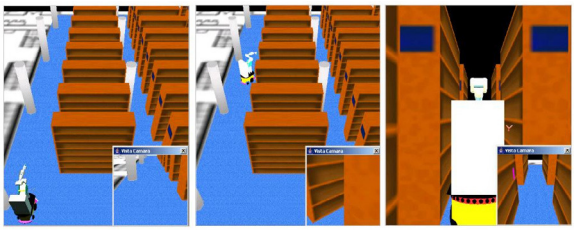

Figura 13. Simulaciones del manipulador acercándose al libro

6.3 Extracción de un libro

La arquitectura presentada en la sección 4 permite la ejecución de tareas complejas tales como la extracción de un libro de una estantería. Este problema ya ha sido tratado anteriormente (Prats *et al.*, 2005), por medio de una pinza de dos dedos plano paralelos, lo cual limitaba en gran medida las posibilidades de manipulación del sistema. Con la recientemente ensamblada mano de *Barrett*, ha sido posible considerar la extracción de libros por imitación de gestos humanos.

Así para extraer un libro, usualmente situamos un dedo en la parte superior y lo hacemos girar con respecto a su base. Es entonces cuando, asistidos por otros dedos, finalmente el libro bascula y lo extraemos. Esta ha sido la estrategia a imitar por el robot *Jaume-2*. La Figura 14 muestra una secuencia completa. El sistema de referencia de la tarea asociado a cada paso es mostrado en rojo. Las flechas representan transiciones entre diferentes acciones. Sobre cada flecha se representa la condición que activa la transición correspondiente. La siguiente acción a realizar aparece debajo de la flecha.

Como muestra la Figura 14, el sistema de referencia de la tarea inicial, en rojo, es asociado al dedo del robot. El robot comienza a moverse en la dirección Z de dicho sistema de referencia, hasta que se alcanza un valor para la fuerza de alrededor de 8 N, lo cual es adecuado para no dañar el libro. En este punto el robot comienza a moverse en la dirección X del sistema de referencia, manteniendo al mismo tiempo una fuerza de 8 N a lo largo de la dirección Z, y rotando alrededor del eje Y. Este movimiento es llevado a cabo por el controlador de impedancia descrito en la sección 3. Cuando los módulos sensoriales del robot detectan que el ángulo con respecto a la posición original es superior a 15 grados, se activa una nueva acción, la cual cierra los dedos y agarra el libro. En este punto, el sistema de referencia es modificado en línea y ajustado a la posición original donde se realizó el primer contacto. El libro se extrae finalmente moviendo la mano con velocidad positiva en X y velocidad negativa en la dirección Z del nuevo sistema de referencia.

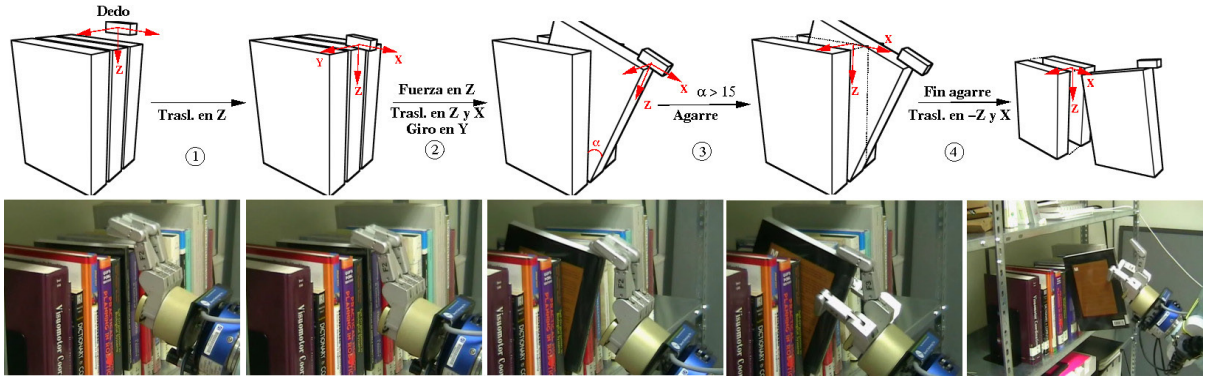

Figura 14. Tarea de extracción de un libro

La Figura 15 representa las fuerzas que aparecen durante la tarea, y las consecuencias en el comportamiento del robot. La primera etapa corresponde al movimiento inicial, cuando el dedo se mueve a lo largo de la dirección Z, buscando establecer el contacto. En este caso no aparece ninguna fuerza (no hay contacto), y el robot se mueve con una velocidad de 10 mm/s a lo largo del eje Z. Cuando se establece el contacto, la fuerza opuesta se incrementa repentinamente, hasta que se alcanza el

valor máximo de 8 N. En este punto, la velocidad a lo largo del eje Z es aproximadamente cero, y el robot comienza a moverse a lo largo de la dirección X, y alrededor del eje Y. Esto hace que el robot consiga imprimir un movimiento de rotación al libro. Con vistas a evitar el deslizamiento del libro en esta etapa, la fuerza sobre el libro debe permanecer constante durante la segunda etapa, y así se muestra en la figura. La etapa 3 corresponde al instante donde los dedos agarran el libro. Esta acción induce cierta inestabilidad en la lectura de las fuerzas, de manera que las velocidades se ajustan a cero durante esta fase. Finalmente, durante la cuarta etapa, las velocidades se ajustan a valores constantes, y el libro es extraído de la estantería. La fuerza positiva que aparece durante esta etapa es debida al peso del libro.

Figura 15. Fuerzas y velocidades durante la extracción del libro

La figura 16 muestra la trayectoria seguida por los dedos durante la ejecución de la tarea, representado en el plano XY del sistema de coordenadas de la base del robot.

Figura 16. Trayectorias de los dedos durante la extracción del libro

7. CONCLUSIONES Y LÍNEAS FUTURAS

En el transcurso de tareas de manipulación, las fuerzas externas juegan un importante papel, y deben tenerse en cuenta siempre para prevenir daños y errores irreparables. Esto es crucial cuando el robot ha de llevar a cabo tareas de servicio en entornos cotidianos, como una biblioteca, donde comparte el espacio con personas. En este articulo, se ha presentado una arquitectura de control que permite la ejecución adaptativa de este tipo de tareas.
Dicha arquitectura implementa un control de impedancia velocidad/fuerza, con una ganancia procesada en línea, teniendo en cuenta la fuerza máxima que puede ejercer el robot. Así mismo, esta arquitectura ofrece una gran variedad de módulos de percepción que pueden combinarse con acciones del robot mediante *habilidades*.

Siguiendo este enfoque, las tareas pueden definirse y realizarse fácilmente por el robot. Se ha experimentado este formalismo haciendo que el Robot de Servicios de la UJI accione la manivela de una puerta. Los resultados obtenidos muestran como el robot adapta suavemente su comportamiento en correspondencia a las fuerzas que aparecen durante la ejecución de estas tareas, haciéndolo robusto a cambios del entorno o errores del modelado. Además esta arquitectura se ha utilizado para programar una nueva versión del Robot Bibliotecario de la UJI, consiguiendo importantes progresos en el modo en que el robot manipula un libro.

Las dos tareas aquí presentadas son ejemplos realizados diariamente por las personas, y se ha tratado de imitar el movimiento humano (véase la Figura 17). Esto ha sido posible por la similitud entre la anatomía del brazo humano y el sistema robótico usado. Sin embargo, la inclusión de nuevas habilidades en el robot posibilitando comportamientos complejos, a partir de la observación de la actuación de las mismas acciones realizadas por personas, es una línea abierta de investigación actual en nuestro laboratorio. Como punto de partida se están siguiendo las ideas de otros expertos en la materia como el paradigma de "imitación a nivel tarea" (Simmons and Demiris, 2004) relacionado con el aprendizaje de acciones asociadas al agarre sin tener que imitar el movimiento exacto del demostrador humano.

Debido al hecho de que la arquitectura permite la modificación en línea y adición de "habilidades" del robot, se planea usarla para entrenamiento del robot en el contexto del aprendizaje. El robot podría aprender más y más "habilidades", cada una basada en la anterior. También se esta explorando la posibilidad de usar percepción visual combinada con información táctil y de fuerza para una adquisición multimodal de habilidades. El objetivo final es conseguir un robot de servicios, que además de coger y mover objetos cualesquiera, interactúe de modo autónomo coexistiendo con personas y realizando tareas cotidianas tales como abrir puertas y cajones, accionar interruptores de luz, manipular libros, etc.

Figura 17. Ejecución humana de la tarea de extracción de un libro

AGRADECIMIENTOS

Este trabajo ha sido desarrollado en el Laboratorio de Inteligencia Robótica de la Universidad Jaume I de Castellón (España). Los autores agradecen al Ministerio de Educación y Ciencia (MEC), a través de los proyectos DPI2004-01920 y TSI2004-05165-C02-01, y a la Generalitat Valenciana, a través del proyecto CTBPRB/2004/052, por el inestimable apoyo recibido para esta investigación.

REFERENCIAS

Arkin, R. C. (1998). *Behavior-Based Robotics*. (MIT Press).

Baeten, J., H. Bruyninckx, and J. De Schutter (2003). Integrated vision/force robotic servoing in the task frame formalism. *International Journal of Robotics Research*, **22**(10-11):941–954.

Bekey, G.A. (2005). *Autonomous Robots*. (MIT Press).

Bicchi, A., and V. Kumar (2000). Robotic grasping and contact: A review. En *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 348–353, San Francisco, CA.

Borst, C., M. Fischer, and B. Hirzinger (1999). A fast and robust grasp planner for arbitrary 3d objects. En *IEEE International Conference on Robotics and Automation*, pp 1890–1896.

Bruyninckx and J. De Schutter (1996). Specification of force-controlled actions in the 'task frame formalism' : A synthesis. *IEEE Trans. On Robotics and Automation*, **12**(5):581-589.

De Schutter, J. and H. Brussels (1988a). Compliant robot motion i. a formalism for specifying compliant motion tasks. *International Journal of Robotics Research*, **7**(4):3–17.

De Schutter, J., and H. Brussels (1988b). Compliant robot motion ii. a control approach based on external control loops. *International Journal of RoboticsResearch*, **7**(4):18–33.

Ferrari, C., and J. Canny (1992). Planning optimal grasps. En *IEEE International Conference on Robotics and Automation*, 3, pp. 2290–2295, Nice, France.

Hasegawa, T., T. Suehiro, and K. Takase (1992). Model-based manipulation system with skill-based execution. *IEEE Trans. on Robotics and Automation*, **18**(5):535–544.

Hogan, N. (1984). Impedance control of industrial robots. *Robotics and Computer-Integrated Manufacturing*, **1**(1):97–113.

Kröger, T., B. Finkemeyer, U. Thomas, and F. M. Wahl (2004). Compliant motion programming: The task frame formalism revisited. En *Mechatronics & Robotics*, Aachen, Germany.

Marrone, F., F. Raimondi, and M. Strobel (2002). Compliant interaction of a domestic service robot with a human and the environment. En *Proc.of 33rd Int. Symposium on Robotics*, Stockholm.

Martinez, E., M. Prats, A. P. del Pobil, and Pedro J. Sanz (2005). Robots behave in the real world: Looking for books in a library. En *The 9th IASTED International Conference on Artificial Intelligence and Soft Computing*, Benidorm, Spain.

Mason, M. (1981). Compliance and force control for computer-controlled manipulators. *IEEE Trans on Systems, Man, and Cybernetics*, **11**(6):418–432.

Miller, A.T., S. Knoop, H.I. Christensen, and P.K. Allen (2003). Automatic grasp planning using shape primitives. En *IEEE International Conference on Robotics and Automation*, pp. 1824–1829.

Okamura, A.M., N. Smaby, and M.R. Cutkosky (2000). An overview of dexterous manipulation. En *IEEE International Conference on Robotics and Automation*, pp. 255–262.

Orebäck, A. and H. I. Christensen (2003). Evaluation of architectures for mobile robotics. *Autonomous robots*, **14**(1):33–49.

Petersson, L. ,M. Egerstedt, and H. Christensen (1999). *A Irbid control architecture for mobile manipulation*.

Petersson L., Austin, D. and Kragic D. (2000). High-level control of a mobile manipulator for door opening. En *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Takamatsu, Kagawa, Japan.

Prats, M., P. J. Sanz, and A. P. del Pobil (2005). Model-based tracking and Hybrid force/vision control for the UJI librarian robot. En *Proc.of International Conference on Intelligent Robots and Systems*, pp. 3308-3313. Edmonton, Canada.

Raibert, M. H., and J. J. Craig (1981). Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, **102**, pp.126–133.

Ramos-Garijo R., M. Prats, P. J. Sanz, and A. P. del Pobil (2003). An autonomous assistant robot for book manipulation in a library. En *Proc. of IEEE Int. Conference on Systems, Man & Cybernetics*, Washington D.C., USA.

Sciavicco, L. and B. Siciliano (2000). *Modelling and Control of Robot Manipulators* (Springer (Advanced Text books in Control and Signal Processing))

Simmons, G. and Y. Demiris (2004). Imitation of human demonstration using a biologically inspired modular optimal control scheme. En *IEEE-RAS/RSJ International Conference on Humanoid Robots*, pp 215–234, Los Angeles, USA.

Suthakorn, J., S. Lee, Y. Zhou, R.Thomas, and S. Choudhury (2002). A robotic library system for an off-site shelving facility. En *IEEE International Conference on Robotics and Automation*, pp 3589–3594.

Thomas, U., B. Finkemeyer, T. Kröger, and F. M. Wahl (2003). Error-tolerant execution of complex robot tasks based on skill primitives. En *Proc. of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan.

Tomizawa, T., A. Ohya, and S. Yuta (2003). Remote book browsing system using a mobile manipulator. En *IEEE International Conference on Robotics and Automation*, pp. 256–261, Taipei, Taiwan.