

## EXPERIENCIAS EN EL DESARROLLO DE UNA APLICACIÓN ROBÓTICA CON CONTROL DE FUERZA PARA TALADRADO DE HUESOS

**J.C. Fraile, J. Pérez-Turiel, J.L. González-Sánchez,  
J. López-Cruzado, J.L. Rodríguez**

*Centro Tecnológico: Fundación CARTIF  
Divisiones de Ingeniería Biomédica y Robótica  
Parque Tecnológico de Boecillo. Valladolid, España  
carfra@cartif.es*

Resumen: En los sistemas de cirugía ortopédica asistida por ordenador (CAOS), el taladrado del hueso en la reparación de fracturas de huesos largos (para estabilizarlos con placas, clavos o tornillos) es una tarea compleja que el cirujano debe realizar con gran precisión, evitando el calentamiento excesivo del tejido óseo. En la cirugía asistida por robots, se han utilizado principalmente robots industriales (Robodoc, Caspar, Crigos) adaptados a ambientes quirúrgicos para ayudar al cirujano en esta tarea. En esta tarea de taladrado de huesos es necesario tener en cuenta varios factores: temperatura del hueso, velocidades de giro y avance de la herramienta de taladrado y fuerzas y pares que actúan sobre el hueso durante su taladrado.

Es fundamental el desarrollo de algoritmos de control del robot manipulador para garantizar que las distintas variables que intervienen en el proceso de taladrado tomen los valores adecuados, bajo el control del cirujano, y así lograr una mayor calidad y precisión en la tarea quirúrgica de taladrado del hueso.

Desafortunadamente, la mayoría de los robots industriales tienen una arquitectura de control propietaria que impide una fácil implantación de algoritmos de control externos basados en realimentación multisensorial (visión, fuerza).

En este artículo se presenta el diseño de una arquitectura de control distribuido y abierto, que se ha implantado sobre el controlador de un robot Staubli. En ella se ha desarrollado una estrategia de control basada en realimentación de fuerza, que permite al cirujano realizar la tarea quirúrgica de taladrado del hueso con elevada precisión. Durante el taladrado con el robot, las fuerzas y pares que actúan sobre el hueso son medidos y utilizados para controlar la velocidad de avance de la herramienta de taladrado, y para controlar la fuerza aplicada al hueso. Se han obtenido resultados experimentales en condiciones de laboratorio con el manipulador equipado con un sensor de fuerza/par y una herramienta para taladrado quirúrgico. *Copyright © 2008 CEA-IFAC*

Palabras Clave: sensor de fuerza/par, control de fuerza, manipulador quirúrgico, cooperación cirujano-manipulador.

### 1. INTRODUCCIÓN

Los sistemas de cirugía integrada por computador (CAS, Computer Assisted Surgery) incluyen sistemas de planificación quirúrgica, sistemas de navegación y dispositivos mecatrónicos, siendo cada vez más habituales en los quirófanos. Proporcionan un conjunto de herramientas tecnológicas, no para sustituir al cirujano, sino para mejorar su percepción, decisión y capacidad de acción en sus intervenciones sobre pacientes. En los sistemas CAOS (Computer

Assisted Orthopaedic Surgery), un robot manipulador puede emplearse como un instrumento activo para incrementar las capacidades de cirujano y proporcionarle seguridad y precisión en las tareas quirúrgicas.

Durante los últimos años, los robots han sido utilizados en neurocirugía, laparoscopia, cirugía ortopédica y otras áreas médicas. Los robots médicos en cirugía ortopédica permiten al cirujano lograr una mayor precisión y calidad en la colocación de

prótesis sobre los huesos. Robodoc (Kazandides, 1999), Caspar (Taylor, 2003), Acrobot (Jacopek, 2003), Crigos (Brandt, 1997) son ejemplos de robots médicos que han sido usados clínicamente en operaciones de fresado y taladrado de huesos, y en operaciones de cadera y rodilla, principalmente.

El taladrado de huesos es un proceso quirúrgico muy habitual en cirugía ortopédica. Es necesario para la inserción de clavos o tornillos en el hueso, durante la reparación de fracturas de huesos largos, y también para la colocación de prótesis en operaciones de fracturas de cadera y rodilla. Requiere un cirujano experto ya que el hueso es un material anisótropo, y por ello, el guiado manual y preciso de la herramienta quirúrgica durante el taladrado del hueso es una tarea difícil. La densidad de hueso y su grosor varían a lo largo de la trayectoria, haciendo que la herramienta de taladrado tienda a deslizar, pudiendo perder la alineación (Wiggins, 1976).

En el proceso de taladrado del hueso intervienen dos variables muy importantes: la velocidad de avance del taladro y la fuerza aplicada al hueso. Si no se controlan adecuadamente estas dos variables pueden aparecer problemas tales como la excesiva generación de calor, que provoca daños irreversibles en el tejido óseo, y la fractura del hueso.

El hueso humano es un mal conductor de calor, su conductividad térmica se encuentra en el rango de 0.38 a 2.3 W/m·K. Parte del calor generado en la zona de taladrado se disipa en la sangre, tejidos o virutas del hueso durante su taladrado. (Hillery, 1999) analiza el efecto de distintas variables: velocidad de corte, velocidad de avance del taladrado y fuerza aplicada, sobre el aumento de temperatura del tejido óseo durante el proceso de taladrado. (Lundskog, 1972) mostró que se produce termonecrosis en el fémur cuando este alcanza temperaturas de más de 50°C durante más de 30 s., debido a la fricción durante el taladrado.

La seguridad es uno de los requisitos fundamentales y críticos para poder utilizar robots en entornos quirúrgicos (Duchemin, 2004). Para conseguir el nivel de seguridad adecuado en el campo de cirugía ortopédica, es necesario utilizar robots equipados con sensor de fuerza. Esto permite el desarrollo de estrategias control del robot basadas en la realimentación de la fuerza medida, que garantizan que el proceso de taladrado de hueso se pueda realizar con valores adecuados de las variables definidos por el cirujano.

La utilización de robots manipuladores con realimentación de fuerza para operaciones de taladrado de huesos, permite al cirujano:

- controlar y restringir los movimientos del manipulador dentro de un espacio de trabajo seguro.
- controlar la precisión de posicionamiento del manipulador y la fuerza aplicada al hueso durante todo el proceso de taladrado.

El grupo de investigación de robótica médica del centro tecnológico CARTIF dispone de un robot "sala limpia" Staubli RX 90, equipado con un sensor JR3 de fuerza/par montado en su muñeca (figura 2). En los últimos años el grupo de investigación (Pérez-Turiel, 2004), (Murariu, 2004, 2005), (Fraile, 2007), ha estado trabajando con este prototipo para el desarrollo de arquitecturas de control, estrategias de control e integración sensorial, en el ámbito de los sistemas de cirugía ortopédica robotizada.

Los objetivos del trabajo de investigación que se presenta en este artículo son:

- El desarrollo de una arquitectura de control distribuida y abierta que pueda ser implantada sobre el controlador cerrado del robot Staubli RX-90.
- El desarrollo e implantación de algoritmos de control de fuerza para el robot.

La arquitectura y los algoritmos de control desarrollados nos han permitido realizar un conjunto de experimentos para una aplicación médica robotizada: el tratamiento de fracturas de huesos largos (fémur). En esta aplicación, el manipulador debe realizar con precisión el taladrado del hueso, en aquellos puntos en los que se colocarán los tornillos proximales y distales para el bloqueo del clavo intramedular que se inserta en la médula del hueso largo. En esta tarea es necesario controlar la fuerza aplicada al hueso, la velocidad de avance de la herramienta de taladrado y la posición del manipulador.

Este artículo está organizado de la siguiente forma: La sección 2 introduce el modelo de referencia formal SIROCO para la integración de subsistemas y componentes CAS cuyos principios y métodos se han aplicado al desarrollo de nuestra arquitectura de control. En la sección 3 se describe el manipulador y la arquitectura de control propuesta. La sección 4 describe el algoritmo de control de fuerza implantado sobre el controlador del manipulador para el proceso de taladrado del hueso. Previamente, en esta sección se explican los experimentos realizados para determinar los valores de algunos parámetros temporales del manipulador que garanticen la estabilidad del algoritmo de control implantado. En la sección 5, se muestran resultados experimentales obtenidos en el taladrado de huesos (fémur) con un robot Staubli RX-90, usando la ley de control de fuerza desarrollada en la sección 4. Finalmente, en la sección 6 se muestran algunas conclusiones.

## 2. MARCO DE REFERENCIA SIROCO PARA INTEGRACIÓN DE SUBSISTEMAS CAS.

La falta de estructuras de interoperatividad es el principal obstáculo para la integración de subsistemas CAS, como los relacionados con imagen médica, planificación preoperatoria, bases de datos anatómicas y técnicas (implantes e instrumentos), etc. En la literatura especializada podemos encontrar referencias a varias propuestas de arquitecturas de sistemas CAS pero se trata, en casi todos los casos, de soluciones ad-hoc, orientadas a problemas específicos y de muy difícil extensión a otros escenarios (Bzostek, 2000; Pernozzoli, 2000; Taylor, 2001).

El Marco de Integración SIROCO (Sistemas RObotizados para Cirugía Ortopédica) (Pérez-Turiel J. et al., 2004) aborda este problema proporcionando un conjunto de criterios, herramientas y métodos orientados al desarrollo de arquitecturas que faciliten la integración segura y eficiente de subsistemas y componentes en el ámbito de las aplicaciones de cirugía asistida por computador.

Una de las dificultades tradicionales al abordar el problema planteado es la falta de unanimidad en la conceptualización y definición de los términos "Marco de Referencia" ó "Marco arquitectónico" y "Arquitectura". Para nuestro objetivo adoptamos la definición de Marco de Referencia elaborada por el "Open Group", que lo define como: *“una herramienta que puede emplearse para desarrollar un amplio abanico de arquitecturas. Debe describir un método para el diseño del sistema a partir de un conjunto de bloques constructivos, indicando como se combinan dichos bloques. También debe proporcionar un vocabulario común y un conjunto de herramientas. Por último, incluirá una lista de normas recomendadas que puedan usarse en la implantación de los bloques”* (Open Group, 2001).

Por otra parte, la recomendación IEEE 1471 define la arquitectura de un sistema como *“la organización fundamental de un sistema basada en sus componentes, sus relaciones mutuas y con su entorno, así como los principios que guían su diseño y evolución”* (IEEE Std 1471-2000., 2000).

Partiendo de este planteamiento, nuestro objetivo ha sido la elaboración de un Marco de Referencia que proporcione criterios, herramientas y métodos para el desarrollo de arquitecturas que permitan la integración eficiente de subsistemas y componentes de un sistema de cirugía asistida por ordenador. La aplicación del marco de referencia a un escenario concreto da como resultado una arquitectura que contempla los aspectos estructurales, funcionales, de información y control, y que sirve de guía para la definición e integración de subsistemas y

componentes, así como para posibles modificaciones o ampliaciones futuras.

Para la consecución de estos objetivos el marco SIROCO incorpora toda una serie de elementos (reglas, modelos, criterios, procedimientos) aplicables a un amplio abanico de situaciones en el campo de la cirugía asistida por computador. Estos elementos se organizan en un esquema multidimensional según tres ejes: Etapas de definición, Generalidad y Agregación, cada uno de las cuales contempla, a su vez, tres niveles o etapas.

Etapas de definición: Esta dimensión tiene en cuenta las fases principales de elaboración de una arquitectura y especifica las herramientas adecuadas para cada fase. Los tres niveles incluidos son:

- Ámbito, propósito y análisis de dominio.
- Metodología
- Especificación arquitectónica

Generalidad: La relación entre los elementos de uno de los niveles y elementos en otro nivel es del tipo generalización - especialización

- Nivel general (Sistemas CAS)
- Nivel parcial (Sistemas CAOS)
- Nivel particular (Sistema CAOS para un dominio concreto, p.e. cadera)

Agregación: La relación entre los elementos de uno de los niveles y elementos en otro nivel es del tipo agregación – descomposición:

- Nivel de Sistema (p.e. Sistema CAOS para prótesis de rodilla)
- Nivel de Subsistema (p.e. subsistema de navegación intraoperatoria)
- Nivel de Componente (p.e. sensor de esfuerzos o algoritmo de correspondencia)

Un bloque constructivo del marco de referencia refleja un nivel de agregación (sistema, subsistema o componente) de una generalidad dada (dominio CAS o CAOS) en una etapa concreta del desarrollo de la arquitectura del sistema (fig. 1).

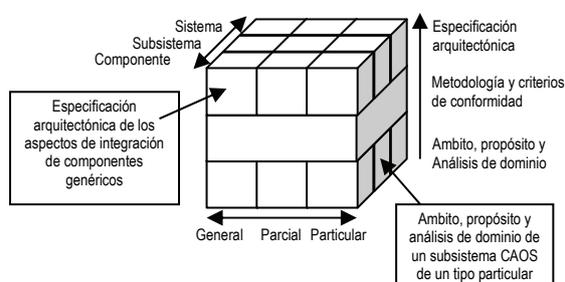


Figura 1. Marco de referencia tridimensional de SIROCO

Nuestro Marco no proporciona descripciones arquitectónicas pero sí facilita modelos de

interacción que representan aspectos concretos del elemento (sistema / subsistema o componente) modelado y, más en particular, de sus interfaces. Estos modelos, agrupados en vistas, forman parte del nivel de especificación arquitectónica y tienen traducción directa en modelos de la descripción arquitectónica (DA) de un sistema, sub-sistema o componente concreto

Un Perfil de interoperabilidad es el "conjunto de una o más especificaciones base y, cuando sea aplicable, la identificación de las clases seleccionadas, subconjuntos de conformidad, opciones y parámetros apropiados de dichas normas, necesarios para realizar una función o actividad concreta". (ISO FDIS 15745-1, 2002). El concepto de perfil de ajusta perfectamente a uno de nuestros propósitos al plantearnos la elaboración del Marco: no desarrollar, salvo que sea imprescindible, nuevas especificaciones de interfaces, protocolos, etc. sino seleccionar, de entre las disponibles, aquellas que más se ajusten a nuestros objetivos y tengan una amplia difusión y aceptación.

Aplicando estos principios, el planteamiento de integración de nuestro Marco se puede resumir en los siguientes puntos:

- La capacidad de interacción de dos recursos, aplicaciones, procesos o subsistemas para conseguir un objetivo común se basa en el uso de **interfaces** estrictamente definidas que proporcionan la funcionalidad correcta.
- La definición rigurosa de esas interfaces se realiza mediante los **perfiles de interacción** asociados.
- Estos perfiles tienen en cuenta los requisitos del sistemas/subsistema/componente, de acuerdo con los **modelos de integración** correspondientes.
- Estos modelos se agrupan en **vistas** que reflejan los aspectos esenciales del sistema/subsistema/componente desde un **punto de vista** dado.
- Para un nivel de generalidad y agregación dado, la **especificación arquitectónica** se realiza mediante la descripción arquitectónica (conjuntos de vistas incluyendo los modelos de integración) y los perfiles de interacción correspondientes.

### 3. ARQUITECTURA DE CONTROL DEL MANIPULADOR

La utilización de un robot manipulador en operaciones quirúrgicas de taladrado de hueso, requiere algunas características importantes tales como: seguridad, fiabilidad, precisión y ejecución de movimientos suaves y seguros (Duchemin, 2004). En

nuestros desarrollos y experimentos se ha utilizado un robot Staubli RX-90 (figura 2). Se trata de un robot manipulador "sala limpia" antropomorfo de 6 grados de libertad, con controlador Adept V+. En la muñeca del robot se ha colocado un sensor JR3 para poder medir fuerzas y pares. Este sensor está conectado al controlador V+ del robot mediante un interfaz VME, y proporciona la lectura de fuerzas y pares cada 1 ms. Se ha acoplado a este sensor una herramienta quirúrgica para taladrado de huesos, que ha sido diseñada y construida para su utilización en esta plataforma quirúrgica robotizada.

El cirujano mueve y desplaza el manipulador "agarrándole" de una pequeña manilla ubicada en el extremo del mismo. El sensor mide las fuerzas y pares ejercidas por el cirujano, y una vez analizadas por el sistema de control, determina la dirección de movimiento del manipulador.

Cuando el manipulador está realizando de manera autónoma el taladrado del hueso, las fuerzas y pares medidos por el sensor son utilizados para controlar la fuerza ejercida sobre el hueso, y determinar la velocidad de avance del taladrado.

La mayor parte de los robots manipuladores industriales disponen de una arquitectura de control cerrada. Por ello, uno de los campos prioritarios de investigación en el ámbito de la robótica, está orientado al desarrollo de una arquitectura abierta y universal para control de robots manipuladores. Un sistema de control cerrado es difícil de modificar, y su arquitectura presenta grandes dificultades para que los usuarios integren software y hardware (Sallinen, 2004). En cambio, los controladores de arquitectura abierta (Kalantar, 2003; Yen, 2000, González, 2003) ofrecen numerosas ventajas, entre las que destaca la disponibilidad pública de interfaces que garantizan la funcionalidad del controlador (Fiedler, 2000; Kahmen, 2001).

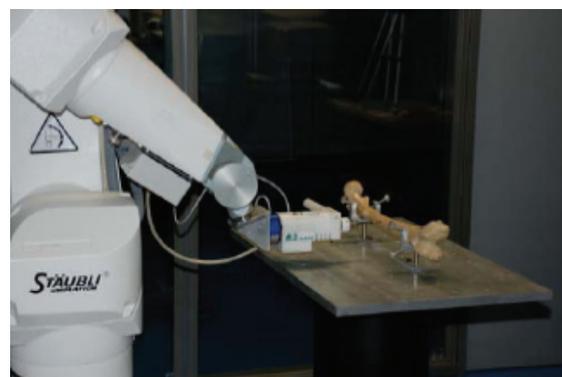


Figura 2. Robot Staubli RX-90 en el laboratorio de CARTIF utilizado en los experimentos.

Existen varias iniciativas de arquitecturas abiertas para planificación y control de movimientos:

LinuxCNC (EMC, 2006), formalmente conocido como EMC (Enhanced Machine Controller), OSACA (Open System Architecture for Controls within Automation Systems) (OSACA, 1996) y OMAC (Open Modular Architecture Controls) (OMAC, 2004), para desarrollar conceptos de arquitectura incluyendo la portabilidad de aplicación a nivel de código fuente y la interoperabilidad de los módulos.

Asimismo, se han propuesto diversas arquitecturas de para el control en aplicaciones de robótica médica en el ámbito quirúrgico:

- Arquitectura de control basada en un ordenador central que mantiene todo el conocimiento sobre los dispositivos conectados (Korb, 2003).
- Aplicación de CORBA (Common Object Request Broker Architecture), en un entorno de control distribuido, con módulos aislados, que se ejecutan sobre diferentes ordenadores, unidos mediante una red de área local (Bzostek, 2000).
- (Peters, 2005) propuso una arquitectura de control distribuida para un sistema de cirugía integrada por computador, en la que los dispositivos son representados por servidores en red, que ofrecen servicios unos a otros

En nuestro caso mediante la aplicación de los principios y métodos que define el marco de referencia SIROCO se ha definido una arquitectura de control distribuida, con el objetivo de conseguir una rápida integración de sensores externos (en nuestro caso, el sensor de fuerza) en el sistema de control del manipulador.

Hay que tener en cuenta que la plataforma utilizada se emplea en nuestro laboratorio para implantar y experimentar otros desarrollos en el ámbito de los sistemas CAOS (guiado a partir de un sistema de visión, toma de nubes de puntos de la superficie del hueso mediante un palpador, etc.). El desarrollo de soluciones ad-hoc para cada dispositivo, tecnología o aplicación complica enormemente la arquitectura resultante e incrementa los costes de desarrollo, implantación y mantenimiento. La aplicación del marco SIROCO nos permite realizar esta integración de forma homogénea y facilita la reutilización de desarrollos.

El interfaz de sensor de fuerza desarrollado es novedoso debido a que:

- la información proporcionada por los sensores puede ser manejada en tiempo real estricto (usando módulos en el espacio del kernel) y no estricto.
- se integra el control de alto y bajo nivel de tal forma que la compensación de bajo nivel (dentro de las tolerancias de los límites de supervisión de sistema) se propague hacia los niveles

superiores de ejecución y control, proporcionando consistencia en el estado y las coordenadas de la trayectoria.

- el sensor de fuerza y su control se construyen sobre el controlador del manipulador, manteniendo habilitados el control interno original y la supervisión de seguridad. Esto hace posible que el usuario pueda seguir utilizando todas las características del sistema original (lenguaje del manipulador, gestión de E/S, etc.).
- los elementos añadidos al controlador original del manipulador pueden ser diseñados y desarrollados usando herramientas normalizadas, lo que permite reducir el tiempo entre la investigación y la aplicación de nuevos algoritmos.

La arquitectura de control del manipulador debe permitir cerrar bucles de realimentación rápidos en un nivel bajo, y también, soportar la programación y la supervisión en los niveles jerárquicos superiores (Blomdell, 2005).

Los complejos algoritmos usados en el control de fuerza, la supervisión de procesos y la gestión de información tienen un alto coste computacional, haciendo inviable que se pueda implantar una arquitectura de control centralizada. Por esa razón, se ha desarrollado e implantado una arquitectura de control distribuido, manteniendo el controlador original (Adept V+), y desarrollando sobre él una arquitectura de control adicional.

La figura 3 muestra los componentes de nuestro sistema: manipulador, ordenador de control y ordenador de gestión-supervisión, unidos mediante red Ethernet.

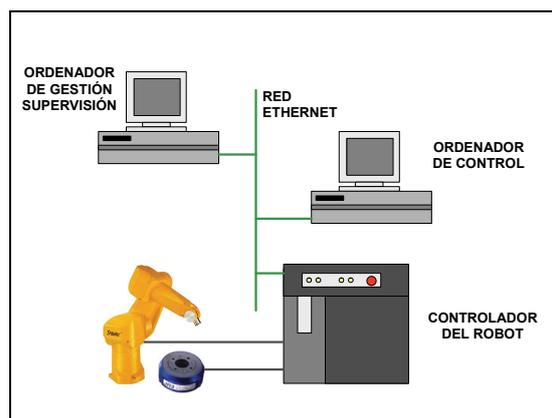


Figura 3. Componentes del sistema

En este caso, se ha aplicado la metodología SIROCO para la “Especificación Arquitectónica”, tercera fase de la dimensión “Etapas de definición” aprovechando desarrollos previos, en el “nivel particular” de la dimensión de generalidad (sistema CAOS para mecanizado óseo), correspondiente a los niveles conceptuales de “subsistema” y “componente” de la

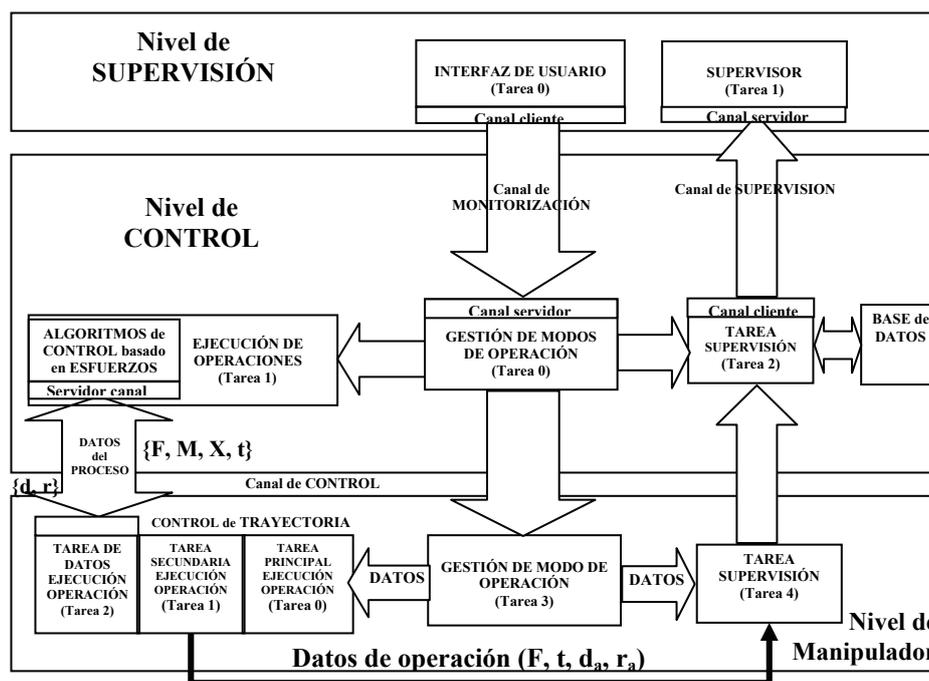


Figura 4. Arquitectura de control implantada en el sistema robotizado para la operación de taladrado de huesos

dimensión de agregación (integración de los componentes “sensor de esfuerzos” y “taladro quirúrgico” en el subsistema “robot para cirugía ortopédica y traumatología”).

La especificación de interfaces (aspecto central del Marco) solo ha tenido que ser adaptada, con relación a desarrollos previos (Pérez-Turiel J. et al., 2004), para tener en cuenta los requisitos tiempo real de la aplicación.

La figura 4 muestra los tres niveles de la arquitectura de control distribuido que se ha implementado sobre el controlador de nuestro manipulador. Estos tres niveles de control están interconectados mediante un enlace TCP/IP:

- **Nivel de supervisión** (PC Windows): Es el responsable de la selección del modo de operación (autónomo, cooperativo) del manipulador, y del algoritmo de control de fuerza que debe emplearse en cada operación. También se encarga de la supervisión del estado del manipulador.
- **Nivel de control** (PC Linux): Gestiona las tareas relacionadas con la ejecución “on-line” del algoritmo de control de fuerza utilizado para la operación de taladrado del hueso. Dada la complejidad de estos algoritmos de control, y sus restricciones temporales, este nivel se ejecuta sobre una plataforma RtLinux.
- **Nivel de manipulador** (Adept V+): Ejecuta las tareas de control de trayectorias del manipulador, la captura de datos de los sensores (posición, fuerza/par), y todas

aquellas tareas básicas necesarias para la correcta operación del manipulador

El **nivel de supervisión** (Figura 4) se ha implantado en una plataforma PC bajo S.O. Windows y está compuesto por dos tareas:

- Tarea 0 – “Interfaz de usuario”: Con dos bloques fundamentales:
  - Gestión y control del manipulador: Posición del TCP, posición y velocidad de los ejes, carga / descarga de programas, arranque / parada de programas, cálculo del TCP y centro de gravedad de la herramienta, definición del volumen de trabajo, selección del algoritmo de control de fuerza, selección del modo de operación del robot (movimiento autónomo o guiado), fuerza deseada durante el taladrado.
  - Gestión de herramienta: Velocidad de rotación, arranque/parada.
- Tarea 1 – “Supervisión del sistema”: Recibe de los otros dos niveles datos de la tarea: fuerzas  $F=\{F_x, F_y, F_z\}$  y pares  $M=\{M_x, M_y, M_z\}$  medidos por el sensor de fuerza, coordenadas actuales de posición  $\{d_a\}$  y de orientación  $\{r_a\}$  del TCP del manipulador, desplazamientos de posición  $\{d\}$  y orientación  $\{r\}$  del TCP del manipulador calculados por el algoritmo de control de fuerza, velocidad de avance del robot y velocidad de rotación de la herramienta.

En el **nivel de control** (figura 4) se ejecutan los algoritmos de control de fuerza desarrollados.

Debido a la complejidad de estos algoritmos, a su alto coste computacional, y a las restricciones temporales, este nivel se ha implementado sobre una plataforma RtLinux, con Kernel Linux (2.4.22) y patch RtLinux (rtl3.2-pre3). Este nivel está compuesto por tres tareas (*threads*) que intercambian información mediante mecanismos IPC:

- Tarea 0 – “Gestión de modos de operación”: Recibe los parámetros del modo de operación del manipulador (movimiento autónomo o guiado), y lanza la ejecución de los programas asociados a dicho modo. Se ejecuta de forma cíclica.
- Tarea 1 – “Ejecución de operaciones”: Es la tarea con el mayor coste computacional, ya que se encarga de ejecutar el algoritmo de control de fuerza seleccionado. Se han implantado dos controladores: PID y control por aprendizaje iterativo. Este algoritmo de control de fuerza calcula los desplazamientos de posición  $\{d\}$  y orientación  $\{r\}$  del TCP del manipulador cada 16 ms. La implantación del algoritmo de control esta ligada a la ejecución de la instrucción Alter del controlador del manipulador, y esta instrucción se puede ejecutar, como mínimo, cada 16 ms, lo que determina que este sea el periodo del algoritmo de control de fuerza que se implante.
- Tarea 2 – “Supervisión”: Se ejecuta en paralelo con las tareas 0 y 1. Obtiene información del modo de operación del sistema. Es una tarea de baja prioridad.

El *nivel de manipulador* (figura 4) es responsable del control de la trayectoria del manipulador y de la captura de datos del sensor de esfuerzos. Se ha implantado en el controlador del manipulador. Las tareas que se ejecutan en este nivel son:

- Tarea principal de ejecución de operación (tarea 0): Se encarga de mover el TCP del robot a su punto de destino, mediante la instrucción “move” del manipulador. Esta tarea se ejecuta en modo cíclico.
- Tarea secundaria de ejecución de operación (tarea 1): Ejecuta la instrucción Alter del manipulador cada 16 ms.
- Tarea de datos para ejecución de operación (tarea 2): Es una tarea cliente para las dos anteriores.
- Tarea de gestión de modo de operación (tarea 3): Dependiendo de los datos recibidos, activa la tarea 0 correspondiente a la operación a realizar.
- Tarea de supervisión (tarea 4): Envía periódicamente al cliente supervisor los datos  $\{F, t, d_a, r_a\}$ .

El protocolo de nivel de aplicación implantado sigue una semántica cliente-servidor con sus

correspondientes tareas en cada uno de los niveles de la arquitectura (figura 4). Entre otros, se dispone de los siguientes tipos de mensajes:

- Inicio/fin de la comunicación entre el cliente y el servidor.
- Modo de operación: selección, inicio y fin del modo activo de operación, junto a sus parámetros asociados.
- Datos de entrada: necesarios para la ejecución del algoritmo de control.
- Datos de salida: proporcionados por el algoritmo de control al controlador del robot.
- Datos de supervisión: contiene los datos necesarios para realizar una supervisión externa del estado del robot.

#### 4. TALADRADO DE HUESO CON CONTROL DE FUERZA

El problema de control de fuerza aplicado a robótica se ha estudiado en profundidad en los últimos años (Yoshikawa, 2000). Se han desarrollado numerosas estrategias de control de fuerza para manipuladores: control de impedancia (Hogan, 1985), control de rigidez activo (Salisbury, 1980), control de fuerza explícito (Volpe, 1993), control híbrido posición/fuerza (Pujas, 1993) y el control de fuerza externo (Perdereau, 1993). (Caccavale, 2005) propone e implementa varias estrategias de control de fuerza en una arquitectura de control abierta para un manipulador industrial.

En el campo de los robots médicos, algunos autores han implantado algoritmos de control con realimentación de fuerza: (Schauer, 2003) desarrolla un robot en miniatura controlado por realimentación dinámica de fuerza, que puede ser utilizado en una amplia gama de aplicaciones médicas. (Kazandides, 1992) emplea estrategias de realimentación de fuerza para el control del robot quirúrgico Robodoc. (Jacopec, 2003) presenta pruebas clínicas con el robot Acrobot para intervenciones quirúrgicas en la rodilla. Utiliza control con realimentación de fuerza, aumentando gradualmente la rigidez del movimiento del robot, para restringir su posicionamiento dentro de una región predefinida. (Duchemin, 2005) presenta el robot quirúrgico Dermanrob, para operaciones de trasplante de piel, con un control de fuerza híbrido. (Ortamier, 2006) presenta resultados experimentales de la utilización de un robot para taladrado de vértebras, para la colocación de los tornillos pediculares. Analiza errores de fuerza y de posicionamiento del robot.

El controlador de fuerza diseñado en nuestro proyecto actúa sobre la velocidad de avance de la herramienta de taladrado, es decir, sobre la

velocidad de avance del manipulador en el eje de la herramienta quirúrgica. Para ello utiliza el control de trayectoria en tiempo de ejecución que permite el control del manipulador mediante la instrucción Alter.

La sintaxis de esta instrucción viene dada por:

Alter Dx, Dy, Dz, Rx, Ry, Rz

donde:

- Dx, Dy, Dz: indican el desplazamiento del TCP del robot a lo largo de los ejes X, Y, Z respectivamente.
- Rx, Ry, Rz: indican la rotación de la herramienta alrededor de los {X, Y, Z} respectivamente.

El sistema de referencia {X,Y,Z}, esta situado en extremo de la herramienta de taladrado (TCP del manipulador), con el eje Z en la dirección del eje de giro de la herramienta de taladrado. Los ejes {X,Y} forman un triedro directo con el eje Z.

Los seis parámetros de la instrucción Alter permiten especificar la magnitud del cambio que debe aplicarse a la trayectoria del manipulador durante el siguiente periodo de cálculo de dicha trayectoria. Las especificaciones técnicas del manipulador Staubli obligan a que dicho periodo sea de  $t_c=16$  ms (frecuencia 62.5 Hz). Como la velocidad de taladrado del hueso debe ser menor que 1.5 mm/s, para evitar daños irreversibles en el tejido óseo, la frecuencia de 62.5 Hz es adecuada para satisfacer los requisitos temporales (transmisión de datos y ejecución de algoritmos de control), que garantizan la estabilidad del sistema.

Se han realizado varios experimentos con el manipulador, para poder determinar el valor de algunos parámetros temporales necesarios para garantizar la estabilidad de los algoritmos de control de fuerza programados.

Estos experimentos han consistido en la ejecución de N (N=10, 100, 200) instrucciones Alter (una instrucción cada  $t_c$  ms.), variando únicamente el desplazamiento Dz (Dz = 0.10, 0.25, 0.50, 0.75, 1.0, 2.5, 5.0, 7.5, 10 mm).

El análisis de los datos obtenidos en los experimentos, permite concluir que el tiempo "T" que el robot tarda en ejecutar el conjunto de N instrucciones, viene dado por:

$$T = (N-1) \cdot t_c + t_a$$

donde  $t_a$  es el tiempo que emplea el manipulador en completar el movimiento, después de recibir la

última instrucción ( $t_a > t_c$ ). Es decir, el robot emplea T ms en recorrer  $N \cdot Dz$  mm.

Los resultados obtenidos demuestran que el parámetro T es independiente del desplazamiento Dz del manipulador, lo que permite realizar un control sobre su velocidad de movimiento. Cuando el manipulador recibe la primera instrucción Alter, trata de recorrer la distancia Dz en un tiempo  $t_a$ . Transcurrido un tiempo  $t_c$ , el manipulador recibe una nueva instrucción y ajusta su velocidad para, en un tiempo  $t_a$ , recorrer la distancia Dz, más lo que le quedaba por recorrer en el instante de recibir la nueva instrucción.

Por tanto, cuando el manipulador recibe la k-ésima instrucción Alter  $k \in [1, N]$ , ajusta su velocidad, y realiza el correspondiente desplazamiento  $Dz(k)$  con una velocidad  $Vz(k)$  dada por:

$$Vz(k) = \frac{Dz(k) + \sum_{i=1}^{k-1} Dz(i) - (Z_k - Z_{ini})}{t_a} \quad (1)$$

donde:

- $Vz(k)$ : Velocidad del robot para completar el desplazamiento  $Dz(i)$ , más la distancia que le queda por recorrer de los desplazamientos programados en las iteraciones k-1 previas.
- $Dz(k)$ : Desplazamiento programado en la k-ésima iteración.
- $Z_{ini}$ : Posición del manipulador en el instante inicial, antes de iniciar el movimiento.
- $Z_k$ : Posición alcanzada por el manipulador en el instante en que recibe la instrucción Alter, en la iteración k.
- $t_a$ : Tiempo que emplea el manipulador en completar el movimiento, después de recibir la última instrucción Alter ( $t_a > t_c$ ).

Entonces, la velocidad media  $V_{mz}$  del manipulador cuando se ejecuta una secuencia de N instrucciones viene dada por:

$$V_{mz} = \frac{Dz \cdot N}{(N-1) \cdot t_c + t_a} \quad (2)$$

El proceso completo de taladrado del hueso tiene una duración de varios segundos. Teniendo en cuenta que cada  $t_c$  se genera una instrucción Alter, resulta que el número "N" de instrucciones generadas durante dicho proceso, es muy elevado. En estas condiciones se verifica que  $t_a - t_c \ll N \cdot t_c$ . Por lo tanto, para un valor de N elevado, a partir de (2) se obtiene la velocidad media  $V_{mz}$ :

$$V_{mz} \approx \frac{Dz}{t_c} \quad (3)$$

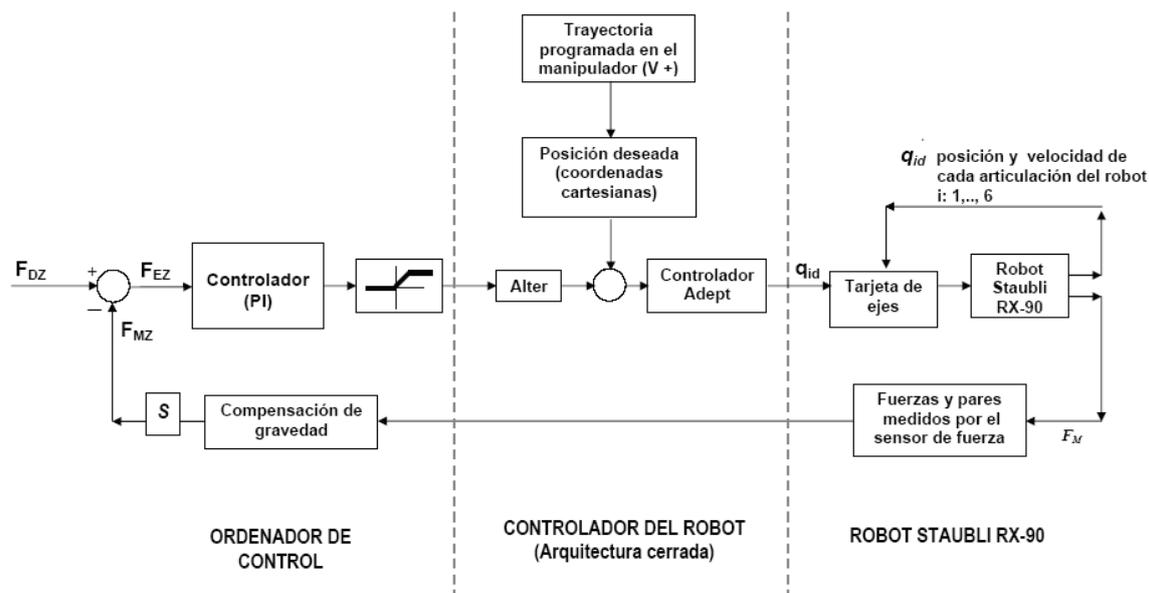


Figura 5. Diagrama de bloques de la estrategia de control del manipulador para la tarea de taladro de huesos utilizando realimentación de fuerza

Se ha utilizado el concepto de control de fuerza externo (Degoulange, 1993), para implantar el controlador de fuerza que actúa sobre la velocidad de desplazamiento del TCP del manipulador durante el proceso de taladrado del hueso.

La tarea del manipulador durante la operación de taladrado del hueso, supone la realización de una trayectoria lineal a lo largo del eje Z del sistema de referencia situado en el TCP del manipulador, entre los puntos  $Z_{ini}$  (posición inicial) y  $Z_{fin}$  (posición final), con la restricción adicional de que la fuerza aplicada al hueso durante el proceso de taladrado, debe mantenerse constante a un valor dado  $F_{DZ}$ .

En la figura 5 se muestra el diagrama de bloques del control de fuerza. Trabaja en las coordenadas cartesianas del marco de referencia del sensor y las entradas son las fuerzas deseadas.

El sensor de fuerza mide la magnitud de la fuerza  $F_M$  en el extremo de la herramienta quirúrgica de taladrado (TCP del manipulador), tanto en el desplazamiento manual guiado del manipulador, como durante su movimiento autónomo en el taladrado del hueso. La fuerza de gravedad actúa en todo instante sobre la herramienta de taladrado. Por ello, se ha realizado una compensación de gravedad en el sensor de fuerza. El valor de esta compensación de gravedad es función de la posición y orientación del conjunto “manipulador + herramienta de taladrado”. Ello obliga a que sea calculada y aplicada “on-line” en todo instante de movimiento del manipulador.

La matriz de selección  $S$  (figura 5) es diagonal y binaria (1, 0), y permite determinar el/los ejes del sistema de referencia  $\{X, Y, Z\}$ , en el/los que la fuerza de taladrado debe ser controlada. En nuestro caso, como el movimiento del manipulador durante el taladrado del hueso es lineal, en la dirección del eje Z, solo es necesario controlar la fuerza en dicho eje. Por ello,  $S = \text{diag}(0, 0, 1, 0, 0, 0)$ , y la fuerza a controlar será la fuerza  $F_{MZ}$  aplicada sobre el hueso durante el taladrado.

En cada iteración de la estrategia de control de fuerza del manipulador con realimentación de fuerza, mostrada en la figura 5, se realiza la siguiente secuencia de cinco pasos:

#### 1) Obtención de la fuerza $F_{MZ}$ :

Lectura del sensor de fuerza, para obtener en ese instante el valor de la fuerza  $F_M$  que se ejerce sobre el hueso durante el proceso de taladrado del mismo. Una vez leído, se filtra para eliminar ruidos de la señal. A continuación, se calcula y aplica la compensación de gravedad, para así obtener  $F_{MZ}$ .

#### 2) Cálculo del error de fuerza $F_{EZ}$ :

El error de fuerza  $F_{EZ}$  es la diferencia entre la fuerza  $F_{DZ}$  que se desea aplicar al hueso durante el proceso de taladrado, y la fuerza  $F_M$  medida por el sensor de fuerza.

#### 3) Cálculo de la velocidad de avance del robot $V_Z$ :

La fuerza axial que se genera en el proceso de taladrado del hueso es función de: la velocidad de avance del manipulador, la velocidad de giro de la herramienta, roca, la geometría de la broca y de las

características del material que se taladra. En nuestra plataforma de experimentación, todos los factores se han considerado constantes, excepto la velocidad de avance del manipulador, que se ha utilizado como variable para controlar la fuerza  $F_{MZ}$  en el proceso de taladrado.

$F_{EZ}$  es la entrada al regulador PID de la acción de control. Este tipo de regulador ha sido ajustado para lograr que  $F_Z \cong F_{DZ}$  durante todo el proceso de taladrado del hueso, y así conseguir la estabilidad del control en lazo cerrado.

El controlador PID recibe como entrada la fuerza  $F_{EZ}$ , y genera la velocidad de avance del TCP del manipulador  $V_Z$  a lo largo del eje Z. Muestreando esta ecuación y teniendo en cuenta los valores de  $V_Z$  en los instantes “K” y “K-1”, se obtiene:

$$V_Z(k) = V_Z(k-1) + K_p [F_{EZ}(k) - F_{EZ}(k-1)] + K_i \cdot t_c \cdot F_{EZ}(k) + \frac{K_d}{t_c} [F_{EZ}(k) - 2 \cdot F_{EZ}(k-1) + F_{EZ}(k-2)]$$

donde:

- $K_p, K_i, K_d$  son las ganancias del PID.
- $V_{Z0}$  es el valor de la velocidad inicial.
- $t_c = 16$  ms. Es el periodo de muestreo.

Hay que tener en cuenta que  $V_Z$  no puede tener un valor negativo, pues ello obligaría al manipulador a retroceder durante el proceso de taladrado del hueso. Para evitar esta situación indeseable, e impedir movimientos bruscos en el avance de la herramienta de taladrado, que podrían dañar el hueso, se ha restringido la velocidad a valores  $V_Z \geq 0$ . Por otro lado, un valor muy elevado de  $F_{EZ}$  puede provocar un avance muy rápido de la herramienta, superior al que el proceso de taladrado del hueso pueda soportar, en función de su naturaleza. Por ello, es necesario limitar el valor de la velocidad de avance a una cota superior, es decir  $V_Z \leq V_{ZSAT}$ . En el esquema de control de la figura 5 estas dos restricciones se especifican a través del bloque no lineal de saturación, que obliga a que  $V_Z \in [0, 1]$ , con lo que se ha considerado  $V_{ZSAT} = 1.5$  mm/s.

#### 4) Desplazamiento del manipulador

Para conseguir controlar la velocidad de desplazamiento del manipulador, se utiliza la instrucción Alter, que permite el control de trayectoria del manipulador en tiempo de ejecución. Según vimos en (3), el parámetro de desplazamiento que debemos proporcionar a esta instrucción para que el manipulador avance con una velocidad media  $V_{zm}$ , viene dado por:

$$Dz \approx V_{zm} \cdot t_c$$

Si se considera el instante “k” de desplazamiento del manipulador, se tiene:

$$Dz(k) \approx V_{zm}(k) \cdot t_c$$

donde:

- $V_{zm}(k)$  velocidad media del robot en el instante k
- $Dz(k)$  valor del desplazamiento en la instrucción Alter, en la iteración k
- $t_c$  periodo de muestreo

#### 5) Control de trayectoria en tiempo de ejecución

La realización del desplazamiento del manipulador a la velocidad calculada  $V_{zm}(k)$ , se obtiene ejecutando la instrucción Alter con el parámetro desplazamiento  $Dz(k)$ .

## 5. RESULTADOS EXPERIMENTALES

En nuestro laboratorio se han realizado un conjunto de experimentos utilizando el esquema de control de fuerza mostrado en el diagrama de bloques de la figura 5, que ha sido implementado en una arquitectura de control desarrollada sobre el controlador del robot Staubli RX-90. El controlador utilizado ha sido un PI, en el que se han ajustado los valores de las ganancias  $K_p, K_i$ , utilizando el método de Ziegler-Nichols. Los experimentos llevados a cabo en el sistema real, permitieron realizar el ajuste definitivo de estos parámetros, obteniéndose los valores que se indican en los experimentos,

Los experimentos realizados han consistido en el taladrado del fémur para la colocación de los tornillos distales, tarea esta realizada por el cirujano durante el acto quirúrgico de reparación de fracturas de huesos largos. Dicha tarea consiste en la siguiente secuencia:

1. El cirujano mueve manualmente el manipulador (controlado por fuerza), agarrándolo y empujándolo en su extremo, desplazando la herramienta a una región del espacio cercana al punto inicial de taladrado del hueso.
2. El manipulador se posiciona automáticamente en el punto inicial de la trayectoria de taladrado, y se orienta hasta alcanzar la dirección de la trayectoria de taladrado.
3. A partir de este momento, el manipulador actúa de manera autónoma. El cirujano indica la fuerza deseada a aplicar al hueso durante el taladrado, y el manipulador realiza automáticamente la tarea de taladrado del hueso sin intervención directa del cirujano, utilizando el algoritmo de control de fuerza mostrado en la figura 5.

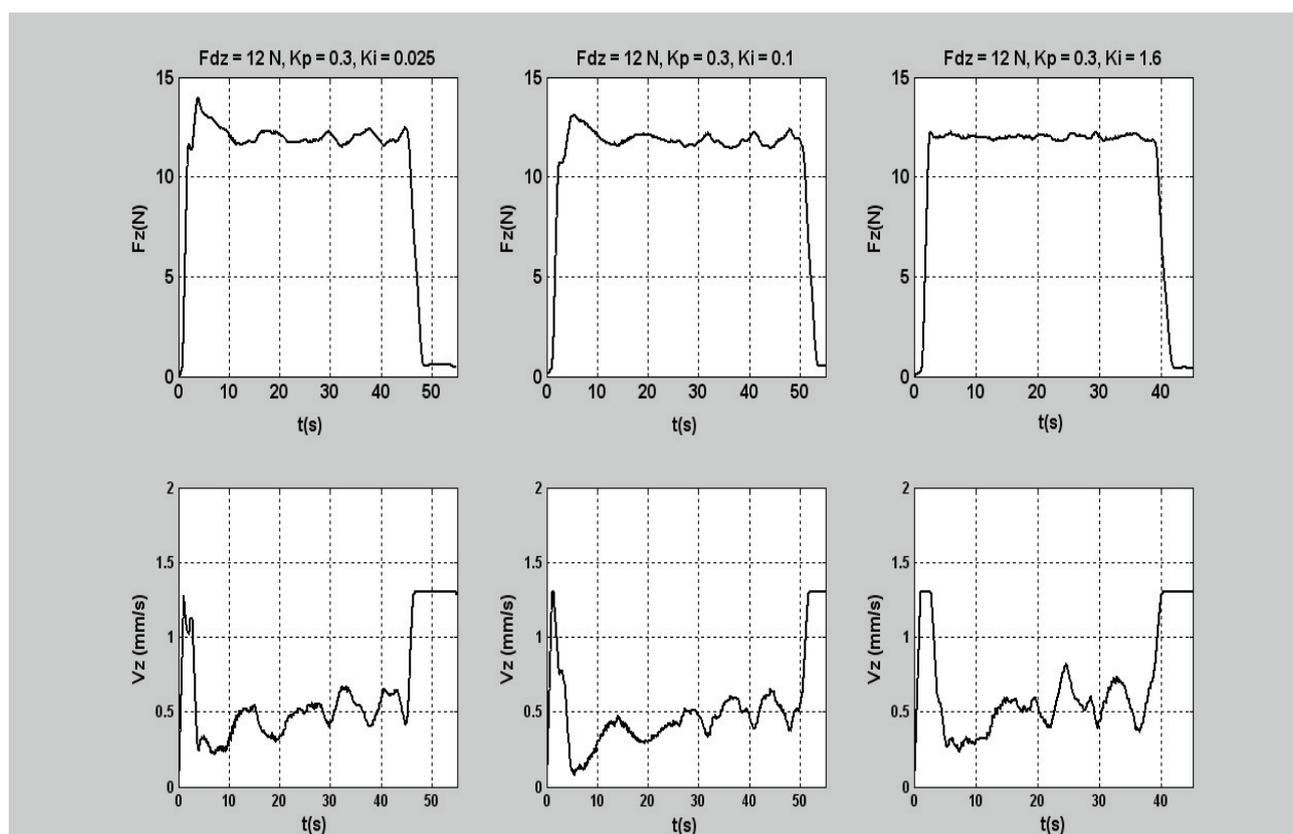


Figura 6. Resultados experimentales obtenidos en la planta real durante el proceso de taladrado del hueso

Los parámetros que han caracterizado los experimentos realizados en el taladrado de fémur humano son:

- Velocidad de giro de la herramienta de taladrado: 5000 rpm
- Diámetro de herramienta: 3 mm.
- La dirección de taladrado coincide con el eje Z del sistema de referencia  $\{X,Y,Z\}$  ligado al TCP del manipulador.
- Fuerza  $F_Z$  que se desea aplicar al hueso durante su taladrado: 12 N.
- La velocidad  $V_Z$  debe ser  $\leq 1.5$  mm/seg. Este requisito es necesario para reducir las fuerzas de impacto y evitar daños al tejido óseo.

Se han realizado experimentos utilizando realimentación de fuerza con un control PI. La fuerza deseada aplicada al hueso durante el taladrado, se ha tomado  $F_{DZ} = 12$  N. Como se ha indicado, en una primera fase se han ajustado las ganancias  $K_p$  y  $K_i$  del controlador PI utilizando Ziegler-Nichols. En una segunda fase, se realizó un conjunto de mediciones sobre la planta real, que permitieron hacer el “ajuste empírico fino” de los valores de las ganancias del controlador PI.

En la figura 6 se muestran los resultados obtenidos realizando mediciones sobre la planta real, durante

procesos de taladrado de huesos, y utilizando para el controlador PI las ganancias:  $K_p = 0.3 \text{ mm}\cdot\text{N}^{-1}$  y  $K_i = 0.025, 0.1, 1.6 \text{ mm}\cdot\text{N}^{-1}\cdot\text{s}^{-1}$ , obtenidas según el procedimiento antes descrito. El objetivo es anular el error estacionario de la fuerza aplicada al hueso durante su taladrado, mientras se mantiene el efecto de la ganancia  $K_p$  sobre el transitorio de la fuerza dentro de unos niveles adecuados.

En las tres gráficas superiores de la figura 6 se muestra la evolución temporal de la fuerza real  $F_z$  que se aplica al hueso durante el taladrado del mismo, utilizando realimentación de fuerza con un controlador PI. Se observa que modificando la ganancia integral  $K_i$ , se logra una reducción del error estacionario de la fuerza  $F_z$ .

En las tres gráficas inferiores de la figura 6 se muestra la evolución temporal de la velocidad de avance del TCP del manipulador  $V_z$ , durante el proceso de taladrado del hueso. Se observa que se mantiene por debajo del valor deseado (1.5 mm/s), pero presenta fluctuaciones en el intervalo 0.5 – 0.75 mm/s, que son debidas al diferente grado de dureza que presenta el hueso en las capas que conforman su estructura.

## 6. CONCLUSIONES

Este artículo presenta el diseño y desarrollo de una arquitectura de control implantada sobre el controlador de un robot Staubli, utilizando realimentación de esfuerzos. Como aplicación, se muestra el proceso de taladrado de huesos con control de fuerza. En esta tarea, el robot actúa como una herramienta activa que permite al cirujano ortopédico realizar el taladrado de hueso utilizando valores adecuados de los parámetros de posición, velocidad y fuerza, que se pueden supervisar durante todo el proceso.

El desarrollo de la arquitectura a partir de un marco de referencia previo (SIROCO), nos ha permitido centrar nuestros esfuerzos en los aspectos relacionados con los requisitos de tiempo real y fiabilidad de la aplicación, ya que el problema de la interoperabilidad queda resuelto implantando los interfaces correspondientes, a partir de los perfiles de interacción proporcionados por el Marco.

El algoritmo de control de fuerza presentado y probado permite controlar la velocidad de avance de la herramienta a lo largo de la trayectoria de taladrado del hueso, según las fuerzas medidas “on-line” en la punta de la herramienta. Esto permite adaptar la velocidad de taladrado a diferentes densidades del hueso garantizando en todo momento que la magnitud de la fuerza que actúa sobre el hueso es prácticamente constante, e igual al valor de referencia deseado por el cirujano.

Los experimentos realizados apoyan nuestra tesis de que el manipulador con sensor de fuerza, y utilizando estrategias de control de realimentación de fuerza, mejora la precisión de la tarea quirúrgica de taladrado de hueso.

Se han realizado varios experimentos para demostrar la robustez de la arquitectura de control propuesta y analizar y validar la herramienta quirúrgica desarrollada para el proceso de taladrado de hueso.

La arquitectura de control implantada sobre el controlador original del robot se utilizará en trabajos y desarrollos futuros como una plataforma experimental, para el desarrollo de nuevos algoritmos de control de fuerza.

Los robots quirúrgicos normalmente no son dispositivos aislados, sino elementos que forman parte de un sistema completo diseñado para trabajar en una sala de operaciones. Ningún sistema puede cubrir todas las necesidades. Por tanto, el desarrollo de arquitecturas altamente modulares que permitan un elevado grado de reutilización de componentes certificados, con interfases abiertas entre los

principales subsistemas, será un gran desafío, tanto técnica como socialmente, dentro de la comunidad de robótica médica.

## AGRADECIMIENTOS

Este trabajo de investigación ha sido parcialmente financiado por el proyecto CICYT DPI2003-09705-C02-01.

## REFERENCIAS

- Blomdell A. et al. (2005) Extending an industrial robot controller: implementation and applications of a fast open sensor interface. *IEEE Robotics & Automation Magazine*, **12** (3), 85- 94.
- Bzostek A. et al. (2000) Distributed Modular Computer-Integrated Surgical Robotic Systems. Implementation using Modular Software and Networked Systems. *MICCAI*, 969-78.
- Brandt G. et al. (1997) A compact robot for image guided orthopaedic surgery. *Proc. CURMed-MCRAS 97*, 779-788. *Lectures Notes in computer science series*, **1205**. Springer-Verlag.
- Bzostek A. et al. (2000) Distributed Modular Computer-Integrated Surgical Robotic Systems: Implementation using Modular Software and Networked Systems; *Proc. of Medical Image Computing and Computer Assisted Interventions*, Pittsburgh, PA, USA.
- Caccavale, F. et al. (2005) Integration for Next Generation. Embedding Force Control into Industrial Robots. *IEEE Robotics & Automation Magazine*, **12** (3), 53- 64.
- Davidson S.R., James D.F. (2003) Drilling in bone: modeling heat generation and temperature distribution, *J. Biomech. Eng.*, **125**, 305-314.
- Degoulange E., Dauchez P. Pierrot F. (1993) Determination of a force control law for an industrial robot in contact with a rigid environment. *Proc. Int. Conf. on Systems, Man and Cybernetics*, **2**, 270 – 275.
- Duchemin G. et al. (2004) Medically safe and sound. *IEEE Robotics and automation magazine*, vol. **XX**, 46-55.
- Duchemin G. et al. (2005) A hybrid position/force control approach for identification of deformation models of skin and underlying tissues. *IEEE Tran. On Biomedical Engineering*. **52** (2), 160-170.
- EMC : Enhanced Machine Controller Project. (2006) <http://www.linuxcnc.org/>.

- Fiedler P., Schilb C. (2000) Genesis System Group. Open Architecture Robot Controllers and Workcell Integration.
- Fraile J.C. et al. (2007) Experiments on bone drilling control with a robot arm for computer integrated surgery. *Proc. CAOS 2007 International – Computer Assisted Orthopaedic Surgery*, Heidelberg, 157-162.
- González, J.L., et al. (2004) Desarrollo de un controlador abierto para un robot industrial tipo SCARA. *RIAI: Revista Iberoamericana de Automática e Informática Industrial*. **1** (1), 44-49.
- Hillery M.T., Shuaib I. (1999) Temperature effects in the drilling of human and bovine bone. *Journal of Materials Processing Technology* **92-93**, 302-308.
- Hogan N. (1985) Impedance control: An approach to manipulation. *Tran. ASME. Journal of dynamics systems, measurement and control*, **107**, 1-24.
- IEEE Std 1471-2000. (2000) Recommended Practice for Architectural Description of Software-Intensive Systems. The Institute of Electric and Electronic Engineers.
- ISO/IEC TR 10000-1. (1998) Information technology - Framework and taxonomy of International Standardized Profiles. Part 1: General principles and documentation framework.
- ISO FDIS 15745-1.(2002). Industrial automation systems and integration - Open systems application integration framework. Part 1: Generic reference description.
- Jacopek M. et al. (2003) The hands-on orthopaedic robot Acrobot: Early clinical trials of total knee replacement surgery. *IEEE Tran. on Robotics and Automation*, **15**, 902-911.
- Kahmen A. (2001) Open Controller Enabled by an Advanced real time Network. OCEAN Consortium.
- Kalantar S. (2003) *YAOC-13: A generic open system architecture and software framework for general motion control and industrial robotic systems*. University of Tehran.
- Kazandides P. et al. (1992) Force sensing and control for a surgical robot. *Proc. IEEE ICRA*, 612-617
- Kazandides P. (1999) Robot assisted surgery: The Robodoc experience. *30th Symp. on robotics*, pp. 281-286. Tokyo, Japan.
- Korb W. et al. (2003) Development and First Patient Trial of a Surgical Robot for Complex Trajectory Milling. *Comp Aid Surg.* **8**, 248-257
- Lundskog J. (1972) Heat and bone tissue. An experimental investigation of the thermal properties of bone tissue and threshold levels for thermal injury. Thesis, University of Göteborg.
- Murariu A.M. et al. (2004) The effects of drilling on cortical temperatures. *Proc. Biomaterials and medical devices components – BiomMedD 2004*, : Bucarest, 133-137.
- Murariu M.A. et al (2005) Heat generation and temperature distribution in human cortical bone drilling. *Proc. CAOS 2005 International – Computer Assisted Orthopaedic Surgery*, Helsinki, 329-334.
- OMAC Users group. (2004) <http://www.omac.org/>
- Ong F.R. et al. (1999) Detection of drill bit breakthrough for the enhancement of safety in mechatronic assisted orthopaedic drilling. *Mechatronics*, **9**, 565-588.
- Open Group, (2001) The Open Group Architectural Framework V.7.
- Ortamaier T. et al. (2006) Experiments on robot-assisted navigated drilling and milling of bones for pedicle screw placement. *International Journal of Medical Robotics and Computer Assisted Surgery*, **2**, 350-363.
- OSACA Project. (1996) <http://www.osaca.org/>.
- Perdereau V., Drouin M. (1993) A new scheme for hybrid force/position control. *Robotica*, **11**, 453-464.
- Pérez-Turiel J. et al (2004) Application of the SIROCO framework for the specification of a CAOS architecture. *Proc. CAOS 2004 International – Computer Assisted Orthopaedic Surgery*, Chicago (USA), 95-99.
- Pernozzoli J.B., et al. (2000) *A real-time CORBA based system architecture for robot assisted craniofacial surgery*. MMVR.
- Peters H., Raczowsky J. Woern, H. (2005) Approach to an architecture for a generic computer integrated surgery system. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2455- 2460.
- Pujas A. et al. (1993) Hybrid position/force control: Task description and control scheme determination for a real implementation. *Proc. IROS 93*, Yokohama, Japón. 841-846.
- Salisbury K. (1980) Active stiffness control of a manipulator in cartesian coordinates. *Proc. 19th IEEE Conf. on decision and control*, Albuquerque. 95-100.
- Sallinen M. (2004) Hardware and software architecture for rapid and safety use of robots systems. *1st IFAC Symposium on Telematics Applications in Automation and Robotics*. Helsinki, 143 - 147.
- Schauer D., Hein A., Leuth T.C. (2003) Dynamic force control for aminiaturised medical robot system. *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 1090-1095.
- Taylor R.H. (2001) Computer-Integrated Surgery: Coupling Information to Action in 21st Century Medicine. *ICRA 2001*, Plenary Speech.

- Taylor R.H., Stoianovic D. (2003) Medical robotics in computer integrated surgery. *IEEE Tran. on Robotics and Automation*, **19**(5),765-781,.
- Volpe R., Khosla P. (1993) A theoretical and experimental investigation of explicit force control strategies for manipulators. *Proc. IEEE Transaction on Automatic Control*, **38** (11), 1624-1650.
- Wiggins K.L., Malkin S. (1976) Drilling of Bone. *Journal of Biomechanics*, **9**, 553-559.
- Yen J. (2000) *Open, modular architecture controls and the impact of softSERCANS*. General Motors Powertrain.
- Yoshikawa T. (2000) Force control of robots manipulators. *Proc. IEEE ICRA*, San Francisco, CA. 220-226.