

DIIS – I3A  
Universidad de Zaragoza  
C/ María de Luna 1  
E-50018 Zaragoza

## **CONTROL DE ESFUERZOS CON UN ROBOT INDUSTRIAL**

**L. Montano, C. Sagüés**

Si usas este informe, por favor usa la cita:

“Control de Esfuerzos con un Robot Industrial”, L. Montano y C. Sagüés,  
Informática y Automática, Vol 30(2), Junio 1997.

Este trabajo ha sido financiado parcialmente por CICYT, proyecto ROB91-0949.

# CONTROL DE ESFUERZOS CON UN ROBOT INDUSTRIAL

Montano Gella Luis, Sagiés Blazquiz Carlos

**Resumen.** En este artículo se presenta la utilización de un sensor de esfuerzos montado en el órgano terminal de un robot PUMA 560 para la realización de movimientos acomodaticios con contacto. Se ha diseñado un procedimiento de calibración del sensor, que se ejecuta sobre el propio robot. Se consideran dos aspectos en la programación y la ejecución de este tipo de movimientos: la especificación de tareas y el control de movimientos. Con respecto al primer aspecto, se introducen las consideraciones generales de especificación de diferentes tareas y las primitivas del lenguaje desarrolladas para su programación. En lo relativo al aspecto de control, se detalla el esquema de control propuesto y su implementación en el robot industrial. Finalmente, se muestran algunos ejemplos de aplicación en tareas industriales y los resultados experimentales obtenidos.

## 1 INTRODUCCIÓN

La utilización de sensores en los sistemas robóticos hace posible que los robots puedan ir incorporándose a tareas más complejas donde el entorno es menos estructurado y conocido. Con este objetivo hemos desarrollado un entorno de programación y control para un sistema robótico multisensorial, denominado APRIL [7].

Este trabajo se centra en la utilización de un sensor de esfuerzos para la ejecución de tareas en las que el robot interacciona con los objetos del entorno de forma que su movimiento viene determinado por la geometría externa. Debido a ello, su trayectoria no puede conocerse en el momento de la programación. Estos movimientos se han denominado clásicamente movimientos acomodaticios (compliant motions) y se han utilizado en tareas de ensamblado y mecanizado, exploración para el agarre, la manipulación o incluso medida de características de superficies en robots móviles [11].

La programación de tareas en el sistema está basada en las especificaciones propuestas en [5] y [2]. En éste último trabajo se presenta un formalismo de especificación de tareas que amplía el presentado en [5] y se consideran aspectos de aplicaciones reales como fricciones o el mantenimiento de determinada orientación del órgano terminal frente a una superficie. En este artículo se adopta una aproximación similar y se describen las primitivas incorporadas al lenguaje de programación, para la especificación de tareas. Se esbozan algunas ideas de especificación de alto nivel, que facilitan al usuario la programación, evitando tener que

conocer detalles de implementación.

Se propone un esquema de control de movimientos basado en la idea de amortiguamiento generalizado [13]. La información del sensor de esfuerzos, colocado en la muñeca del robot, se utiliza para corregir el movimiento nominal programado. Este esquema presenta la ventaja, frente a otros propuestos en la literatura ([6], [9]) de poder ser implantado en robots con control de posición/velocidad, que son los habitualmente utilizados en la industria [1].

En §2 se describe brevemente la arquitectura informática del sistema en el que se ha llevado a cabo la implementación. En §3 se presenta el sensor de esfuerzos utilizado y el procedimiento de calibración. El §4 se dedica a la especificación de tareas con movimientos acomodaticios y se esbozan algunas ideas sobre un método de especificación de alto nivel. En §5 se detalla el sistema de control implementado. En §6 se explican algunos ejemplos de aplicación y los resultados experimentales obtenidos.

## **2 ENTORNO DE IMPLEMENTACION**

Las ideas que se desarrollan en este trabajo se han implementado sobre el lenguaje de programación y control de robots denominado APRIL, diseñado para ofrecer al usuario un lenguaje de nivel textual explícito, y coordinar el robot y los diferentes sensores que constituyen el sistema.

En lo referente a su arquitectura, el sistema está constituido por un robot PUMA 560 y sensores de esfuerzos, proximetría y visión. El soporte informático está compuesto por estación de trabajo SUN-SPARC, el controlador MK-II del robot y un sistema multiprocesador de tiempo real, denominado OSIRIS (Figura 1).

La estación SUN coordina el funcionamiento de todos los subsistemas y en ella reside el entorno de programación desarrollado como extensión de un lenguaje de propósito general (MODULA-2). OSIRIS funciona como subsistema esclavo de SUN, procesando la información sensorial y controlando los movimientos del robot en función de dicha información. Está constituido por un bus VME, en el cual se han instalado tarjetas basadas en microprocesadores de la familia MC680x0. Cada sensor tiene asociado un procesador, lo que posibilita la utilización de varios sensores simultáneamente. Los programas de OSIRIS se han codificado en C.

La estación SUN está comunicada con el controlador MK-II a través de una línea serie (SUPERVISOR), por la que envía órdenes de movimiento y recibe información relativa al

estado del robot. La comunicación SUN-OSIRIS se realiza mediante otra línea serie, que le permite ordenar operaciones de percepción y recibir la información sensorial. Una tercera línea serie (ALTER) comunica directamente el sistema OSIRIS con el controlador del robot. Esta línea permite realizar los movimientos de acomodación (en los que la trayectoria del robot se modifica en tiempo real a partir de la información sensorial), así como atender condiciones de excepción (p. e. sobreesfuerzos).

Actualmente se está procediendo a la reimplatación del sistema de programación, sustituyendo los lenguajes MODULA-2 y C por Ada, con el fin de unificar lenguajes y facilitar la consideración de aspectos de tiempo real.

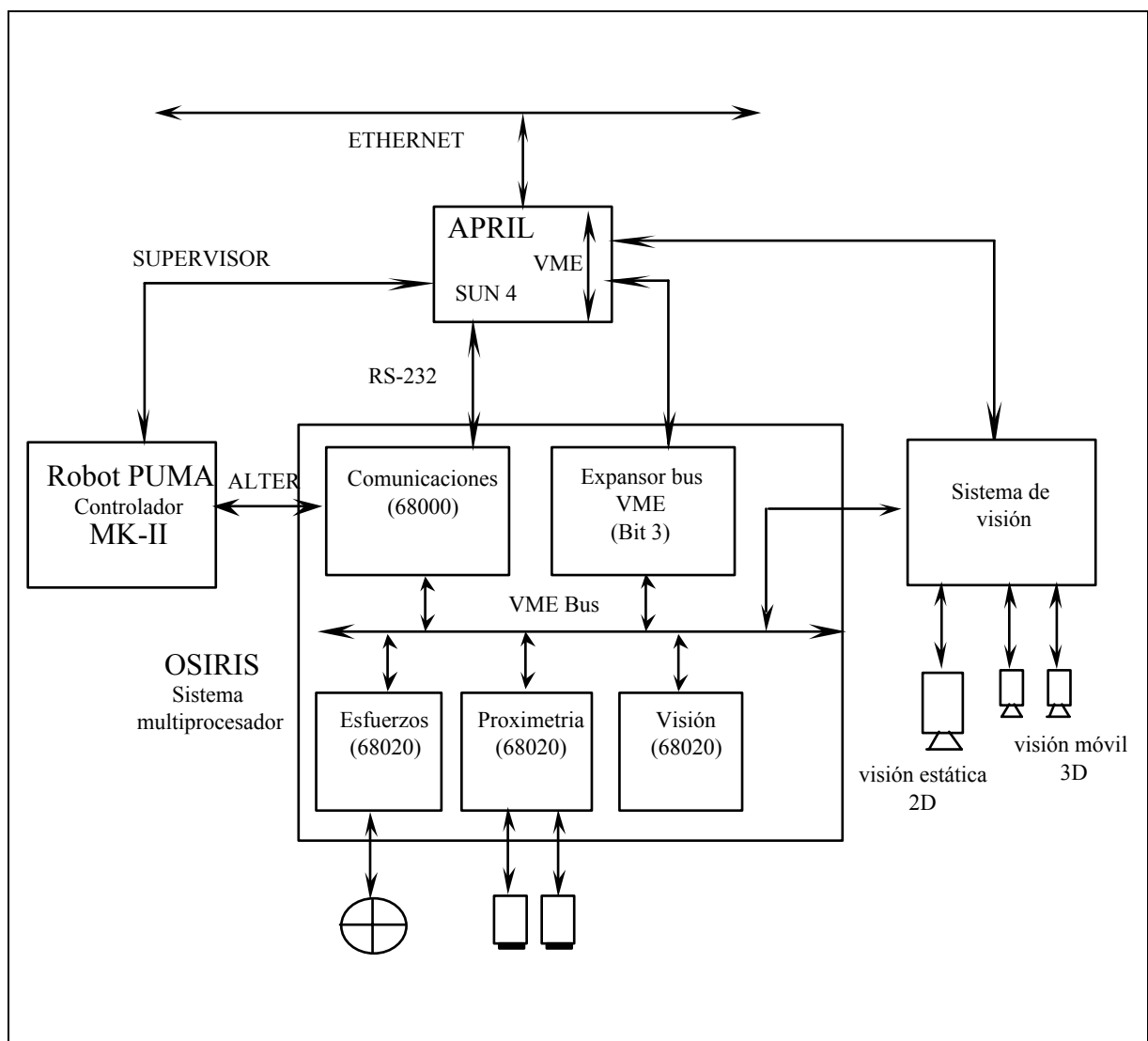


Figura 1. Arquitectura informática

### 3 SENSOR DE ESFUERZOS

El sensor de esfuerzos (diseñado en el C.P.S. de Ingenieros de Zaragoza) permite medir 3 fuerzas y 3 momentos en los ejes de un sistema de referencia asociado al mismo. En § 3.1 se describe el sensor y en § 3.2 se presenta el procedimiento de calibración del mismo.

#### 3.1 Descripción del sensor

El sensor es una estructura mecánica en forma de Cruz de Malta construída de una aleación de aluminio (Figura 2). Está constituido por un anillo central que se fija a la garra del robot. Desde él, salen cuatro vigas de sección variable diseñadas para ser el soporte de las galgas extensiométricas. Mediante un anillo exterior se fija en una caja que protege y aísla al sensor del exterior y que a su vez está atornillada a la muñeca del robot. Su peso es aproximadamente de 200 gramos y el material elegido para su construcción presenta especiales cualidades mecánicas y de deformación.

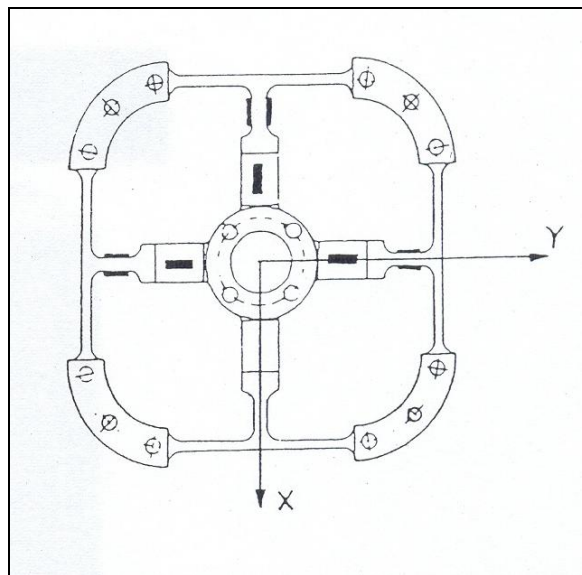


Figura 2. Sensor de esfuerzos

La detección de esfuerzos se realiza por medio de 6 pares de galgas extensiométricas, situadas a ambos lados de los radios, de forma que las deformaciones detectadas por cada par, son complementarias. Ante una fuerza o par ejercidos sobre el sensor, se produce la deformación de las galgas, que a su vez provoca una variación de su resistencia, desequilibrando la tensión en los puentes de Wheatstone sobre los que están conectadas. Esta

tensión es del orden de milivoltios y se amplifica con amplificadores de precisión (ganancia de 2000) que están dispuestos junto a la estructura.

Las medidas obtenidas son sensibles al ruido ambiental y para reducirlo se utilizan filtros analógicos de tercer orden (frecuencia de transición de 40 Hz) antes de la conversión A/D. Como medida de seguridad se dispone además de un comparador que activa una parada de emergencia del robot cuando las tensiones después del amplificador son superiores a un determinado nivel, lo que corresponde a fuerzas ejercidas mayores que las consideradas como máximas soportables por la estructura del sensor.

Los esfuerzos máximos medibles son 6 kg. en las componentes de fuerza, 22.2 kg.cm en los momentos  $m_x$  y  $m_y$  y 20 kg.cm en el momento  $m_z$ . El análisis de los errores de las medidas realizadas en un banco de ensayos diseñado a tal efecto y para la calibración del sensor, ha dado como resultado unos valores de desviación típica de 0.03 kg. en fuerzas y 0.2 kg.cm en momentos.

### 3.2 Calibración del sensor

El proceso de calibración consiste en la obtención de la matriz de calibración  $K$ , que relaciona las tensiones eléctricas medidas en cada una de las seis galgas del puente,  $s = (s^I, s^{II}, s^{III}, s^{IV}, s^V, s^{VI})^T$  y los esfuerzos (fuerzas y momentos) ejercidos,  $f = (f_x, f_y, f_z, m_x, m_y, m_z)^T$ , expresada como  $f = K s$ .

La calibración se ha realizado en el banco de ensayos y en el robot. En este último caso, se utilizan pesas y un objeto de peso conocido que se colocan en la garra del robot, en distintas posiciones para generar una serie de torsores de esfuerzos. Con 6 torsores de esfuerzos,  $F = [f_1, f_2, f_3, f_4, f_5, f_6]$  y la lectura de las tensiones correspondientes en las galgas,  $S = [s_1, s_2, s_3, s_4, s_5, s_6]$ , la matriz de calibración  $K$  se puede obtener como:

$$K = F S^{-1}$$

En la figura 3 se representan las dimensiones del conjunto garra-sensor, así como la situación de los sistemas de referencia asociados a la garra (tool) y el sensor.

El procedimiento de calibración implementado se basa en el método propuesto en [10]. Sobre él se han hecho algunas modificaciones con objeto de mejorar la precisión de los resultados.

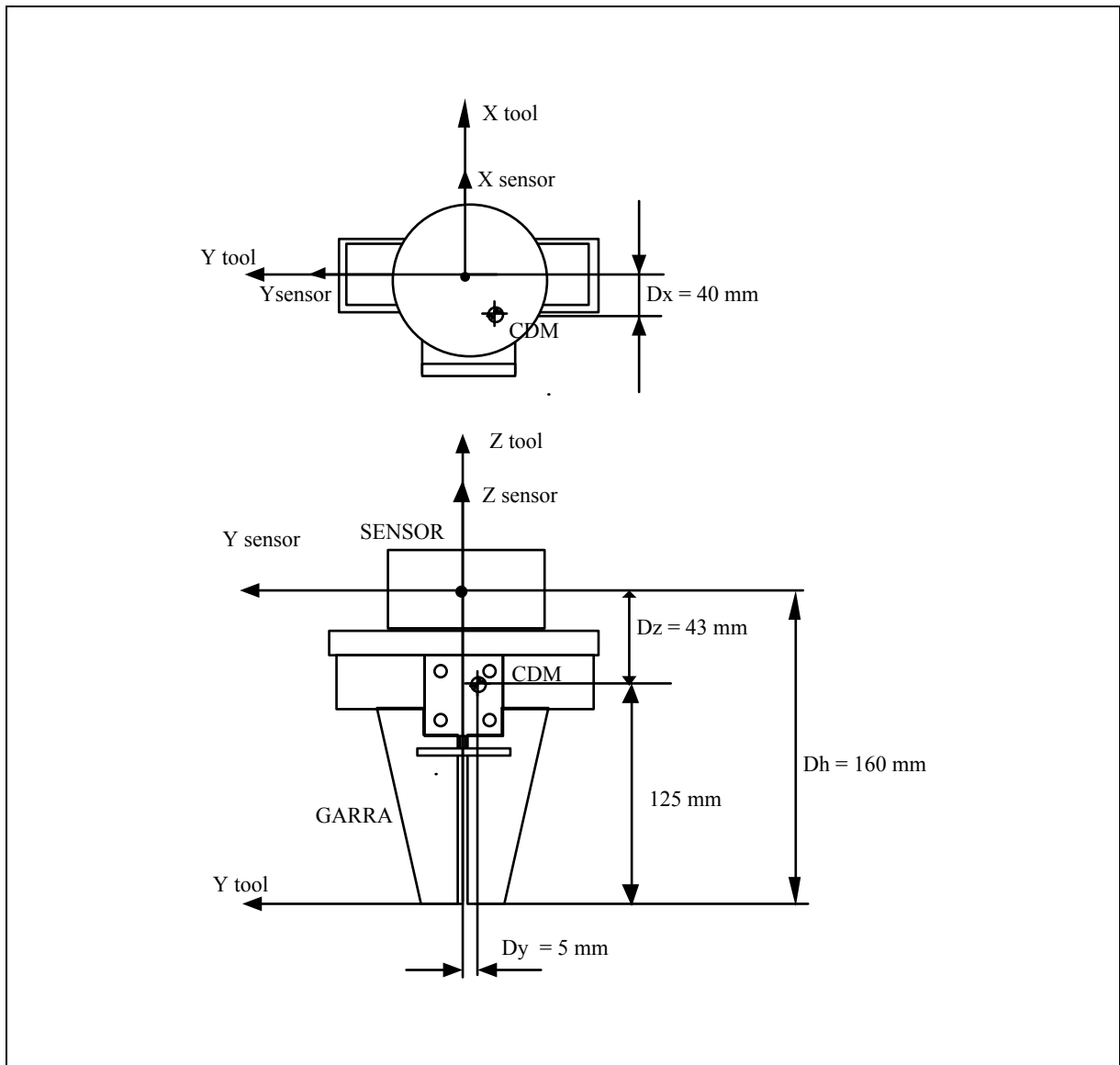


Figura 3. Dimensiones del conjunto garra-sensor y sistemas de referencia asociados

Se han utilizado siete posiciones del robot para conseguir redundancia en las medidas y siete esfuerzos independientes. En las posiciones 1 y 2 el robot únicamente porta un objeto (0.502 Kg.) y no se introducen pesas. En las posiciones 3 y 4 se cambian las pesas colocadas sobre la garra (hasta un máximo de 1 Kg.) y centradas en los dedos. En las posiciones 5 y 6 se colocan objeto y pesas (hasta un máximo de 0.6 Kg.) para provocar momentos mayores. En la posición 7 se coloca objeto sin pesas, ya que las medidas son ya apreciables. Las expresiones generales de los vectores de esfuerzos medidos en estas posiciones se resumen en la tabla 1, donde G es el peso de la garra, O es el peso del objeto sobre la misma y P el peso de las pesas que se colocan adicionalmente en algunas posiciones.

	P1	P2	P3	P4	P5	P6	P7
$f_x$	0	0	G+P	-G -P	0	0	0
$f_y$	0	0	0	0	G+O+P	-G-O-P	G+O
$f_z$	G+O	-G-O	0	0	0	0	0
$m_x$	-Dy*G	Dy*G	0	0	Dz*G+Dh*(O+P)	-Dz*G-Dh*(O+P)	Dz*G+Dh*O
$m_y$	Dx*G	-Dx*G	-Dz*G-Dh*P	Dz*G+Dh*P	0	0	0
$m_z$	0	0	Dy*G	-Dy*G	-Dx*G	Dx*G	-Dx*G

Tabla 1. Esfuerzos (en la referencia del sensor) en las posiciones de calibración

Para calibrar se realizaron diferentes ensayos, en cada una de las posiciones con distintos pesos. A partir de la matriz de esfuerzos  $F = [ f_1 f_2 f_3 f_4 f_5 f_6 f_7 ]$  y la matriz de medidas  $S = [ s_1 s_2 s_3 s_4 s_5 s_6 s_7 ]$  se calcula la matriz de calibración para cada ensayo:

$$K = F \cdot S^{\#}$$

$$\text{siendo } S^{\#} = (S^T S)^{-1} S^T$$

La matriz de calibración media obtenida de los diferentes ensayos es:

$$K = \begin{pmatrix} -0.356 & 0.401 & -0.364 & 0.963 & -0.491 & -0.156 \\ -0.709 & 0.244 & 0.364 & -1.276 & -0.449 & 1.109 \\ 1.945 & -1.968 & -0.963 & 0.427 & 0.295 & -0.070 \\ 4.418 & 1.595 & -2.910 & -2.392 & 0.108 & 2.082 \\ 1.709 & 0.500 & -0.870 & -3.160 & -6.814 & 6.588 \\ 2.112 & 0.279 & -2.935 & -0.450 & 1.126 & 0.005 \end{pmatrix}$$

Puede observarse que cada galga es especialmente sensible a unos esfuerzos determinados, en función de su colocación en la estructura mecánica.

La matriz obtenida se verificó con medidas de fuerzas conocidas en distintas posiciones del robot. En la figura 4 se representan algunos de los resultados de las pruebas efectuadas. Puede comprobarse el comportamiento aproximadamente lineal del sensor.

Actualmente se está experimentando con un sensor comercial (Zebra Robotics Inc.), esperando obtener mejores resultados. El fabricante de este sensor especifica una sensibilidad de 25 gramos.



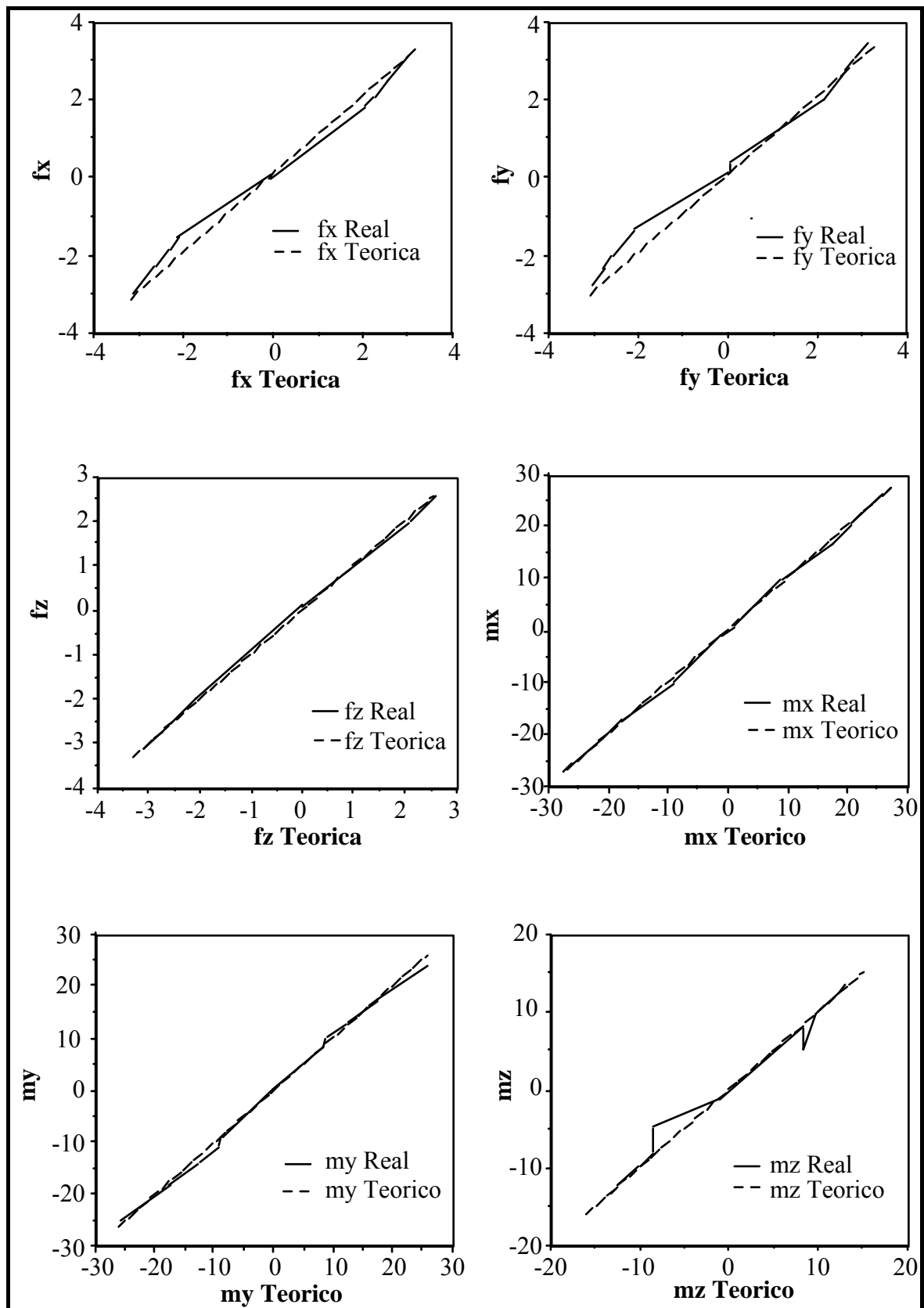


Figura 4.- Resultados de calibración

#### 4 ESPECIFICACION DE TAREAS

En este apartado se aborda la especificación de tareas con movimientos del robot restringidos por la geometría del entorno. En la figura 5 se representan dos tareas tipo: la inserción de un vástago en un taladro y el seguimiento de una superficie. En ambos casos existen grados de libertad en los que el movimiento es libre y otros en los que está restringido por la geometría de la tarea. Los primeros pueden ser controlados en posición/velocidad y los segundos en fuerza/par.

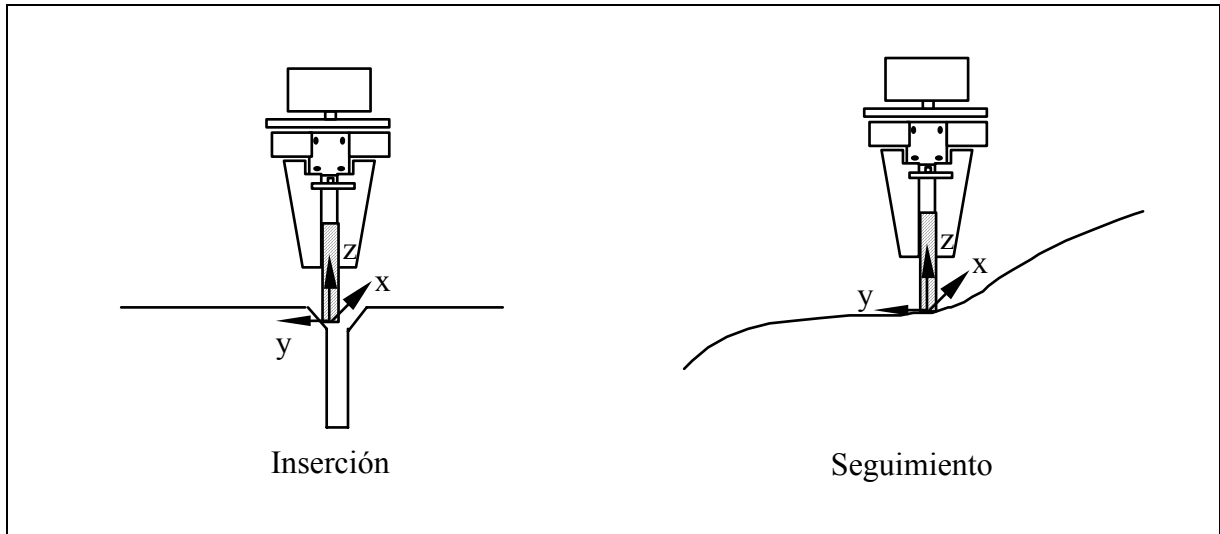


Figura 5. Tareas típicas con movimientos acomodaticios

Si se denomina  $\mathbf{v} = (v_x, v_y, v_z, w_x, w_y, w_z)^T$  al vector de velocidades y  $\mathbf{f} = (f_x, f_y, f_z, m_x, m_y, m_z)^T$  al vector de esfuerzos, en la tarea de inserción y con la referencia de acomodación asociada al extremo del vástago, la especificación de la tarea de inserción en el caso ideal es la siguiente:

$$v_z = v \text{ mm./seg.} \quad w_z = 0$$

para los grados de libertad de no acomodación, y

$$f_x = 0 \text{ kg.}, \quad f_y = 0 \text{ kg.}, \quad m_x = 0 \text{ kg.mm.}, \quad m_y = 0 \text{ kg.mm.}$$

para los grados de libertad de acomodación. El movimiento en las direcciones de acomodación  $(x, y, \psi_x, \theta_y)$  debe corregirse para mantener las fuerzas especificadas.

En la tarea de seguimiento de superficie, manteniendo la orientación de la garra normal a la misma, las especificaciones son:

$$v_x = 0, \quad v_y = v \text{ mm./seg.}, \quad w_y = 0, w_z = 0$$

para los grados de libertad de no acomodación, y

$$f_z = f \text{ kg.} \quad m_x = 0$$

para los grados de libertad de acomodación. El movimiento debe acomodarse en la dirección Z para mantener la fuerza especificada y en torno al eje X para mantener la orientación perpendicular a la superficie.

Como se ha comentado en la introducción, el formalismo para la especificación de tareas es semejante al presentado en [2]. En el sistema de programación desarrollado se han definido primitivas que permiten al usuario especificar los parámetros necesarios. Estos parámetros son:

- *Movimiento nominal.* Para especificar el desplazamiento y velocidad con que se realiza el mismo. Se expresa con dos parámetros:
  - *Referencia nominal de destino del movimiento.* Se refiere a la situación nominal que alcanzaría el órgano terminal del robot si no existiese contacto con el entorno.
  - *Velocidades de traslación y rotación nominales.* Se expresan en porcentaje de la velocidad máxima. Estos parámetros, junto con el anterior, definen el movimiento nominal, es decir la trayectoria y la velocidad de consigna.
- *Referencia de acomodación.* Es la referencia en la que se calculan las correcciones del movimiento en los grados de libertad de acomodación, que hacen posible que el robot se adapte a la geometría del entorno. En su implementación actual puede definirse una referencia fija con el sistema de coordenadas absoluto o fija con el órgano terminal.
- *Esfuerzos de consigna.* Se expresan en la referencia de acomodación y representan las fuerzas y pares de consigna en los g. d. l. de acomodación. Los g.d.l. en los que no se produce acomodación se especifican como indefinidos (Ind).
- *Condición de finalización del movimiento.* Son dos los tipos de condiciones actualmente implementadas: (1) detección de esfuerzos máximos en alguno de los ejes del sistema de referencia, utilizada, por ejemplo, en la inserción de piezas; y (2) alcance de la situación nominal de destino en alguno de los ejes del sistema de referencia de no acomodación, utilizadas, por ejemplo, en el seguimiento de superficies.

Adicionalmente a estos parámetros, para cada tarea particular deben definirse unos parámetros de control. Dichos parámetros están directamente relacionados con el esquema de control y deben elegirse para obtener un comportamiento dinámico deseado. En el sistema implementado es el propio programador el que las define o bien pueden ser seleccionadas en una biblioteca de tareas típicas definidas previamente. La definición y elección del valor de dichos parámetros se aborda en los apartados 5 y 6. En la Tabla 2 se indican las primitivas definidas para la programación de las tareas.

<p>PROCEDURE ValorRef (T1: Transformacion) : Referencia; (* Asigna el valor de T1 a una Referencia *)</p> <p>PROCEDURE LeeValorRef (Ref: Referencia) : Transformacion ; (* Obtiene el valor de la transformacion correspondiente a la referencia Ref *)</p> <p>PROCEDURE TransfDifRef (R1,R2: Referencia) : Transformacion; (* Calcula la transformacion relativa entre R1 y R2 *)</p> <p>PROCEDURE CompTransf (T1, T2 : Transformacion) : Transformacion; (* Compone las transformaciones T1 y T2 *)</p> <p>PROCEDURE Velocidad (VAR Vel: VelTrasRot; VTras,VRot: REAL) ; (* Asigna a Vel valores de velocidades de traslacion y de rotacion *)</p> <p>PROCEDURE RefAcom (Ref : Referencia) ; (* Define la referencia de acomodacion *)</p> <p>PROCEDURE DefFuerza (VAR F: Fuerza; fx, fy, fz, mx, my, mz : REAL) ; (* Define un vector de esfuerzos de consigna F. El valor Ind en alguna componente representa g.d.l. de no acomodacion *)</p> <p>PROCEDURE Fx (Tipo: TipoFuerza; F: REAL) : Condicion; PROCEDURE Fy (Tipo: TipoFuerza; F: REAL) : Condicion; PROCEDURE Fz (Tipo: TipoFuerza; F: REAL) : Condicion; PROCEDURE Mx (Tipo: TipoFuerza; F: REAL) : Condicion; PROCEDURE My (Tipo: TipoFuerza; F: REAL) : Condicion; PROCEDURE Mz (Tipo: TipoFuerza; F: REAL) : Condicion; PROCEDURE DestAlcanzado () : Condicion; (* Funciones correspondientes a condiciones de finalizacion del movimiento. TipoFuerza es (Mayor, Menor, Igual) *)</p> <p>PROCEDURE MatCorreccion (D:MCorreccion;                                   d11, d12, d13, d14, d15, d16,                                   d21, d22, d23, d24, d25, d26                                   d31, d32, d33, d34, d35, d36                                   d41, d42, d43, d44, d45, d46                                   d51, d52, d53, d54, d55, d56                                   d61, d62, d63, d64, d65, d66 : REAL) ;</p> <p>PROCEDURE MatCorrecDiag (D: MCorreccion; d1, d2, d3, d4, d5, d6 : REAL) ; (* Procedimientos para definir la matriz de correccion D *)</p> <p>PROCEDURE MovAcom (Tr: Transformacion; Vel: VelTrasRot; Fcons: Fuerza; D:MCorreccion;                                   CondFinal: Condicion) ; (* Provoca la realizacion de un movimiento acomodaticio de acuerdo con los parametros indicados. Tr representa la transformacion relativa entre las referencias origen y destino nominales *)</p>
---

Tabla 2. Primitivas del lenguaje para programación de movimientos acomodaticios

## Especificación de tareas a alto nivel

La forma de especificación de tareas presentada tiene el inconveniente de que la elección de los parámetros, tanto los relacionados directamente con la tarea como los de control, puede no resultar sencilla en determinadas aplicaciones, ya que el programador debe conocer detalles del sistema de control de movimiento.

Para facilitar la programación, se ha iniciado la implementación de un método de especificación de alto nivel, en el que el usuario no define directamente los parámetros antes descritos, sino que es el propio sistema de programación el que los genera automáticamente. Con este método, basado en [3], la especificación de la tarea se realiza a nivel de relaciones de contacto entre las diferentes características geométricas que constituyen los objetos (vértices, aristas, caras), indicando los movimientos permitidos y las restricciones de movimiento debidas al contacto deseado durante la tarea.

A modo de ejemplo considérese la tarea de la figura 6 en la que un bloque (b) debe desplazarse a lo largo de la guía (g). Los parámetros que el programador debe especificar en este caso son:

- *Secuencia de configuraciones de contacto.* Define los tramos del movimiento en los que las configuraciones de contacto entre los objetos son diferentes. En el ejemplo existe un único tramo, ya que las características en contacto son siempre las mismas.
- *Tipos de contacto en cada configuración.* Se definen los contactos entre características de los objetos bloque y guía: vértice-arista, cara-arista, arista-arista, etc. En la configuración de la figura 6 existen tres contactos de tipo cara-cara (cg1-cb1, cg2-cb2, cg3-cb3).
- *Localización de la referencia común.* Se define un sistema de referencia común a todas las características (en este caso O).
- *Restricciones de contacto.* Para cada característica en contacto con otra se definen las restricciones de movimiento, especificando el vector de esfuerzos. Por ejemplo, para el contacto cg3-cb3 el vector sería  $\mathbf{f} = (0, 0, F, 0, 0, 0)^T$  cuyas componentes representan las fuerzas y pares en el sistema de referencia asociado a la característica.
- *Movimiento deseado.* Para cada característica se definen los grados de libertad de movimiento mediante un vector de velocidades (3 lineales y 3 angulares) en el sistema de referencia asociado a la característica. En el caso de la característica cb1, el vector sería  $\mathbf{v} = (v, 0, 0, 0, 0, 0)^T$ .

A partir de estas especificaciones el sistema genera las consignas al sistema de control de esfuerzos, basado en un *control híbrido* velocidad-fuerza, aunque implementado utilizando el controlador de posición/velocidad del robot.

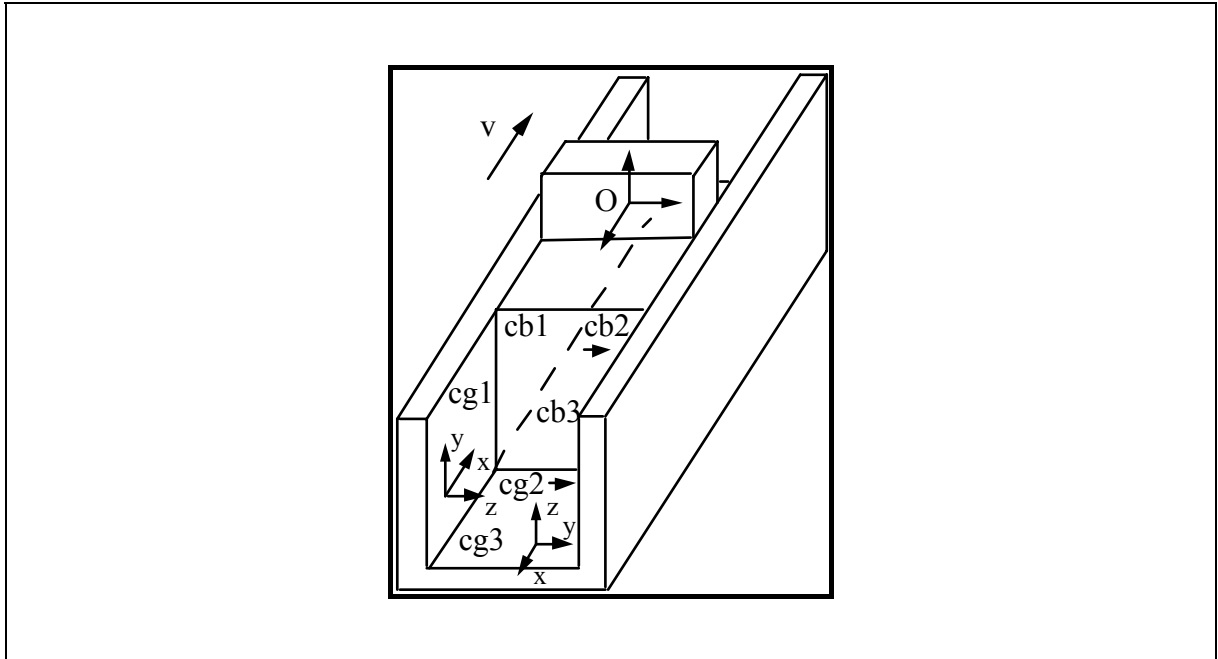


Figura 6. Ejemplo de tarea con movimiento restringido

Este método simplifica notablemente la especificación de tareas, sobre todo cuando el movimiento del robot está fuertemente restringido. Una descripción más detallada escapa a los objetivos de este trabajo y será objeto de futuras publicaciones.

## 5 CONTROL DE ESFUERZOS

El controlador diseñado permite modificar la trayectoria nominal programada en un robot PUMA 560, en función de las fuerzas y pares detectados y es compatible con el método de especificación de tareas del apartado 4. Se describe a continuación el esquema básico de control y se detalla la implementación del algoritmo de control utilizado.

### 5.1 Esquema de control

La ley de control se basa en un esquema de *amortiguamiento generalizado* [13]. En la figura 7 se representa el esquema básico de control. En él, la velocidad nominal del movimiento representada en un sistema de referencia (3 componentes de velocidad lineal y 3 de velocidad angular) se corrige a partir del error existente en cada instante entre la consigna de esfuerzos y los esfuerzos detectados.

Matemáticamente puede expresarse de la siguiente forma:

$$A\dot{\mathbf{x}}_e = D (A\mathbf{f}_C - A\mathbf{f})$$

$$A\dot{\mathbf{x}}_r = A\dot{\mathbf{x}}_n + A\dot{\mathbf{x}}_e$$

donde:

$D$  : matriz de corrección (inversa de la matriz de amortiguamiento). Es la matriz a la que se ha hecho referencia en el apartado 4 y es específica para cada tarea.

$A\mathbf{f}_C = (f_x^c, f_y^c, f_z^c, m_x^c, m_y^c, m_z^c)^T$ : consigna de esfuerzos en la referencia de acomodación  
(A)

$A\mathbf{f} = (f_x, f_y, f_z, m_x, m_y, m_z)^T$ : esfuerzos medidos en la referencia de acomodación

$A\dot{\mathbf{x}}_e = (\dot{x}_e, \dot{y}_e, \dot{z}_e, \dot{\psi}_e, \dot{\theta}_e, \dot{\phi}_e)^T$ : corrección de velocidad en la referencia de acomodación

$A\dot{\mathbf{x}}_n = (\dot{x}_n, \dot{y}_n, \dot{z}_n, \dot{\psi}_n, \dot{\theta}_n, \dot{\phi}_n)^T$ : velocidad nominal en la referencia de acomodación

$A\dot{\mathbf{x}}_r = (\dot{x}_r, \dot{y}_r, \dot{z}_r, \dot{\psi}_r, \dot{\theta}_r, \dot{\phi}_r)^T$ : velocidad corregida en la referencia de acomodación

La velocidad  $A\dot{\mathbf{x}}_r$  debe transformarse al sistema de referencia en el que el robot la interpreta.

Dado que los esfuerzos medidos por el sensor ( $S\mathbf{f}$ ) están representados en el sistema de referencia del propio sensor (S), es necesario realizar una transformación de esfuerzos entre los sistemas de referencia S y A.

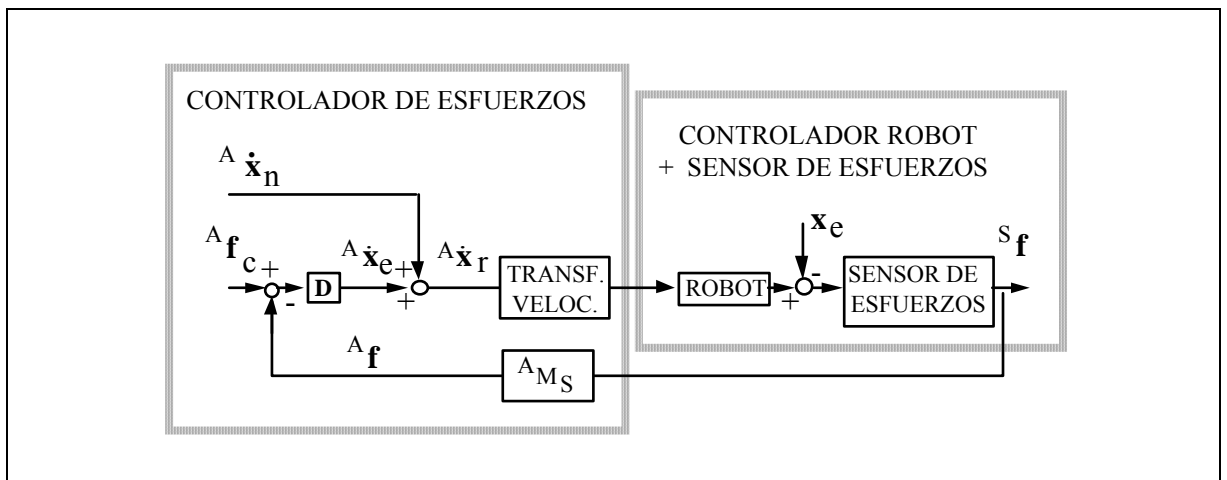


Figura 7. Control por amortiguamiento generalizado

Dado un vector de esfuerzos  ${}^S\mathbf{f}$  en el sistema de referencia S, el vector  ${}^A\mathbf{f}$  expresado en el sistema A, se calcula como:

$${}^A\mathbf{f} = {}^A\mathbf{M}_S {}^S\mathbf{f}$$

donde  ${}^A\mathbf{M}_S$  es la matriz:

$${}^A\mathbf{M}_S = \begin{pmatrix} \mathbf{R} & \mathbf{O}_{3 \times 3} \\ \mathbf{S} & \mathbf{R} \end{pmatrix}$$

siendo  $\mathbf{R} = (\mathbf{n} \ \mathbf{o} \ \mathbf{a})$  y  $\mathbf{S} = (\mathbf{p} \times \mathbf{n} \ \mathbf{p} \times \mathbf{o} \ \mathbf{p} \times \mathbf{a})$ , obtenidas de la matriz homogénea  ${}^A\mathbf{H}_S$  que representa la transformación entre ambas referencias:

$${}^A\mathbf{H}_S = \begin{pmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 1 \end{pmatrix}$$

Cuando la referencia de acomodación es fija con el órgano terminal,  ${}^A\mathbf{M}_S$  es constante. En otro caso es variable y es preciso calcularla en cada período de muestreo.

Los elementos de la matriz de corrección D (6x6) dependen de cada tarea y su valor del comportamiento dinámico deseado. En el apartado 6 se presenta su elección para algunas tareas.

## 5.2 Implementación del control

En la implementación del esquema de control, se utiliza el propio controlador del robot PUMA 560 sobre el que se cierra un bucle externo de realimentación de esfuerzos implementado en el sistema multiprocesador (Figura 7). Se han considerado dos maneras posibles de realizar el control:

- (1) el controlador del robot recibe del controlador de esfuerzos una consigna de incremento de posición en cada período de muestreo, siendo aquél el que realiza la integración. En este caso puede considerarse que se trata de un control de velocidad.
- (2) el controlador del robot recibe del controlador de esfuerzos una consigna de posición y orientación, siendo éste el que realiza la integración.

En ambos casos, en cada período de muestreo (28 ms., impuesto por el controlador del robot) se calcula la velocidad o la posición y orientación que debe transmitirse al robot,



debida al movimiento nominal deseado y al error de esfuerzos en los ejes del sistema de referencia de acomodación.

Teniendo en cuenta que la corrección de velocidad se calcula en la referencia de acomodación (A), debe realizarse una transformación del vector velocidad a la referencia, asociada al órgano terminal en cada instante ( $T_i$ ). Esta se obtiene mediante el Jacobiano [8]:

$${}^{T_i}J_A = \begin{pmatrix} R & S \\ O_{3 \times 3} & R \end{pmatrix}$$

siendo  $R = (\mathbf{n} \ \mathbf{o} \ \mathbf{a})$  y  $S = (\mathbf{p} \times \mathbf{n} \ \mathbf{p} \times \mathbf{o} \ \mathbf{p} \times \mathbf{a})$ , obtenidas de la matriz homogénea  ${}^{T_i}H_A$  que representa la transformación entre ambas referencias.

Sea  ${}^{T_i}\mathbf{x}_A = (x, y, z, \psi, \theta, \phi)^T$  el vector de localización [12] que representa la transformación entre las referencias  $T_i$  y A, en la que los parámetros de orientación son Yaw-Pitch-Roll. Sea  ${}^A\delta\mathbf{x}_d = (\delta x, \delta y, \delta z, \delta\psi, \delta\theta, \delta\phi)^T$  el vector diferencial de localización, calculado a partir del error de esfuerzos, expresado en la referencia A. La corrección, expresada en forma de incremento de posición y orientación en la referencia  $T_i$ , se obtiene de la siguiente manera:

$${}^{T_i}\delta\mathbf{x}_d(k) = {}^{T_i}J_A(k) {}^A\delta\mathbf{x}_d(k)$$

y la corrección total en cada período de muestreo es:

$${}^{T_i}\delta\mathbf{x}_r(k) = {}^{T_i}\delta\mathbf{x}_n(k) H {}^{T_i}\delta\mathbf{x}_d(k) \quad (1)$$

siendo  ${}^{T_i}\delta\mathbf{x}_n(k)$  la correspondiente al movimiento nominal y  ${}^{T_i}\delta\mathbf{x}_d(k)$  la correspondiente a la acomodación. La operación H representa la composición de vectores de localización [12]. Como ambas correcciones son vectores incrementales, la expresión (1) puede aproximarse por:

$${}^{T_i}\delta\mathbf{x}_r(k) \approx {}^{T_i}\delta\mathbf{x}_n(k) + {}^{T_i}\delta\mathbf{x}_d(k) \quad (2)$$

En el primer tipo de control implementado (control de velocidad), se envía al controlador del robot el incremento de posición calculado en (2) expresado en una referencia  $T_t$ , cuyo origen coincide con el de  $T_i$  y su orientación con la de la referencia asociada al órgano terminal al comienzo del movimiento acomodaticio ( $T_o$ ):

$${}^{T_t}\delta\mathbf{x}_r(k) = {}^{T_t}J_{T_i}(k) {}^{T_i}\delta\mathbf{x}_r(k)$$

donde  ${}^{T_t}J_{T_i}(k)$  es el Jacobiano de la transformación entre  $T_i$  y  $T_t$ .

En el segundo tipo de control (control de posición), la posición y orientación enviada al controlador del robot se calcula componiendo la situación actual expresada en  $T_o$  con la corrección calculada en (2):

$${}^{T_o}\mathbf{x}_{T_i}(k+1) = {}^{T_o}\mathbf{x}_{T_i}(k) H {}^{T_i}\delta\mathbf{x}_r(k)$$

Los parámetros de orientación utilizados por el controlador del robot PUMA 560 no son Yaw-Pitch-Roll, y corresponden a la transformación:

$$\text{Orient}(\psi', \theta', \phi') = \text{Rot}(x, \psi') \text{Rot}(y, \theta') \text{Rot}(z, \phi')$$

Por tanto, los vectores que se envían al controlador del robot son  ${}^{T_t}\delta\mathbf{x}'_r = (\delta x, \delta y, \delta z, \delta\psi', \delta\theta', \delta\phi')^T$  en el caso del control de velocidad y  ${}^{T_o}\mathbf{x}'_{T_i} = (x, y, z, \psi', \theta', \phi')^T$  en el caso del control de posición.

La tabla 3 resume los cálculos a realizar en cada período de muestreo.

${}^A\delta\mathbf{x}_d(k) = D ({}^A\mathbf{f}_c - {}^A\mathbf{f}(k))$ ${}^{T_i}\delta\mathbf{x}_d(k) = {}^{T_i}J_A(k) {}^A\delta\mathbf{x}_d(k)$ ${}^{T_i}\delta\mathbf{x}_r(k) = {}^{T_i}\delta\mathbf{x}_n(k) + {}^{T_i}\delta\mathbf{x}_d(k)$ <p>Control de velocidad:</p> ${}^{T_t}\delta\mathbf{x}_r(k) = {}^{T_t}J_{T_i}(k) {}^{T_i}\delta\mathbf{x}_r(k)$ ${}^{T_t}\delta\mathbf{x}'_r(k) = \text{Transf} ({}^{T_t}\delta\mathbf{x}_r(k) )$ <p>Control de posición:</p> ${}^{T_o}\mathbf{x}_{T_i}(k+1) = {}^{T_o}\mathbf{x}_{T_i}(k) H {}^{T_i}\delta\mathbf{x}_r(k)$ ${}^{T_o}\mathbf{x}'_{T_i}(k+1) = \text{Transf} ({}^{T_o}\mathbf{x}_{T_i}(k+1) )$
---

Tabla 3. Cálculos en cada período de muestreo

La matriz de corrección  $D$  debe ser seleccionada para cada tarea. Para ello se han definido dos primitivas recogidas en la tabla 2. Sus parámetros deben ajustarse para obtener un comportamiento dinámico adecuado [4].

Una cuestión adicional a resaltar es que el esquema de control propuesto es válido también durante el movimiento de aproximación a una superficie. Así, durante la aproximación la velocidad de corrección es:

$${}^A\dot{\mathbf{x}}_e = D {}^A\mathbf{f}_c$$

siendo  ${}^A\mathbf{f}_c$  la fuerza de consigna (la fuerza detectada  ${}^A\mathbf{f} = 0$ , ya que el movimiento es en el espacio libre). La velocidad resultante es:

$${}^A\dot{\mathbf{x}}_r = {}^A\dot{\mathbf{x}}_n + {}^A\dot{\mathbf{x}}_e$$

donde puede imponerse un límite superior a  ${}^A\dot{\mathbf{x}}_r$  para que la velocidad de aproximación no sea excesiva. Ello evita la necesidad de conmutar de esquema de control cuando se produce el contacto con la superficie.

En lo relativo a aspectos computacionales de implementación, todas las operaciones se han optimizado, utilizando únicamente los elementos necesarios, lo que permite completar los cálculos dentro de cada período de corrección impuesto por el controlador del robot (28 ms.).

## 6 EJEMPLOS DE APLICACION

Como aplicación de las técnicas propuestas, se aborda la realización de dos tipos de tareas: el seguimiento de superficies y la inserción de un vástago en un taladro. Se describen los parámetros utilizados y se presentan los resultados experimentales obtenidos.

### 6.1 Seguimiento de una superficie

Se presentan las tareas de seguimiento de una superficie horizontal y de una superficie inclinada manteniendo la garra perpendicular a la superficie.

## Superficie horizontal

En la figura 8 se representa un esquema de una tarea de seguimiento de una superficie horizontal manteniendo una fuerza vertical constante. De acuerdo con lo indicado en el apartado 4, los parámetros correspondientes a la especificación de la tarea son:

- *Movimiento nominal*:  ${}^A\mathbf{v}_n = (0, v, 0, 0, 0, 0)^T$ , expresado mediante los parámetros:
  - *Desplazamiento nominal*: desplazamiento horizontal desde  $R_i$  a  $R_f$
  - *Velocidad nominal*: velocidad de traslación de 10% y velocidad de rotación nula
- *Referencia de acomodación* : referencia Tool del robot
- *Esfuerzos de consigna* :  ${}^A\mathbf{f}_c = (\text{Ind}, \text{Ind}, 2, \text{Ind}, \text{Ind}, \text{Ind})^T$ . Ind representa g.d.l. de no acomodación.
- *Condición de finalización* : alcanzar el destino  $R_f$  en los g. d. l. en los que no existe acomodación

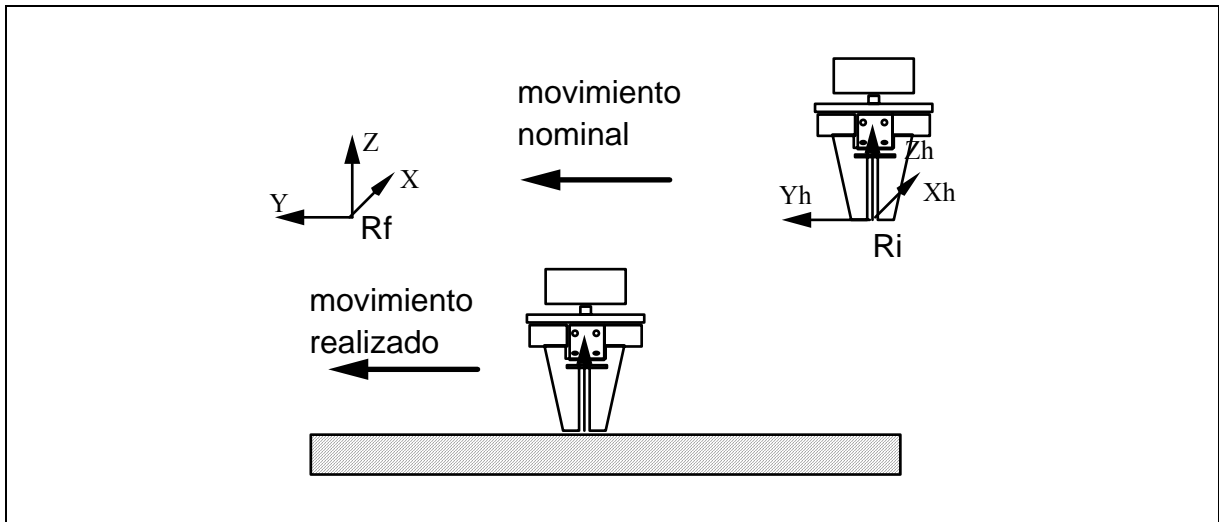


Figura 8. Esquema de tarea de seguimiento horizontal

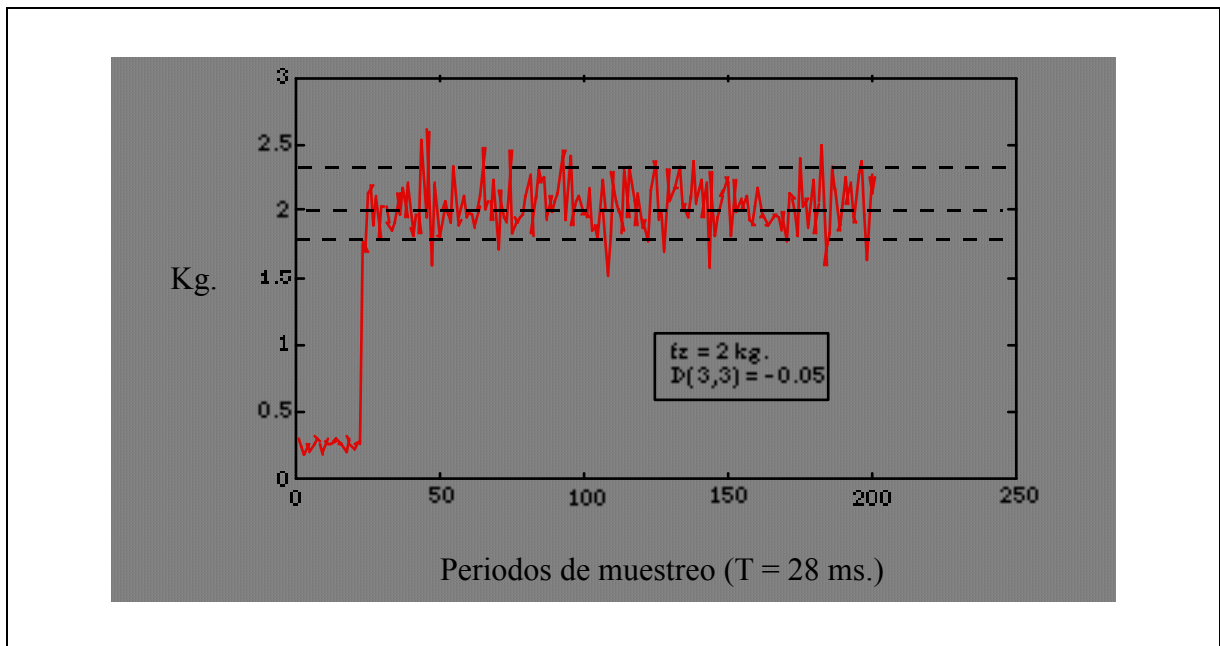


Figura 9. Fuerza medida en el eje Z en un seguimiento horizontal

El único g.d.l. en el que se produce acomodación es el que define la fuerza de consigna. Por tanto, durante el movimiento se realizan correcciones de la velocidad nominal únicamente en la dirección Z del sistema de referencia de acomodación. En este caso la matriz de corrección es diagonal:  $D = \text{diag}(0, 0, d_{33}, 0, 0, 0)$

El valor del parámetro de control  $d_{33} = -0.05$  se ha elegido experimentalmente para obtener un comportamiento próximo al de un sistema sobreamortiguado.

En la figura 9 se representa la fuerza  $f_z$  medida durante el seguimiento. Las oscilaciones sobre la consigna son debidas a la propia rugosidad de los elementos y al ruido de medida. Como puede comprobarse en la figura, se obtiene una precisión de 0.5 Kg. en el mantenimiento de la fuerza sobre la superficie.

### Superficie inclinada con órgano terminal perpendicular a la misma

La figura 10 representa el esquema de la tarea. Las especificaciones impuestas son similares a las de la tarea anterior. La única diferencia es que se desea que el órgano terminal se mantenga perpendicular a la superficie. Para ello debe especificarse que el giro en torno a  $X(\psi_x)$  corresponde a un g.d.l. de acomodación.

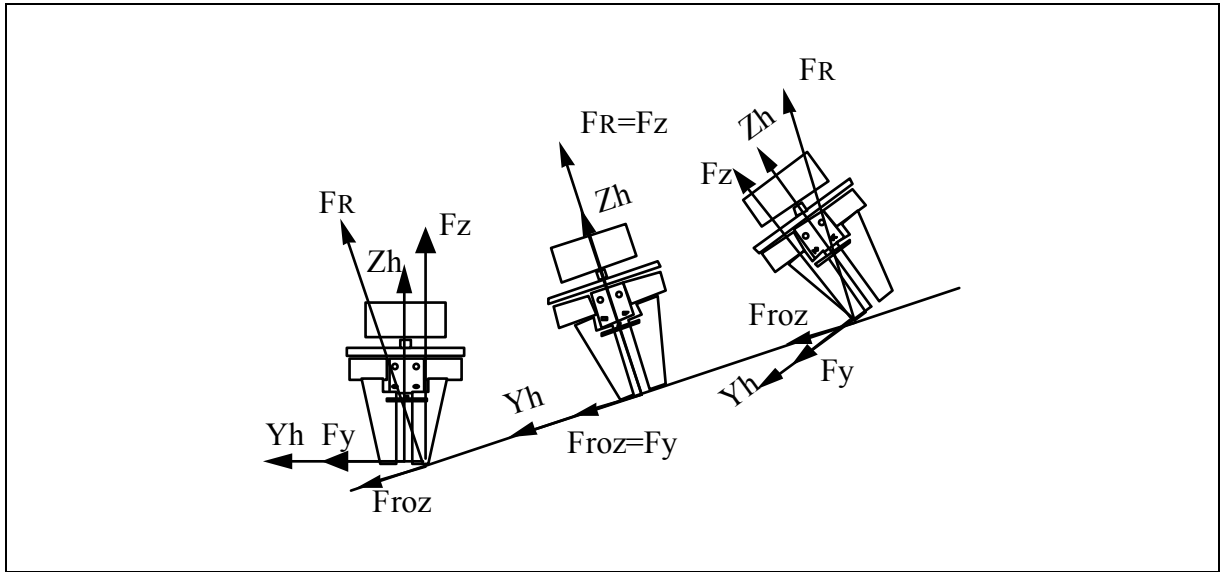


Figura 10. Seguimiento superficie inclinada

En esta tarea deben producirse correcciones a lo largo del eje Z ( $\delta z$ ) producidas por errores de fuerza en dicho eje y en torno al eje X ( $\delta \psi$ ) producidas por los momentos en dicho eje. Por tanto, la forma general de la matriz D es:

$$D = \text{diag}(0, 0, d_{33}, d_{44}, 0, 0)$$

con  $d_{33} = -0.15$ , y  $d_{44} = -0.02$ , obtenidos experimentalmente.

## 6.2 Inserción de vástago en taladro

Esta tarea consiste en el desplazamiento de un vástago dentro de un orificio de profundidad desconocida (Figura 11). Se especifica como concatenación de dos movimientos. En el primero se produce el contacto y seguimiento de una superficie horizontal hasta un punto próximo al orificio (RefSup) con velocidad nominal  ${}^A\mathbf{v}_n = (0, -v_1, 0, 0, 0, 0)^T$ , y consigna de fuerza  ${}^A\mathbf{f}_c = (\text{Ind}, \text{Ind}, 1, \text{Ind}, \text{Ind}, \text{Ind})^T$ . En este caso no se fuerza la acomodación en torno al eje X por tratarse de una superficie horizontal con el órgano terminal perpendicular a la misma.

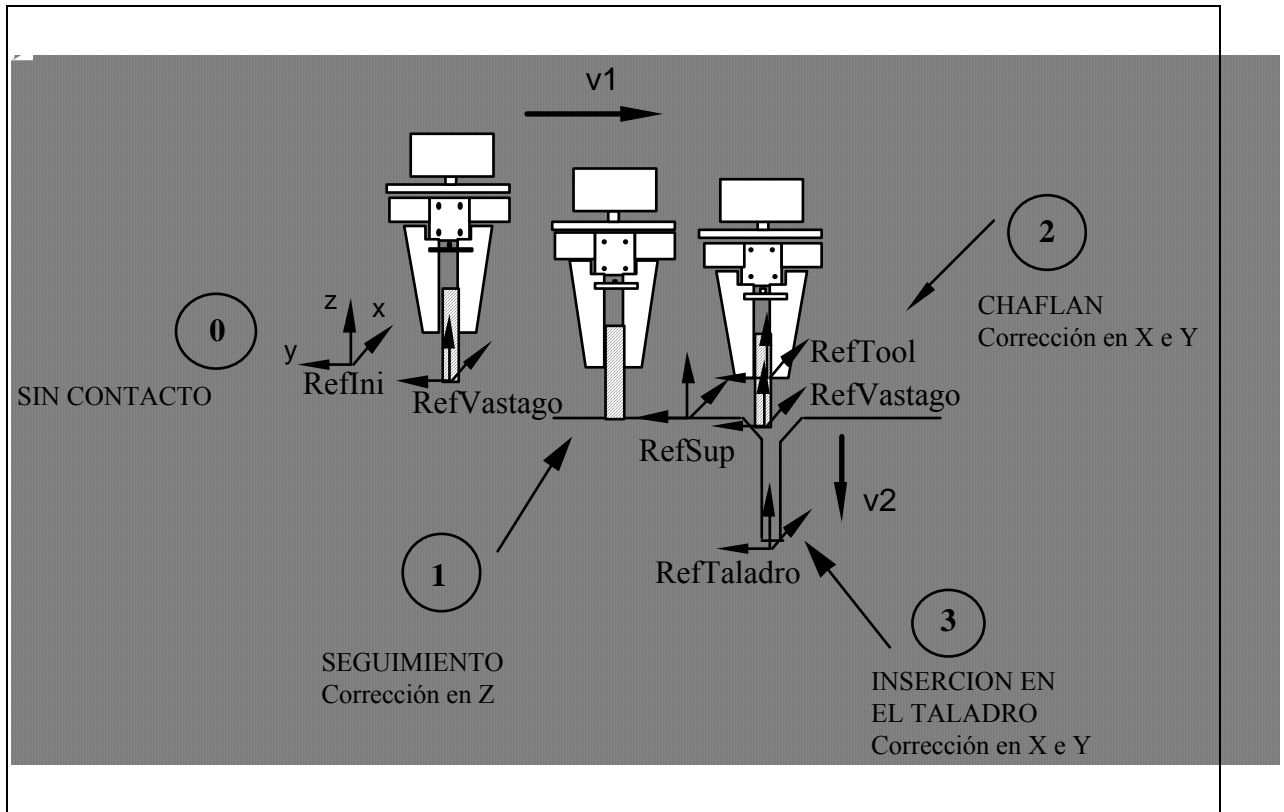


Figura 11. Inserción de un vástago en un taladro.

El segundo movimiento es el de inserción vertical propiamente dicha con condición de finalización de detección de una fuerza  $f_z$  superior a un valor umbral. La velocidad nominal de consigna durante la inserción es  $A_{v_n} = (0, 0, -v2, 0, 0, 0)^T$  y la consigna de fuerza durante la inserción es  $A_{f_c} = (0, 0, 1, 0, 0, Ind)^T$ . Este caso difiere de la especificación en el caso ideal (apartado 4). Así, la fuerza de consigna en Z se introduce para forzar la inserción en los momentos iniciales del movimiento evitando posibles atascamientos debido al rozamiento. El g.d.l. correspondiente a  $\delta\phi_z$  no es relevante para la tarea y, por tanto, no se especifica como g.d.l. de acomodación. En esta situación real no sería necesario especificar consigna de velocidad en Z ( $v2$ ), ya que la corrección debido a la fuerza provoca movimiento en esa dirección.

Durante la inserción deben producirse correcciones del movimiento nominal en las direcciones X e Y ( $\delta x$ ,  $\delta y$ ) en función de fuerzas en dichos ejes, y para evitar bloqueos se calculan correcciones en torno a los ejes X e Y ( $\delta\psi_x$ ,  $\delta\theta_y$ ) a partir de las fuerzas  $f_y$  y  $f_x$ , respectivamente.

De acuerdo con lo anterior, la matriz de corrección elegida es:

$$D = \begin{pmatrix} -0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.2 & 0 & 0 & 0 \\ 0 & -0.01 & 0 & 0 & 0 & 0 \\ 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

En la figura 12 se representan las fuerzas en cada uno de los tramos del movimiento. En la ejecución de la tarea se distinguen cuatro zonas:

- 0.- Todos los esfuerzos son aproximadamente nulos.
- 1.- Únicamente se ejercen fuerzas a lo largo del eje Z.
- 2.- Se producen fuerzas de contacto en la punta del vástago, que provocan el desplazamiento y el giro del órgano terminal de forma que no se producen atascamientos.
- 3.- Las fuerzas sobre las paredes del orificio ( $f_x$ ,  $f_y$ ) disminuyen debido a las correcciones del movimiento y el robot se detiene cuando la fuerza sobre el fondo del taladro es la fijada como umbral.

En la tabla 4 se muestra la programación de la tarea de inserción. Se supone que el robot porta un vástago con una referencia asociada *RefVastago* y parte de una situación inicial representada por la referencia *RefIni*. Se quiere realizar la inserción de tal manera que se alcance la situación *RefTaladro*. Para ello se definen dos subtareas, *Aproximación* e *InsercionVertical*. La primera realiza la aproximación a la superficie hasta alcanzar la referencia *RefSup*, permitiendo el deslizamiento del vástago sobre la superficie hasta dicha referencia. Con la subtaska *InsercionVertical* se lleva a cabo la inserción del vástago en el agujero hasta alcanzar el fondo del mismo. No es necesario que *RefSup* y *RefTaladro* estén definidas con exactitud, ya que el sistema de control modifica la trayectoria en función de los esfuerzos medidos.

En todas las subtareas se utiliza el procedimiento *MovAcom*, que es el que implementa el control de esfuerzos (véase tabla 2).



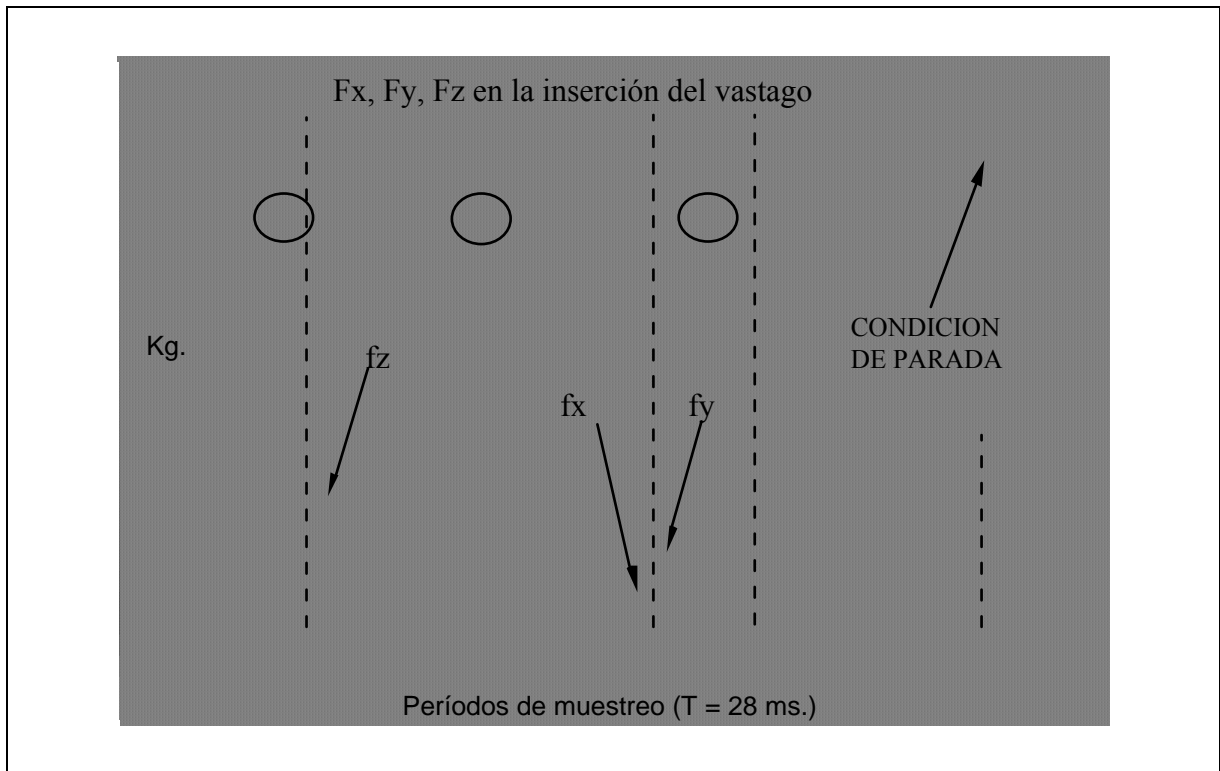


Figura 12. Fuerzas medidas durante el movimiento.

## 7 CONCLUSIONES

En este artículo se ha abordado la programación y el control de movimientos acomodaticios con un sensor de fuerza en la muñeca de un robot PUMA 560 (MKII). El sensor utilizado ha sido diseñado en el propio entorno universitario y actualmente se experimenta con otro comercialmente disponible con el que se espera obtener mejores resultados.

Se ha presentado un método de especificación de tareas en las que pueden utilizarse este tipo de movimientos y se han definido primitivas del lenguaje de programación adecuadas a la misma. Asimismo, se ha propuesto un esquema de control de movimientos acomodaticios basado en la idea de amortiguamiento generalizado. Este esquema tiene la ventaja de permitir la programación y el control de movimientos acomodaticios en robots industriales que poseen un sistema de control de posición/velocidad, utilizando el propio controlador del robot y añadiendo un bucle externo para el control de esfuerzos.

```

MODULE Insercion ;
(* Definicion de referencias e inicializacion de variables *)
.....
PROCEDURE Aproximacion (RIn,RDes:Referencia; Fcons:Fuerza);
(* Este procedimiento realiza la aproximacion a una superficie con control de fuerza
  RIn : Referencia inicial
  RDes : Referencia destino
  Fcons : Esfuerzos de consigna deseados *)
VAR Tr : Transformacion; Vel: VelTrasRot ;
BEGIN
  Tr := TransfDifRef (RDes,RIn) ;
  Velocidad (Vel,10.0,0.0) ;
  RefAcom (RefTool) ;
  MatCorreccionDiag (D,0.0,0.0,-0.05,0.0,0.0,0.0) ;
  MovAcom (Tr, Vel,Fcons,D,DestAlcanzado)
END Aproximacion;

PROCEDURE InsercionVertical (RIn,RDes:Referencia; Fcons:Fuerza);
(* Este procedimiento realiza una insercion vertical con control de fuerza.
  RIn : Referencia inicial
  RDes : Referencia destino
  Fcons : Esfuerzos de consigna deseados *)
VAR Tr : Transformacion; Vel: VelTrasRot ;
BEGIN
  Tr := TransfDifRef (RDes,RIn) ;
  Velocidad (Vel,5.0,2.0) ;
  RefAcom (RefTool) ;
  MatCorreccion (D, -0.2,0.0,0.0,0.0,0.0,0.0,
                  0.0,-0.2,0.0,0.0,0.0,0.0,
                  0.0,0.0,-0.2,0.0,0.0,0.0,
                  0.0,-0.01,0.0,0.0,0.0,0.0,
                  0.01,0.0,0.0,0.0,0.0,0.0,
                  0.0,0.0,0.0,0.0,0.0,0.0);
  MovAcom (Tr, Vel,Fcons,D,Fz(Mayor,5.0))
END InsercionVertical;
BEGIN (* Programa Principal *)
  (* Ejecucion de la aproximacion con consigna F *)
  DefFuerza (F,Ind,Ind,1.0,Ind,Ind,Ind) ;
  Aproximacion (RefIni,RefSup,F) ;
  (* Actualizacion de la Referencia RefSup *)
  T := LeeValorRef (RefTool) ;
  T1 := CompTransf (T, TVastago) ;
  RefSup := ValorRef (T1) ;
  (* Ejecucion de la insercion con consigna F *)
  DefFuerza (F,0.0,0.0,1.0,0.0,0.0,Ind) ;
  InsercionVertical (RefSup,RefTaladro,F) ;
  (* Actualizacion de la Referencia RefTaladro*)
  T := LeeValorRef (RefTool) ;
  Tfinal := CompTransf (T, TVastago) ;
  RefTaladro := ValorRef (Tfinal) ;
END Insercion.

```

Tabla 4. Programación de una tarea de inserción

Los resultados experimentales obtenidos permiten comprobar la posibilidad de realizar este tipo de tareas con un robot industrial como el mencionado. Una mayor precisión en las medidas del sensor, así como un período de muestreo inferior al impuesto por el propio robot permitirían mejorar la precisión en el mantenimiento de las fuerzas de consigna especificadas.

La aplicabilidad de este tipo de movimientos puede encontrarse en tareas de montaje de precisión, que no podrían realizarse con la precisión propia del robot. Otro campo de aplicación es la de operaciones en las que el robot y la herramienta que porta deben ejercer alguna fuerza sobre los objetos, como por ejemplo en tareas de mecanizado, desbarbado o pulido. El guiado de un robot para memorización de trayectorias es otra de las tareas en las que la realimentación y el control de esfuerzos resulta aplicable; permite al usuario dirigir el robot midiendo la fuerza ejercida sobre él y generando un movimiento en la dirección indicada por dicha fuerza. La posibilidad de obtener la geometría de los objetos durante el seguimiento permite que también puedan ser utilizadas en el reconocimiento de los mismos.

Un inconveniente del método de especificación presentado es que los parámetros a definir en una tarea son altamente dependientes de la misma, lo que dificulta en alguna medida su programación. Por ello se ha iniciado la implantación de un método de especificación de tareas a nivel de geometría de los objetos en contacto. Al tratarse de especificaciones a nivel geométrico, la utilización de un sistema de modelado geométrico puede facilitar su programación.

## **AGRADECIMIENTOS**

Este trabajo ha sido financiado parcialmente por CICYT, proyecto ROB91-0949.

## **REFERENCIAS**

- [1] Dégoulange E., Dauchez P., Pierrot P. <<Force Control of an Industrial Puma 560 Robot under Environmental Constraints Implementation Issues and Experimental Results>>. *1993 IEEE International Conference on Robotics and Automation*, pp 213-218, Atlanta, May 1993.
- [2] De Schutter J., Van Brussel H. <<Compliant Robot Motion I. A Formalism for Specifying Compliant Motion Tasks>>. *The International Journal of Robotics Research*, Vol. 7, No. 4, pp. 3-17, August 1988.
- [3] De Schutter J., Bruyninckx H. <<Model-Based Specification and Execution of

- Compliant Motion>>. *Tutorial M6 en IEEE Int. Conference on Robotics and Automation*, Nice, May 1992.
- [4] Luo Z.W., Ito M. <<Control Design of Robot for Compliant Manipulation on Dynamic Environments>>. *IEEE Trans. on Robotics and Automation*. Vol. 9, No. 3 June 1993.
  - [5] Mason M.T. <<Compliance and Force Control for Computer Controlled Manipulators>>. *Robot Motion: Planning and Control*. Brady et al editors, MIT Press 1982.
  - [6] McCormick W., Schwartz H.M. <<An Investigation of Impedance Control for Robot Manipulators>>. *The International Journal of Robotics Research*, Vol. 12, No. 5 pp. 473-489, October 1993.
  - [7] Montano L., Tardós J.D., Sagüés C., Neira J. <<Entorno de programación y control para un sistema robótico multisensorial>>. *1º Congreso de la Asociación Española de Robótica*, pp. 365-375, Zaragoza, Nov, 1989.
  - [8] Paul R.P. *Robot Manipulators: Mathematics, Programming and Control*. MIT Press, 1981.
  - [9] Raibert M.H., Craig J.J. <<Híbrido Position/Force Control of Manipulators>>. *ASME Journal of Dynamics Systems, Measurement and Control* Vol. 2, pp. 126-133, 1981.
  - [10] Shimano B., Roth B. <<On Force Sensing Information and its use in Controlling Manipulators>>, *Tutorial of Robotics. IEEE Computer Society*, pp. 254-261, 1983.
  - [11] Sinha P.R., Xu Y., Bajcsy R.K., Paul R.P. <<Robotic Exploration of Surfaces with a Compliant Wrist Sensor>> *The International Journal of Robotics Research*, Vol. 12, No. 2, pp. 107-120, April 1993.
  - [12] Smith R., Self M., Cheeseman P. <<Estimating uncertain spacial relationships in robotics>>. *Uncertainty in Artificial Intelligence 2.*, pp. 435-461, Elsevier Science Pub.1988.
  - [13] Whitney D.E. <<Historical Perspective and State of the Art in Robot Force Control>>. *The Int. Journal of Robotics Research*, Vol. 6 No.1, pp.3-14, 1987.