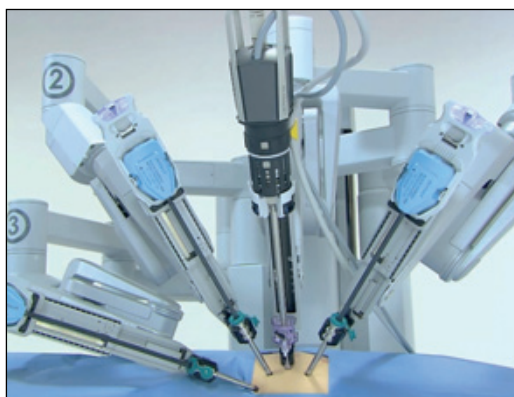


# Nuevo asistente robótico para cirugía: Arquitectura y algoritmos de control



Javier Melo-Urbe  
Álvaro Bertelsen-Simonetti  
Diego Borro-Yáguez  
Emilio Sánchez-Tapia

Ingeniero Electrónico  
Ingeniero Electricista  
Dr. Ingeniero Industrial  
Dr. Ingeniero Industrial

CENTRO DE ESTUDIOS E INVESTIGACIONES TÉCNICAS DE GUIPÚZCOA (CEIT), Paseo de Manuel Lardizábal, 15 – 20018 San Sebastián. Tfno: +34 943 212800. [jmelo@ceit.es](mailto:jmelo@ceit.es).

Recibido: 03/04/2012 • Aceptado: 09/07/2012

DOI: <http://dx.doi.org/10.6036/5011>

## NEW ROBOTIC SURGICAL ASSISTANT: ARCHITECTURE AND CONTROL ALGORITHMS

### ABSTRACT

- This work presents the control system strategy implemented in a new cooperative human-robot system for transpedicular fixation, a type of spine surgery consisting in the immobilization of two or more vertebrae by means of screws and metal bars. The prototype uses a PA-10 robotic arm. The main parts of the control strategy are the admittance module and the generation of virtual fixtures (active constraints) that assist the surgeon and prevent contact with surrounding critical areas. The virtual fixtures are obtained directly from the surgical planner with the purpose of increasing the precision in screw insertion and having safer interventions, reducing radiation doses, invasiveness and the probability of error. Differing from other surgery robotic assistants, the one proposed in this work offers a larger workspace and a degree of versatility that permits its adaptation to different types of surgeries.
- **Keywords:** Admittance control, virtual fixture, force control, cobot, collaborative robot, surgical robotics, transpedicular fixation.

### RESUMEN

Este trabajo presenta la estrategia de control implementada en un nuevo sistema robótico de asistencia para cirugía de fijación transpedicular, la cual es un tipo de cirugía en la espina dorsal que consiste en la inmovilización de dos o más vértebras por medio de tornillos y barras metálicas. El prototipo presentado emplea un brazo robótico industrial PA-10, para la validación de los algoritmos planteados. La estrategia de control posee dos módulos principales: un primer módulo en el que se ha implementado un controlador en admitancia y un segundo módulo para la generación de restricciones virtuales de movimiento, las cuales asisten al cirujano y previenen que éste entre en contacto erróneamente con estructuras críticas del paciente. Con el propósito de mejorar la precisión en la implantación de los tornillos pediculares, se emplea un planificador quirúrgico con el cual el cirujano puede simular el

procedimiento y luego transferir al robot la información obtenida en la planificación, para generar en base a esta las restricciones virtuales de movimiento. A diferencia de otros sistemas de asistencia quirúrgica, el sistema propuesto en este trabajo posee un alto grado de versatilidad y un amplio espacio de trabajo, lo cual permite su futura adaptación a otros tipos de cirugía.

**Palabras clave:** Control en admitancia, restricción virtual de movimiento, control en fuerza, cobot, robot colaborativo, cirugía robótica, fijación transpedicular.

### 1. INTRODUCCIÓN

En este trabajo se presenta la estrategia de control en admitancia y los algoritmos para la generación de restricciones virtuales de movimiento, implementados en un Sistema de asistencia robotizada para el procedimiento quirúrgico de fijación transpedicular. La fi-

jación transpedicular es un tipo particular de cirugía en la espina dorsal y consiste en la inmovilización de múltiples vértebras mediante el uso de tornillos y barras metálicas [1]. Dichos tornillos son insertados a través de los pedículos de las vértebras y fijados entre sí a través de barras metálicas, dando de esta forma estabilidad mecánica a la espina como se muestra en la Fig. 1.

La fijación transpedicular se realiza para prevenir daños mayores en vértebras afectadas por fracturas, tumores o enfermedades degenerativas, que provocan la pérdida de la estabilidad del conjunto óseo de la espina dorsal.

La colocación errónea de los tornillos pediculares no es algo inusual. Según diferentes estudios la tasa de error en la implantación de los tornillos es bastante variable aunque puede llegar a alcanzar valores cercanos al 30%. De igual forma se ha demostrado que esta cifra puede reducirse significativamente empleando Sistemas de Cirugía Asistida por Ordenador (CAS) [2], [3]. Tjardes *et al.* [4] presentan una completa descripción de los avances conseguidos en el campo de la cirugía de espina asistida (guiada) por ordenador, que sirve de marco de referencia para poner en contexto los diferentes aspectos del trabajo presentado.

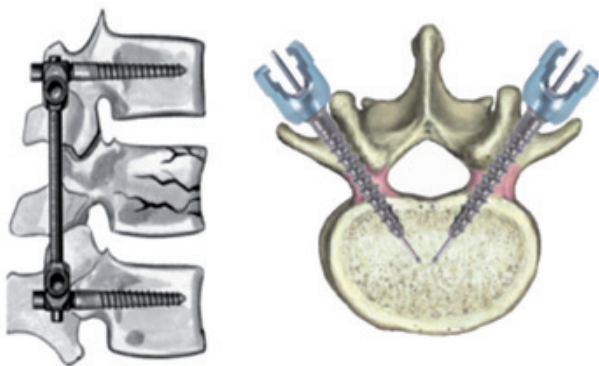


Fig. 1: Fijación de la espina dorsal mediante la implantación de tornillos pediculares

En el presente artículo se propone un sistema robótico de asistencia diseñado para trabajar de forma cooperativa junto con el cirujano, sosteniendo el instrumental quirúrgico en la posición y orientación adecuada para realizar la inserción de los tornillos pediculares. De esta forma, el robot tan sólo se ocupa de facilitar la posición y orientación desde la cual el médico debe iniciar la inserción de los tornillos, quedando

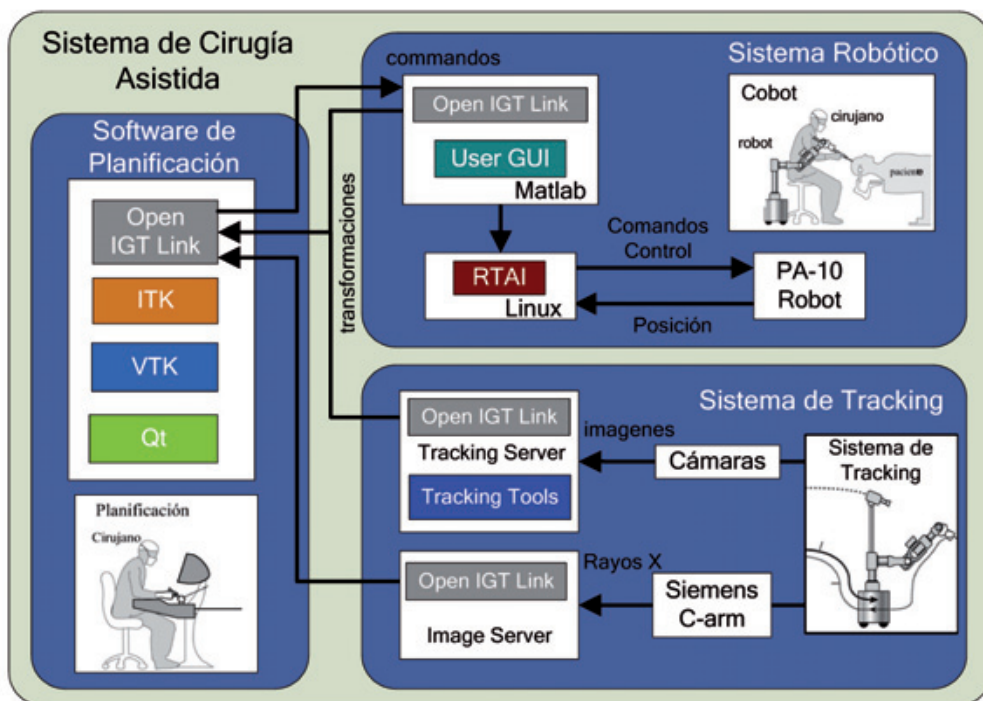


Fig. 2: Arquitectura del Sistema de cirugía asistida propuesta

toda la responsabilidad de la ejecución de la cirugía a cargo del médico.

El sistema consta de tres partes fundamentales (ver Fig. 2): un software de planificación, un sistema de navegación y un dispositivo robótico (robot industrial PA-10 de Mitsubishi). En primera instancia, el cirujano debe realizar la planificación de la cirugía utilizando para ello el software de planificación quirúrgica desarrollado, mediante el cual se determina la ubicación óptima de inserción de los tornillos, basándose para ello en imágenes pre-operatorias de TC (acrónimo de *Tomografía Computarizada*, una técnica de imagen médica) y una reconstrucción 3D de la columna vertebral del paciente.

En el segundo paso (ya dentro del quirófano) es necesario alinear geoméricamente (proceso de registro) la planificación pre-operatoria realizada con la ubicación actual del paciente, la cual es obtenida a partir de un número reducido de radiografías intra-operatorias. Finalmente, una vez realizado el registro de imágenes, es posible comandar al robot para que este se aproxime a las vértebras objetivo, reproduciendo de forma precisa la posición y orientación de los tornillos (seleccionados desde el planificador) y evitando el contacto con cualquier estructura crítica vecina.

Es necesario aclarar que el trabajo presentado se limita a un prototipo de validación de concepto y no a un sistema quirúrgico definitivo. En el artículo además se presenta una descripción general del sistema, haciendo énfasis en los algoritmos y estrategias de control implementadas, para finalmente, presentar los resultados y conclusiones obtenidas.

## 2. DESCRIPCIÓN DE LA ESTRATEGIA DE CONTROL IMPLEMENTADA

El concepto general que se ha implementado en este proyecto es el de *Sistema Colaborativo Hombre-Máquina* (también referenciado como “COBOT”, acrónimo del inglés *Collaborative roBOT*), el cual permite amplificar o asistir las habilidades del cirujano, mientras este ejecuta tareas de alta complejidad donde se requiere conjugar la capacidad de discernimiento del humano y la precisión que aporta el robot [5].

En este artículo se presenta un sistema COBOT que puede ser desplazado libremente por el cirujano, quien sujeta el robot por su extremo final, orientándolo y posicionándolo en el espacio de trabajo sin sentir oposición o resistencia al movimiento ejecutado. Sin embargo, en la proximidad de estructuras críticas (terminaciones nerviosas, vasos sanguíneos), el robot puede limitar el movimiento del cirujano previniendo que este acceda a zonas que pongan en riesgo la salud del paciente. A esta limitación activa de movimiento se le ha denominado “*Restricciones Virtuales*” y permiten acotar el espacio de trabajo del cirujano y aumentar la precisión y repetibilidad e los movimientos que este efectúa durante su labor [6].

A continuación se presenta el esquema de control implementado en el sistema COBOT. Dicho esquema está compuesto por tres bloques independientes (ver Fig. 3): un módulo que representa el controlador de velocidad interno que posee el robot *Mitsubishi PA-10*, un controlador de admitancia adaptativo, y por último un módulo para la generación de las restricciones virtuales.

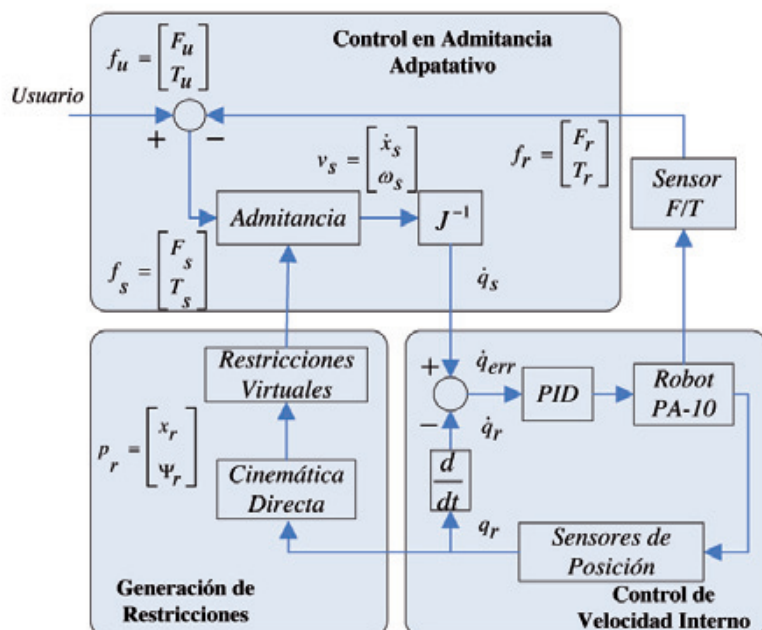


Fig. 3: Esquema de control implementado en el que se incluye un módulo para la generación de restricciones virtuales de movimiento

## 2.1. CONTROLADOR DE VELOCIDAD

Este primer modulo corresponde a un lazo de control en velocidad impuesto directamente por el controlador del robot PA-10. Dicho controlador recibe como entrada comandos de velocidad y retorna como salida la posición real del robot. En nuestro caso el robot PA-10 recibe referencias de velocidad en el espacio articular ( $\dot{q}_s$ ) y retorna así mismo la posición real de cada articulación del robot ( $q_r$ ). Es posible ajustar los parámetros del controlador PID que se encuentran en el lazo de control, aunque para este caso en particular se han empleado los valores por defecto definidos por el fabricante.

## 2.2. MÓDULO DE CONTROL EN ADMITANCIA

Este modulo corresponde a la clásica implementación de un controlador en fuerza, normalmente aplicado a robots cuya estructura es no reversible. Está basado en el concepto tradicional de control en admitancia introducido por [7], [8], [9] en donde se toman como referencia de entrada la fuerza ( $F_u$ ) y el par ( $T_u$ ) aplicado por el usuario, sumados a la fuerza ( $F_r$ ) y par ( $T_r$ ) generados por la interacción entre el robot y su entorno. Como salida, este módulo genera las velocidades en espacio articular ( $\dot{q}_s$ ) que posteriormente se envían al controlador interno del robot descrito anteriormente.

### 2.2.1. Control adaptativo de la admitancia

En el desarrollo de interfaces cooperativas hombre-robot resulta habitual el uso de controladores de admitancia para efectuar el control en fuerza del robot. Sin embargo, es necesario tener en cuenta dos situaciones que se presentan durante la interacción entre el usuario y el robot que pueden llegar a afectar considerablemente la respuesta del controlador, en particular cuando se pretende realizar tareas en las que se requiere posicionar el robot con alta precisión: En primera instancia se deben tener en cuenta los movimientos involuntarios del usuario (“tremor” o temblor en las manos), que introducen un error en la referencia de posición del controlador. El segundo factor a tener en cuenta es la incidencia tanto de la rigidez impuesta por el usuario (en este caso la rigidez del conjunto brazo-mano), como también del tiempo de retraso en la respuesta del robot ante un cambio en la referencia de entrada, sobre la estabilidad del controlador [10].

Es importante resaltar que los factores mencionados tienen una incidencia mayor cuando se ejecutan tareas de posicionamiento fino (preciso), en comparación con desplazamientos continuos de mayor longitud. De esta forma se justifica la implementación de un controlador adaptativo, que ajuste los parámetros de la admitancia (masa y/o viscosidad), con el fin de garantizar la estabilidad del sistema independientemente del tipo de movimiento que ejecute el usuario.

De este modo se emplea el modelo clásico de admitancia, con el que es posible calcular la velocidad  $v_s$  del extremo final del robot a partir de la fuerza/par  $f_s$  ejercida por el usuario, como queda expresado en la siguiente ecuación:



$$M\dot{v}_s + Bv_s = f_s \quad (1)$$

En donde,  $M$  (kg) y  $B$  ( $\frac{Ns}{m}$ ) equivalen a la masa y la viscosidad virtual respectivamente.  $f_s$  representa la fuerza y par aplicados a la entrada de la admitancia. De igual forma  $v_s$  representa la velocidad lineal ( $\dot{x}_s$ ) y la velocidad angular ( $\omega_s$ ) del extremo final del robot, siendo  $v_s$  así mismo, la aceleración tanto lineal como angular en dicho punto. De esta manera es posible expresar la ecuación general (1) en su respectiva representación matricial como sigue:

$$\begin{bmatrix} m & 0 \\ 0 & m_i \end{bmatrix} \begin{bmatrix} \ddot{x}_s \\ \ddot{\omega}_s \end{bmatrix} + \begin{bmatrix} b & 0 \\ 0 & b_i \end{bmatrix} \begin{bmatrix} \dot{x}_s \\ \dot{\omega}_s \end{bmatrix} = \begin{bmatrix} F_s \\ T_s \end{bmatrix} \quad (2)$$

Teniendo en cuenta que las componentes  $m$ ,  $m_i$  (masa virtual) y  $b$ ,  $b_i$  (viscosidad virtual) son matrices diagonales (3 x 3). Del mismo modo, las aceleraciones  $\ddot{x}_s$  ( $\frac{m}{s^2}$ ),  $\ddot{\omega}_s$  ( $\frac{rad}{s^2}$ ), las velocidades  $\dot{x}_s$  ( $\frac{m}{s}$ ),  $\dot{\omega}_s$  ( $\frac{rad}{s}$ ) y la fuerza/par  $F_s$  (N),  $T_s$  (Nm) son vectores (3x1).

Ahora bien, como ya se mencionó anteriormente la alta rigidez impuesta por el usuario hace que el sistema se vuelva inestable, sobre todo en tareas de posicionamiento fino en las que no solo aumenta la rigidez del usuario, sino que también disminuye la velocidad de desplazamiento del extremo del robot. Ante tales condiciones, el factor de amortiguamiento viscoso tiene un efecto menor sobre el comportamiento del sistema, siendo necesario ajustar dicho factor para evitar que el sistema se vuelva inestable. Por esta razón, se aplica una estrategia para el ajuste de los coeficientes de viscosidad de la admitancia de acuerdo con la rigidez estimada del usuario  $K_u$ , quedando expresada la ecuación general de la admitancia (1) de la siguiente forma:

$$M\dot{v}_s + \alpha K_u v_s = f_s \quad (3)$$

En donde  $c$  y  $c_r$  ( $c, c_r \in [0,1]$ ), es una constante proporcional (adimensional) a la cual se le ha asignado un valor  $K_u$  de 0.3 en las pruebas realizadas (valor experimental). corresponde a la rigidez estimada del usuario, la cual se calcula empleando una aproximación experimental propuesta en [11]. Dicha estimación tiene en cuenta la variación en la fuerza aplicada por el usuario  $\Delta f_s$ , con respecto a la variación en la posición del extremo final del robot como se expresa en la siguiente ecuación:

$$K_u = \frac{\Delta f_s}{\Delta p_r} \quad (4)$$

Finalmente, una vez calculada la velocidad  $v_s$  en el espacio Cartesiano, es necesario traducir sus vectores componentes  $\dot{x}_s$  (velocidad lineal) y  $\omega_s$  (velocidad angular) al espacio articular  $\dot{q}_s$  (referencia de entrada del controlador de velocidad interno) utilizando para ello la inversa de la matriz Jacobiana como se muestra en la Fig. 3.

### 2.3. MÓDULO DE RESTRICCIONES VIRTUALES

El uso de restricciones virtuales permite limitar el movimiento del robot dentro de un área específica,

asegurando que el dispositivo no pueda ingresar en zonas que han sido definidas como prohibidas. La cirugía de fijación transpedicular resulta adecuada para aplicar el concepto de restricción virtual de movimiento, debido a que el cirujano tiene que insertar tornillos a través de estructuras de diámetro reducido (pedículos vertebrales) y que a su vez se encuentran rodeadas por la espina dorsal, terminaciones nerviosas y vasos sanguíneos que pueden ser lesionados si la inserción no se realiza con la precisión adecuada [12].

Con el fin de implementar las restricciones virtuales, el espacio de trabajo ha sido dividido en dos sub-espacios complementarios. El primero de ellos contiene las direcciones definidas como “preferentes” o permitidas y el segundo las direcciones “no preferentes” o prohibidas. Por consiguiente, una restricción virtual será el resultado de combinar direcciones preferentes y no preferentes según sea el caso. La metodología empleada para definir restricciones virtuales que se ha utilizado en este trabajo, está basada en los algoritmos propuestos por Bettini *et al.* [6] y posteriormente implementados por Marayong *et al.* [13].

El método empleado consiste en filtrar la señal de fuerza/par (aplicada por el usuario), teniendo en cuenta si la dirección de sus respectivas componentes pertenece al conjunto de direcciones definidas ya sea en el sub-espacio preferente o en el no preferente. De esta forma se define la matriz  $D$   $6 \times n$  (donde  $1 \leq n \leq 6$ ) que representa las direcciones instantáneas preferentes en el espacio, en términos de posición y orientación. A partir de la matriz  $D$  se calculan las matrices (operadores) de proyección ortogonal sobre las direcciones preferentes de movimiento  $[D]$ , y sobre las direcciones no preferentes  $\langle D \rangle$ , como sigue:

$$\begin{aligned} [D] &= D(D^T D)^{-1} D^T \\ \langle D \rangle &= I - [D] \end{aligned} \quad (5)$$

En donde  $[D]$  es el operador de proyección ortogonal de las direcciones preferentes y  $\langle D \rangle$  corresponde al operador de las direcciones no preferentes (en las que se restringirá el movimiento). De esta manera el modelo de admitancia aplicado queda definido como sigue:

$$M\dot{v}_s + \alpha K_u v_s = c([D] + c_r \langle D \rangle) f_s \quad (6)$$

En donde  $c$  y  $c_r$  ( $c, c_r \in [0,1]$ ) y son coeficientes que ajustan el valor de actuación (ponderación) tanto de las direcciones preferentes como de las no preferentes. De esta manera si  $c, c_r = 1$ , no existe restricción alguna en cualquier dirección del espacio (movimiento libre). Pero si se mantiene el valor de  $c$  y se ajusta  $c_r = 0$ , el robot sólo se moverá sobre las direcciones preferentes. Con lo cual, modificando los valores de  $c$  y  $c_r$  es posible atenuar o incrementar el grado de restricción aplicado en una dirección del espacio dada.

Una vez definidos los operadores de proyección ortogonal, deben imponerse las condiciones de contorno que delimitarán el espacio de trabajo, para así generar adecuadamente las restricciones virtuales.

### 2.3.1. Generación de restricciones virtuales en el espacio

A continuación se explicará en detalle la particularización del algoritmo descrito en los anteriores párrafos, para la generación de una zona de restricción virtual de tipo cilíndrico. El interés de generar una restricción virtual cilíndrica se basa en el hecho de que el cilindro se asume como una de las primitivas geométricas a partir de la cual se pueden generar otros casos más generales de restricción de movimiento, tales como el movimiento a lo largo de una línea o sobre un plano determinado. El algoritmo presentado permite generar una superficie virtual que restringe el movimiento del robot dentro de un área específica, pero permite la libertad de movimiento cuando la posición del robot se encuentra dentro de los límites definidos. De igual forma, las mejoras introducidas en la definición de las condiciones de contorno, hacen posible transmitir al usuario la sensación háptica de estar colisionando con una superficie virtual, con lo cual se mejora la forma de interacción entre el usuario y el dispositivo robótico principalmente en tareas de guiado y posicionamiento fino.

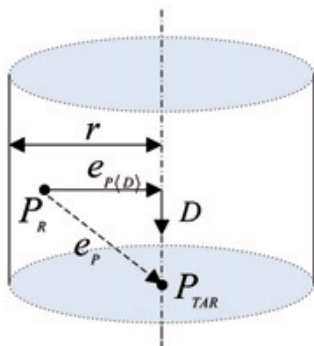


Fig. 4: Definición de las condiciones de contorno para la generación de una restricción virtual cilíndrica

Se define  $D = [d_x \ d_y \ d_z \ 0 \ 0 \ 0]^T$  el radio como la distancia perpendicular entre el eje central del cilindro y la región de frontera (ver Fig. 4). El eje del cilindro se define como un vector unitario de coordenadas cartesianas, a partir del cual se conforma la matriz de direcciones preferentes empleada en (5). En este caso en particular solo se dan los valores de que definen una restricción en traslación, pues en el caso de una restricción de movimiento dentro de un espacio acotado, no tiene sentido práctico restringir también las orientaciones.

Se define  $P_{TAR}$  como el punto objetivo sobre el eje  $D$  especificado anteriormente y a partir del cual se calculan los operadores de proyección ortogonal sobre la dirección preferente de movimiento  $[D]$  y sobre la dirección no preferente  $\langle D \rangle$  como se hizo en (5).

Con el fin de calcular la distancia entre la posición actual del extremo del robot  $P_r$  y el eje central  $D$ , se calcula la proyección del vector de error como  $e_{P(D)}$  sigue:

$$\begin{aligned} e_{P(D)} &= \langle D \rangle (P_{TAR} - P_R) \\ &= \langle D \rangle e_P \end{aligned} \quad (7)$$

En donde  $e_{P(D)} \cdot f_s > 0$ , con lo cual es un vector que está incluido dentro del sub-espacio de direcciones no preferentes  $\langle D \rangle$ .

Es necesario ahora precisar que el robot adoptará dos diferentes comportamientos, dependiendo de la ubicación de su extremo final en el espacio de trabajo. De esta forma si el extremo final del robot se encuentra dentro del área acotada por la restricción virtual, esto es  $\|e_{P(D)}\| < r$ , el robot tendrá libertad de movimiento en todas las direcciones del espacio, teniendo en cuenta que  $(c, c_r = 1)$  de acuerdo con la ecuación de la admitancia definida en (6). Cuando el robot está en la región de frontera, esto es  $\|e_{P(D)}\| \geq r$ , la admitancia deber ser modificada para permitir al usuario moverse a lo largo del contorno de dicha región, manteniendo el extremo del robot dentro del área de restricción. Para tal fin, se restituye una fuerza/par estable, que permite al usuario percibir la sensación háptica del contacto con la pared virtual, sin llegar a traspasarla [14].

Para el cálculo de la fuerza/par que será restituida al usuario, son necesarios los operadores de proyección ortogonal del vector de error  $e_{P(D)}$ . Estos operadores de proyección son  $[E]$  y  $\langle E \rangle$ , los cuales se obtienen a partir de la ecuación (5), substituyendo  $E = e_{P(D)}$ . De esta forma la fuerza/par que se restituye queda definida como:

$$f_F = \begin{cases} \langle E \rangle f_s, & e_{P(D)} \cdot f_s > 0 \\ f_s, & \text{cualquier otro} \end{cases} \quad (8)$$

En donde el primer caso  $e_{P(D)} \cdot f_s > 0$ , se cumple cuando el usuario intenta traspasar la restricción virtual  $f_F$  es la fuerza/par “filtrada”, en la cual sólo se considera aquella componente de fuerza/par cuya dirección sea la que minimice la distancia entre el robot y el eje del cilindro. En el caso en que una fuerza/par sea perpendicular a la restricción virtual (es decir, que empuje al robot a salirse del contorno definido), esta será filtrada o eliminada. En cualquier otro caso el robot tendrá libertad de movimiento en cualquier dirección, es decir, la fuerza aplicada por el usuario se transmite tal cual al controlador en admitancia.

Con el fin de que el dispositivo esté siempre confinado dentro de la región de restricción definida, se agrega un muelle virtual para prevenir que el módulo del error  $\|e_{P(D)}\|$  llegue a ser mayor que el radio  $r$ . De esta forma se agrega una componente o término de “impedancia” dentro de la ley general de admitancia descrita hasta este punto. Finalmente, la ecuación general para la generación de una restricción virtual de movimiento de tipo cilíndrico queda expresada como sigue:

$$M\dot{v}_s + \alpha K_u v_s = c([E] + c_r \langle E \rangle) f_s + k(e_{P(D)} - \hat{e}_{P(D)} r) \quad (9)$$

En donde  $k$  es la rigidez asociada al muelle virtual implementado. Para este caso  $c, c_r = 1$  y  $\hat{e}_{P(D)}$  denota el vector unitario del error  $e_{P(D)}$ .

### 3. PROTOTIPO EXPERIMENTAL

El robot empleado en este prototipo experimental es un PA10-7C (*Mitsubishi Heavy Industries Ltd., Kobe, Japón*) un brazo robótico industrial con 7 GdL (grados de libertad), el cual ofrece altos niveles de precisión y destreza debido a la redundancia articular que posee. Además de esto, el amplio espacio de trabajo que se puede abarcar con este dispositivo permite acceder a las diferentes secciones de la espina dorsal del paciente, lo cual marca diferencia con otros sistemas como *SpineAssist* [15] cuyo reducido espacio de trabajo resulta una desventaja funcional del sistema. Se ha instalado un sensor de fuerza *mini45* (*ATI Industrial Automation, NC, USA*) en el extremo final del robot, con el cual es posible medir las fuerzas aplicadas por el usuario, o del mismo modo determinar si el dispositivo ha colisionado con el entorno (ver Fig. 5).



Fig. 5: Prototipo experimental

#### 3.1. SOFTWARE DE CONTROL

La plataforma para la ejecución en tiempo-real de los algoritmos de control del robot es *RTAI-Linux* (*Real Time Application Interface*), una extensión del sistema operativo *Linux* que permite utilizar tarjetas genéricas de adquisición de datos y emplear lenguajes de programación de alto nivel a un bajo coste, pero dentro de los estándares industriales [16].

*RTAI-Linux* es una modificación de la arquitectura estándar de *Linux*, que permite añadir un planificador de procesos de tiempo-real. Esta modificación se consigue instalando un *micro-kernel* por encima de la estructura del *kernel* del *Linux* clásico. De este modo, en un sistema *RTAI-Linux*, el *micro-kernel* “modificado” es el encargado de gestionar por un lado cada proceso en tiempo real, y por otro, el *kernel* estándar de *Linux* que se convierte en un proceso adicional más a ejecutar, pero con baja prioridad. Por esta razón, una tarea que se ejecute en “no” tiempo-real, no interferirá (retrasará) con las tareas destinadas al tiempo-real (alta prioridad).

*RTAI-Linux* puede ser utilizado en ordenadores de arquitectura estándar x86 provistos con el sistema operativo *Linux*, en combinación con un software de diseño asistido (en el caso descrito aquí, *Matlab/Simulink*). De esta forma

es posible generar código para ser ejecutado en tiempo-real, ya sea como una tarea simple o en configuración multi-tarea dependiendo de los requerimientos del usuario. Finalmente, se ha establecido una frecuencia de muestreo para el lazo de control de 333 Hz (frecuencia máxima a la que se puede comunicar el controlador del robot PA-10), ejecutándose en un ordenador con procesador *Intel Pentium4* de 3.2GHz.

#### 3.2. EXPERIMENTOS Y RESULTADOS

Empleando los algoritmos presentados anteriormente en el apartado 2.2, se acotó el espacio de trabajo del robot dentro de los límites de una superficie virtual cilíndrica (restricción virtual) de radio  $r = 0,08 \text{ m}$ . En el experimento el usuario debe tomar el robot por su extremo final y guiarlo manualmente (tarea cooperativa) a lo largo del contorno de la restricción virtual impuesta. El usuario repitió en dos ocasiones la tarea, con el objetivo de comparar la respuesta del sistema cuando se aplica el control adaptativo propuesto (Fig.7) con respecto a la respuesta obtenida con un controlador convencional de admitancia (Fig.6).

- **Controlador convencional de admitancia:** En este caso los valores de masa  $M$  y viscosidad  $B$  del controlador en admitancia, permanecen constantes durante el desarrollo de la tarea. El valor de la masa es fijado en  $M = 2,5 \text{ kg}$ , mientras que a la viscosidad se le asignó un valor de  $B = 35 \frac{\text{Ns}}{\text{m}}$ . Es necesario aclarar que estos valores se han impuesto de forma experimental y son aplicados por igual en cada una de las componentes del movimiento en traslación ( $x, y, z$ ). Como ya se había comentado anteriormente, en este caso no se han tenido en cuenta las orientaciones. La respuesta del sistema aplicando el control de admitancia convencional se muestra en la Fig.6.
- **Controlador adaptativo de admitancia:** Para este caso se ha fijado el valor de la masa en  $M = 2,5 \text{ kg}$ , valor que permanece constante durante el desarrollo de la tarea. La viscosidad  $B$  se calcula como un valor proporcional a la rigidez estimada del usuario  $k_u$ , en donde  $B = \alpha K_u$  como se definió anteriormente en la ecuación (4). El valor calculado de la viscosidad  $B$  se restringe dentro de un rango de valores definido experimentalmente, tal que  $35 \leq B \leq 100 \frac{\text{Ns}}{\text{m}}$ . Es necesario aclarar que para cada componente de movimiento ( $x, y, z$ ) se calcula su valor de viscosidad correspondiente. La respuesta del sistema aplicando el control de admitancia adaptativo se muestra en la Fig.7.

##### 3.2.1. Resultados

Como se muestra en la Fig. 5, el usuario sujeta el robot por su extremo final, guiándolo a lo largo del contorno de la restricción virtual cilíndrica impuesta. Los resultados de la trayectoria seguida por el robot, las medidas de fuerza en el extremo del robot, el valor de viscosidad aplicado a la admitancia y la rigidez estimada del usuario, son presentados en las Fig.6 y Fig.7 para los casos del controlador convencional y el controlador adaptativo en admitancia



respectivamente. Es necesario aclarar que sólo se presentan los resultados obtenidos sobre el eje “y”, con el fin de no extender el análisis realizado. De igual forma se presentan en la gráficas (b) y (c) de cada figura, los resultados obtenidos durante un lapso de tiempo de 8 s en un tramo específico de la trayectoria seguida.

Cuando la tarea se realiza utilizando el controlador convencional (masa y viscosidad constantes), el robot comienza a oscilar (vibrar) cuando entra en contacto con la pared virtual. Aunque la trayectoria seguida por el robot se ajusta al contorno de la restricción virtual Fig.6 (a), la gráfica de posición refleja el comportamiento inestable (oscilatorio) que se obtiene con la implementación de este tipo de controlador, aún más, es posible inferir que el comportamiento inestable se amplifica debido a que el usuario intenta realizar una compensación instantánea de las oscilaciones del robot, aplicando una fuerza de sentido contrario al movimiento del robot. Este efecto queda reflejado en la gráfica de fuerza Fig.6 (b), donde se aprecia

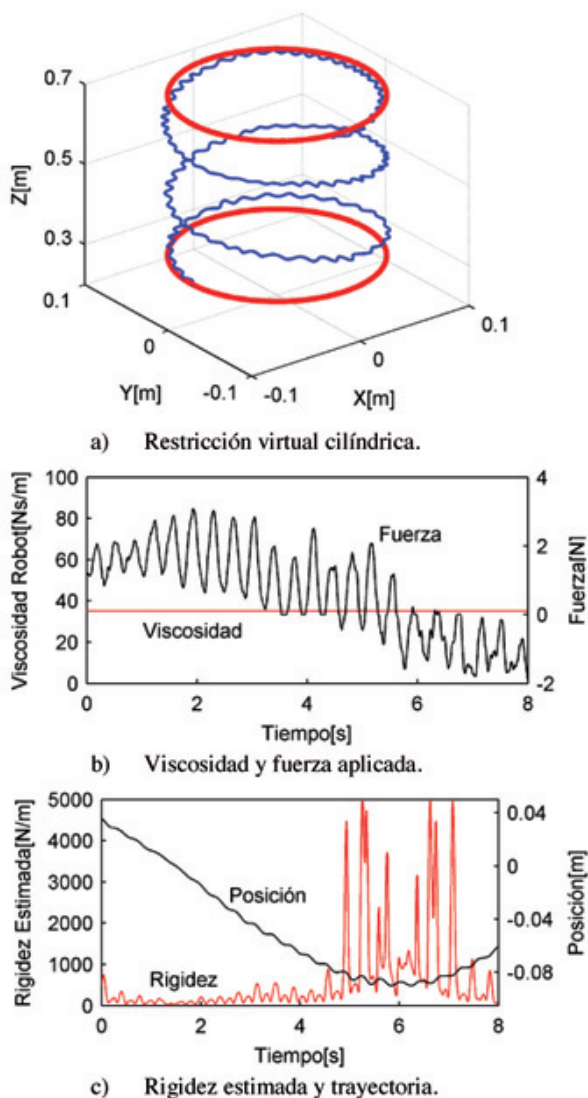


Fig. 6: Resultados obtenidos con el controlador tradicional en admitancia

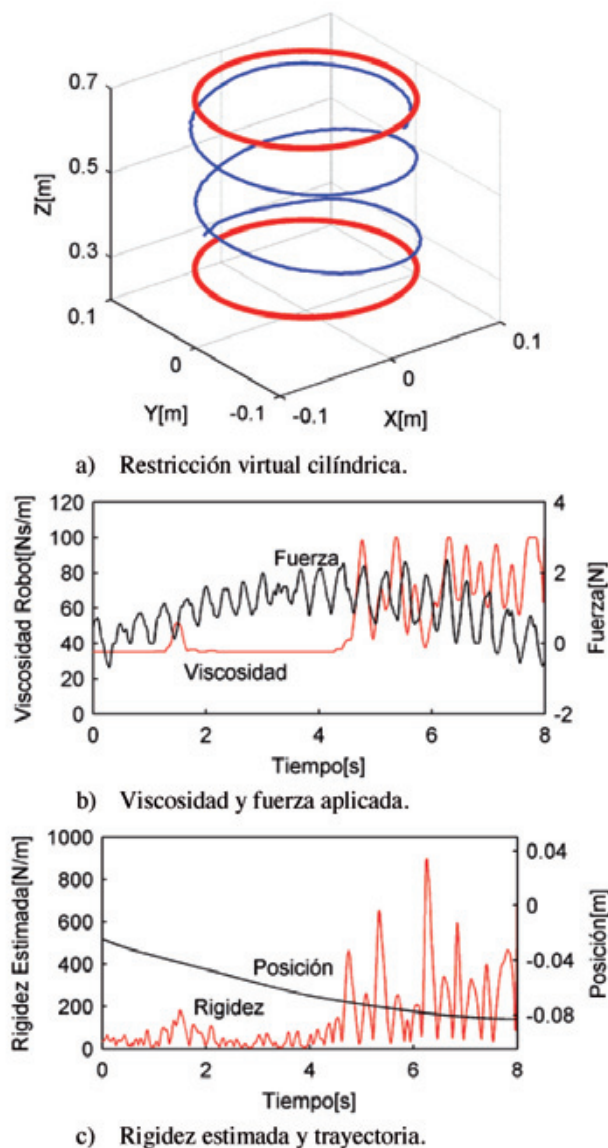


Fig. 7: Resultados obtenidos con el control adaptativo en admitancia

que la fuerza medida oscila, llegando a tener un valor de amplitud máxima de 3,5 N.

Se puede, a partir de esto, explicar el aumento de la rigidez estimada del usuario Fig.6 (c), como consecuencia de la corrección (compensación) que intenta hacer el usuario. Cuando el usuario trata de compensar las oscilaciones que se presentan, aumenta instantáneamente la rigidez de su brazo y del mismo modo aplica una fuerza mayor sobre el extremo final del robot. De hecho como se observa en la Fig.6 (c), la zona en la que existe mayor oscilación en la trayectoria, se corresponde con la zona en donde la rigidez estimada alcanza su máxima amplitud ( $5000 \frac{N}{m}$ ).

En el segundo experimento se emplea el controlador adaptativo, con el cual se calcula en cada instante la viscosidad aplicada a la admitancia. En este caso el comportamiento del robot es estable cuando este entra en contacto con la pared virtual Fig.7 (a). Sin embargo en la gráfica de fuerza Fig.7

(b) se observa que la señal igualmente oscila, aunque con una menor amplitud ( $2.2\text{ N}$ ) en comparación a los valores obtenidos con el controlador convencional. Si se observa la Fig.7 (c), la trayectoria seguida por el robot no presenta mayores variaciones, y reproduce fielmente el contorno de la pared virtual que delimita el espacio de trabajo.

La rigidez estimada del usuario tiene un valor máximo de  $920 \frac{\text{N}}{\text{m}}$ , cuyo efecto sobre la estabilidad del sistema se ve contrarrestado por la acción del controlador adaptativo, que ajusta proporcionalmente la viscosidad de la admitancia de acuerdo con el valor instantáneo calculado de la rigidez estimada del usuario: ver Ecuación (4). De esta forma en la Fig.7 (b) se observa como la viscosidad aumenta hasta un valor máximo de  $100 \frac{\text{Ns}}{\text{m}}$ , coincidiendo con el aumento de la rigidez impuesta por el usuario.

#### 4. CONCLUSIONES Y TRABAJO FUTURO

Este artículo presenta el trabajo realizado para diseñar, implementar y evaluar el funcionamiento de las restricciones virtuales de movimiento impuestas a un *robot colaborativo* (COBOT) dedicado a la asistencia del cirujano en procedimientos de cirugía de fijación transpedicular. No obstante, los algoritmos de control son suficientemente generales como para considerar la aplicación de este tipo de asistencias en otros procedimientos quirúrgicos.

Se ha presentado un algoritmo para el establecimiento de restricciones virtuales de movimiento, el cual permite *renderizar* una superficie virtual de contacto (pared virtual) que acota el espacio de trabajo y que retroalimenta al usuario las fuerzas generadas como producto de la interacción (contactos) efectuada. Con dicho algoritmo es posible transmitir al usuario la sensación háptica de un contacto virtual, facilitando la ejecución de tareas fundamentales en el procedimiento de fijación transpedicular, como lo son el seguimiento y guiado del dispositivo robótico a lo largo de trayectorias predefinidas.

Los resultados experimentales presentados en este trabajo, demuestran que la aplicación de un controlador de admitancia adaptativo permite mejorar la interacción usuario/robot, cuando se desempeñan tareas colaborativas. Es necesario aclarar, que tan sólo se ha realizado una evaluación preliminar del concepto propuesto, y que es necesario realizar una completa campaña de experimentos que involucre al personal médico, para de esta manera poder validar el funcionamiento y aplicabilidad del dispositivo.

Tal y como se adelantó al inicio del presente apartado, este estudio abre un camino para el ensayo y evaluación del sistema presentado en un caso clínico real en un futuro cercano, teniendo en cuenta además que el robot y la arquitectura planteada es bastante genérica, con lo cual el sistema propuesto podría ser adaptado a otras aplicaciones médicas, como cirugías ortopédicas en pelvis o fémur. El sistema presentado, así como también los algoritmos de control se encuentran aún en fases preliminares de desarrollo,

con lo cual es necesario continuar su evolución haciendo énfasis en factores como la seguridad y la ergonomía que son determinantes para que el sistema tenga una adecuada aceptación por parte del personal médico.

#### 5. BIBLIOGRAFÍA

- [1] Kabins M.B and Weinstein J.N. "The History of Vertebral Screw and Pedicle Screw Fixation". The Iowa Orthopaedic Journal, 11, 1991, p 127.
- [2] Holly LT, "Image-guided spinal surgery". The International Journal of Medical Robotics and Computer Assisted Surgery, Vol.2, sup 1, 2006, p. 7-15. DOI: <http://dx.doi.org/10.1002/rcs.69>.
- [3] Richter M, Cakir B, and Schmidt R. "Cervical pedicle screws: conventional versus computer-assisted placement of cannulated screws". Spine, 30, 2005, p 2280.
- [4] Tjardes T, Shafizadeh S, Rixen D, et al. "Image-guided spine surgery: state of the art and future directions". European Spine Journal, Vol. 19, 2010, p. 25-45. DOI: <http://dx.doi.org/10.1007/s00586-009-1091-9>.
- [5] Peshkin M, Colgate J, Wannasupphoprasit W, et al. "Cobot architecture," Robotics and Automation, IEEE Transactions, Vol. 17, no. 4, 2001, p. 377-390. DOI: <http://dx.doi.org/10.1109/70.954751>.
- [6] Bettini A, Lang S, Okamura A, et al. "Vision assisted control for manipulation using virtual fixtures: Experiments at macro and micro scales," Robotics and Automation ICRA'02 Proceedings, IEEE International Conference, Vol. 4, 2002, p. 3354-3361. DOI: <http://dx.doi.org/10.1109/ROBOT.2002.1014229>.
- [7] Hogan N, "Impedance control: An approach to manipulation: Part III applications," Journal of dynamic systems, measurement, and control, Vol. 107, no. 2, 1985, p. 17.
- [8] Seraji H, "Adaptive admittance control: An approach to explicit force control in compliant motion," in Robotics and Automation, 1994. Proceedings. IEEE International Conference. 1994, p. 2705-2712. DOI: <http://dx.doi.org/10.1109/ROBOT.1994.350927>.
- [9] Mason M. "Compliance and force control for computer controlled manipulators". Systems, Man and Cybernetics, IEEE Transactions, 1981 Vol.11, p.418 -432. DOI: <http://dx.doi.org/10.1109/TSMC.1981.4308708>.
- [10] Surdilovic D. "Contact stability issues in position based impedance control: Theory and experiments". Proceedings of Robotics and Automation ICRA'96, IEEE International Conference, Vol 2, 1996, p. 1675-1680. DOI: <http://dx.doi.org/10.1109/ROBOT.1996.506953>.
- [11] Tsumugiwa T, Yokogawa R, and Hara K. "Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task". Proceedings of Robotics and Automation ICRA'02, IEEE International Conference. Vol.1, 2002, p. 644-650. DOI: <http://dx.doi.org/10.1109/ROBOT.2002.1013431>.
- [12] Lee J, Kim S, Kim Y, et al. "Automated surgical planning and evaluation algorithm for spinal fusion surgery with three-dimensional pedicle model," Intelligent Robots and Systems (IROS 2011), IEEE International Conference, 2011, p. 2524-2531. DOI: <http://dx.doi.org/10.1109/IROS.2011.6094644>.
- [13] Marayong P, Li M, Okamura A, et al. "Spatial motion constraints theory and demonstrations for robot guidance using virtual fixtures," in Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference, Vol. 2. IEEE, 2003 p. 1954-1959. DOI: <http://dx.doi.org/10.1109/ROBOT.2003.1241880>.
- [14] Melo J y Sánchez E. "Control de bajo coste para dispositivos hápticos". DYNA Ingeniería e industria, 85(3), 2010, p.237-244. DOI: <http://dx.doi.org/10.6036/3018>
- [15] Shoham M, Brink-Danan S, Friedlander A, and Knoller N. "Bone-Mounted Miniature Robotic System for Spine Surgery". Biomedical Robotics and Biomechanics BioRob 2006. IEEE/RAS-EMBS International Conference, 2006, p. 917-920. DOI: <http://dx.doi.org/10.1109/BIOROB.2006.1639208>.
- [16] Dozio L, Mantegazza P. "Real time distributed control systems using RTAI". Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'03), Hakodate (Japan), 2003, p. 11-8. DOI: <http://dx.doi.org/10.1109/ISORC.2003.1199229>.