

# CONTROL DE FUERZA Y DE IMPACTO DE UN ROBOT CON CAMBIO DE PERÍODO DE MUESTREO

Ranko Zotovic (rzotovic@isa.upv.es)  
Departamento de Ingeniería de Sistemas y Automática  
Universidad Politécnica de Valencia

Pedro José García Gil (pggil@isa.upv.es)  
Departamento de Ingeniería de Sistemas y Automática  
Universidad Politécnica de Valencia

Ángel Valera Fernández (guiprog@isa.upv.es)  
Departamento de Ingeniería de Sistemas y Automática  
Universidad Politécnica de Valencia

Ángel Llosá Guillén (anllogui@upvnet.upv.es)  
Laboratorio de Automóviles  
Universidad Politécnica de Valencia

Manuel Cardó Redolar (mcardo@isa.upv.es)  
Departamento de Ingeniería de Sistemas y Automática  
Universidad Politécnica de Valencia

## Resumen

*Las tareas de control de fuerza de robots consisten de dos fases: el movimiento libre y el movimiento restringido. Al conmutar de una a otra ocurre un cambio en la dinámica del sistema. Además se controlan diferentes magnitudes (velocidad o posición vs. Fuerza), por lo que es lógico conmutar de regulador. Durante el movimiento restringido el sistema suele tener una respuesta mucho más rápida que durante el movimiento libre, por lo que puede ser conveniente utilizar diferentes períodos de muestreo para las dos fases. En este artículo se describe una plataforma experimental desarrollada y los resultados obtenidos para diferentes reguladores aplicado con y sin cambio de período de muestreo.*

**Palabras Clave:** Control de fuerza, impacto, impedancia, robot, período de muestreo, sistemas de tiempo real.

## 1 INTRODUCCIÓN

En numerosas aplicaciones los brazos robot entran en contacto físico con el entorno por lo que las características dinámicas del sistema cambian. En este caso los típicos métodos de control por posición/velocidad pueden ser insuficientes para garantizar el comportamiento del robot.

Dado que los materiales con los que el robot suele entrar en contacto en las aplicaciones industriales suelen ser muy rígidos, la constante de tiempo puede variar de varios ordenes de magnitud entre el movimiento libre y el restringido.

Por otra parte, para algunas tareas (grabado, pulido, desbarbado, ensamblaje, etc.) las fuerzas y pares con las que el brazo actúa sobre el entorno deben estar dentro de un rango determinado. Si las fuerzas de interacción sobrepasan un valor máximo, podría ocurrir un daño irreversible tanto para el entorno como para el robot y la herramienta. En el caso contrario, si las fuerzas son insuficientes, la tarea no se efectuaría satisfactoriamente.

El control de fuerza conlleva los siguientes problemas:

1. La típica tarea de control de fuerza contiene dos fases:
  - a.) Movimiento libre hasta el contacto entre el robot y la pieza. Requiere control por posición o velocidad.
  - b.) Movimiento restringido. Fase de control de fuerza. Comienza una vez el robot este en contacto con el entorno.

Como se ve, la tarea consiste en dos fases bien diferenciadas. Algunos autores [10] proponen una tercera intermedia para el control del impacto. Una solución puede ser utilizar estrategias de control diferentes en cada una de las tres fases. En este caso se deben establecer criterios para pasar de una fase a otra. Otra solución podría ser encontrar una estrategia que funcione bien para los tres casos. Pero este no es un problema trivial.

2. Los sensores de fuerza suelen ser susceptibles a ruido eléctrico debido a su gran sensibilidad. Esto trae problemas obvios de control debidos a la incertidumbre en la señal medida. Adicionalmente, hace prácticamente imposible la implementación de una componente derivativa del regulador de fuerza.
3. Los parámetros del proceso dependerán de las características geométricas y mecánicas del entorno. Estas en el caso general no son conocidas a priori. Esto significa que un diseño del regulador no es posible de antemano en muchos casos. Además las características mecánicas del entorno tienen un altísimo rango de variación.

Dado que es difícil encontrar una forma de control que responda bien a todos los inconvenientes enumerados, se han propuesto diversas estrategias de control, funcionando cada una de ellas en algunos casos.

A continuación se expondrán las principales estrategias de control de fuerza existentes con breves comentarios sobre cada una de ellas. Luego se enseñaran los resultados obtenidos aplicando los diversos reguladores con y sin cambio de periodo de muestreo.

## 2 CONTROL DE FUERZA

Durante las tres últimas décadas se han desarrollado diferentes estrategias para el control de fuerza, teniendo cada una sus respectivas ventajas y desventajas.

Las dos más importantes, que se van a tratar en este trabajo son el control explícito y el control implícito de fuerza. Este último a veces se denomina también el control por impedancia mecánica.

Existen otras estrategias como el control paralelo [7] y el control por bucle interior/ exterior.

En este apartado se va a realizar un breve análisis teórico de las dos principales estrategias de control de fuerza.

Se va a suponer, como se suele en la mayoría de los casos [8], que la fuerza de reacción del entorno es puramente elástica:

$$f = K_e(x - x_e) \quad (1)$$

Donde  $K_e$  y  $x_e$  son la rigidez y la coordenada del entorno respectivamente.

Se va a elegir por simplicidad el origen del sistema coordenadas en la superficie del entorno, con lo que se obtiene:

$$\begin{aligned} x_e &= 0 \\ f &= K_e x \end{aligned} \quad (2)$$

Asumiendo la función de transferencia en movimiento libre:

$$G_l = \frac{X(s)}{U(s)} = \frac{1}{ms^2 + bs} \quad (3)$$

Se puede deducir la función de transferencia por posición en movimiento restringido:

$$G_r = \frac{X(s)}{U(s)} = \frac{1}{ms^2 + bs + K_e} \quad (4)$$

y por fuerza:

$$G_f = \frac{F(s)}{U(s)} = \frac{K_e}{ms^2 + bs + K_e} \quad (5)$$

Asumiendo que la rigidez del entorno es conocida, el diseño del regulador se vuelve un problema trivial, por lo que en se va a suponer que es desconocida.

### 2.1. CONTROL POR FUERZA PURO (EXPLICITO)

El propósito de esta estrategia de control es sencillamente el seguimiento de la fuerza de referencia.

La única magnitud que se realimenta es la fuerza.

Un inconveniente del control explícito reside en la propensidad al ruido del sensor de fuerza. Esto hace poco viable la estimación de la derivada de fuerza.

También se debe resaltar la muy elevada rigidez mecánica de los entornos reales. Esto implica una dinámica muy subamortiguada y de respuesta extremadamente rápida.

A continuación se va a realizar un estudio de diferentes reguladores aplicados al control explícito de fuerza. En este trabajo se va a limitar al control lineal, siendo este el más común, aunque existan resultados en control no-lineal de fuerza [12] [9].

### 2.1.1. Regulador proporcional

Para el control proporcional de fuerza se obtiene la función de transferencia en bucle cerrado:

$$G(s) = \frac{f}{f_{ref}} = \frac{K_p K_e}{ms^2 + bs + (1 + K_p)K_e} \quad (6)$$

El sistema es siempre estable puesto que los coeficientes del denominador son todos positivos.

La ganancia es:

$$|G(s)| = \lim_{s \rightarrow 0} G(s) = \frac{K_p K_e}{(1 + K_p)K_e} < 1 \quad (7)$$

De lo que se deduce que el regulador proporcional no alcanza el valor de referencia de fuerza.

El sistema es subamortiguado cuando el discriminante de la ecuación característica es negativo:

$$b^2 - 4m(1 + K_p)K_e < 0 \quad (8)$$

Al aumentar la constante proporcional el sistema se vuelve aun más subamortiguado.

Ajustar la constante proporcional es complicado dado su efecto ambiguo: cuanto más es elevado su valor el sistema se vuelve más subamortiguado, y al reducir su valor, se aumenta el error de posición.

### 2.1.2. Regulador proporcional con prealimentación de fuerza

En este caso la acción de control es:

$$u = F_{ref} + K_p (F_{ref} - F_{real}) - K_v \dot{x} \quad (9)$$

La función de transferencia en bucle cerrado es:

$$G(s) = \frac{f}{f_{ref}} = \frac{(1 + K_p)K_e}{ms^2 + bs + (1 + K_p)K_e} \quad (10)$$

Se puede ver que con respecto al regulador proporcional solo cambia el numerador.

La ganancia del sistema es uno, por lo que el punto de la fuerza de referencia será alcanzada.

Un inconveniente de este regulador reside en la necesidad de tener un buen modelo del actuador para implementar la prealimentación de fuerza. Errores pueden ocurrir debido a las no-linealidades del motor o deformaciones del robot.

### 2.1.3. El amortiguamiento

Dada la elevada propensión al ruido del sensor de fuerza, la clásica acción de control derivativa no es aplicable. Se suelen utilizar dos alternativas.

La primera consiste en pasar el error de fuerza por un filtro paso bajo. En este caso la acción de control sería:

$$U(s) = F_{ref}(s) + (K_p + K_d s) \left( F_{ref} - \frac{a}{s-a} F_{real} \right) \quad (11)$$

La segunda posibilidad es utilizar la velocidad en lugar de la derivada de la fuerza para amortiguar el sistema:

$$u = F_{ref} + K_p (F_{ref} - F_{real}) - K_v \dot{x} \quad (12)$$

En el caso en el que la deformación del entorno es puramente elástica, como se ha supuesto en este trabajo, la velocidad es directamente proporcional a la derivada de la fuerza.

### 2.1.4. Regulador integral:

Se puede deducir la f.d.t. en bucle cerrado:

$$G(s) = \frac{f}{f_{ref}} = \frac{K_i K_e}{ms^3 + bs^2 + K_e s + K_i K_e} \quad (13)$$

A diferencia del regulador proporcional, en este caso es posible la pérdida de estabilidad del sistema puesto que es de tercer orden.

Tiene dos ventajas sobre el proporcional:

- Garantiza el error cero.
- Dado que el integrador es un filtro paso bajo, tiene mejor robustez en cuanto a ruido.

Es importante destacar que en el control de fuerza el valor de referencia suele ser constante, por lo que la limitación del ancho de banda del integrador no tiene efecto sobre las prestaciones del sistema.

### 2.1.5. Regulador proporcional- integral

Se puede deducir la f.d.t. en bucle cerrado:

$$G(s) = \frac{f}{f_{ref}} = \frac{K_p K_e s + K_i K_p}{ms^3 + bs^2 + (1 + K_p)K_e s + K_i K_p} \quad (14)$$

Este tipo de control no tiene ninguna ventaja que no tenga el proporcional o el integral y tiene las desventajas de los dos.

### 2.1.6. Conclusiones sobre el control explícito

El regulador proporcional por si solo es insuficiente. Para garantizar que se va alcanzar la referencia existen dos formas: prealimentación de fuerza o acción de control integral.

La primera requiere un buen modelo del robot y también que este sea rígido.

La segunda podría volver el sistema inestable.

Hay dos formas de amortiguar el sistema. La más común es la realimentación de la velocidad.

El control de fuerza explícito sería trivial de no ser por el inconveniente de que se parte del movimiento libre que requiere otro tipo de regulador (por velocidad o por posición). También se debe tener en cuenta la posibilidad de pérdida de contacto debida, por ejemplo, a las irregularidades en el entorno.

También hay que tener en cuenta que muchas veces no conocemos las características del entorno para diseñar el regulador

## 2.2. CONTROL DE IMPEDANCIA MECÁNICA (IMPLICITO)

El control explícito de fuerza es apropiado únicamente para la fase de movimiento restringido, sea cual sea el regulador utilizado de los mencionados en el apartado anterior.

Esto significa que se debe utilizar otro regulador (típicamente por velocidad) para la fase de movimiento libre. También es necesario conmutar entre las dos acciones de control, lo que puede causar problemas adicionales como modo deslizante o ciclos límite.

Para evitar estos inconvenientes, Nevile Hogan [5] ha propuesto como alternativa una técnica que utiliza el mismo regulador para las dos fases, llamada el control por impedancia mecánica.

La impedancia mecánica se define como la relación entre la fuerza aplicada y la velocidad del sistema:

$$Z(s) = \frac{F(s)}{v(s)} = \frac{Ms^2 X + BsX + KX}{sX} = Ms + B + \frac{K}{s} \quad (15)$$

Durante el movimiento restringido, el objetivo del control por impedancia no es seguir una referencia de fuerza o de posición, sino hacer que el proceso tenga el comportamiento dinámico deseado.

Por esta razón se llama también control implícito de fuerza, dado que la fuerza no se controla directamente.

Variando los parámetros de la impedancia mecánica del brazo (masa, amortiguamiento y rigidez) es posible en buena medida controlar la interacción con el entorno.

Como es evidente, el efecto de cada uno de los parámetros es intuitivo, lo que aporta una ventaja adicional a esta forma de control.

Sin embargo, su principal ventaja con respecto al control explícito de fuerza reside en el hecho de que el mismo regulador puede ser utilizado tanto para la fase de movimiento libre como la de movimiento restringido. Esto es posible gracias al hecho que los reguladores de impedancia activa son básicamente en reguladores por posición.

La principal limitación del control implícito es que no garantiza que se va alcanzar la referencia de fuerza. Existen algunos problemas más, como la medición de los desplazamientos, velocidades y aceleraciones muy pequeñas durante la fase de movimiento de restringido.

### 2.2.1. Formas de definir la impedancia mecánica en el control de fuerza:

Diferentes autores han utilizado formulaciones variadas para la impedancia mecánica.

Las más comunes están enumeradas en [10]:

$$\begin{aligned}
 F_{ext} &= M_d(\ddot{x} - \ddot{x}_{ref}) + B_d(\dot{x} - \dot{x}_{ref}) + K_d(x - x_{ref}) \\
 F_{ext} &= M_d\ddot{x} + B_d\dot{x} + K_d(x - x_{ref}) \\
 F_{ext} &= M_d\ddot{x} + B_d(\dot{x} - \dot{x}_{ref}) + K_d(x - x_{ref}) \\
 F_{ext} &= B_d\dot{x} + K_d(x - x_{ref}) \\
 F_{ext} &= B_d(\dot{x} - \dot{x}_{ref}) + K_d(x - x_{ref})
 \end{aligned} \tag{16}$$

Donde  $M_d$ ,  $B_d$ , y  $K_d$ , son la masa deseada el amortiguamiento deseado y la rigidez deseada respectivamente.

La primera formulación, de la que se pueden derivar todas las demás simplificando, se puede escribir de la siguiente forma:

$$\begin{aligned}
 f &= M_d\ddot{x} + B_d\dot{x} + K_d x - (M_d\ddot{x} + B_d\dot{x} + K_d x) = \\
 &= M_d\ddot{x} + B_d\dot{x} + K_d x - ref
 \end{aligned} \tag{17}$$

Es evidente que los valores de referencia de aceleración, velocidad y posición tienen el mismo efecto.

### 2.2.2. Técnicas para implementar la impedancia mecánica

En el control por impedancia se trata de imponer al sistema un comportamiento dinámico requerido.

Existen dos formas de lograrlo.

En el primer caso, llamado el control lineal, la acción de control ha de ser:

$$u = ref - (M_r\ddot{x} + B_r\dot{x} + K_d x) \tag{18}$$

El comportamiento del sistema obtenido es:

$$\begin{aligned}
 (M_r + m)\ddot{x} + (B_r + b)\dot{x} + K_r x &= \\
 = M_d\ddot{x} + B_d\dot{x} + K_d x = ref - f
 \end{aligned} \tag{19}$$

De esta forma se puede imponer una masa, amortiguamiento deseado al sistema.

La forma alternativa de implementar el control de impedancia es por dinámica inversa. La acción de control para el caso de un grado de libertad puede ser:

$$U(s) = f + \frac{ms^2 + bs}{M_d s^2 + B_d s + K_d} (ref - f) \tag{20}$$

La primera forma de implementar la impedancia mecánica es sencilla, computacionalmente poco costosa y no necesita sensor de fuerza. Sin embargo, no funciona bien para un robot de más de una articulación debido a que se trata de un regulador lineal aplicado a un proceso no-lineal.

La segunda forma se puede utilizar para un robot de más articulaciones. Sin embargo, es más compleja y necesita un sensor de fuerza. Cabe destacar que en esta técnica se compensa la fuerza, lo que puede generar problemas debido a los retardos y al ruido.

### 2.2.3. Conclusiones sobre el control implícito

Es una técnica alternativa que permite utilizar el mismo regulador para ambas fases de la tarea.

Su principal inconveniente es que no alcanza el valor de referencia de fuerza.

Sin embargo, puede imponer al sistema el comportamiento dinámico requerido, lo que es suficiente para muchas tareas.

En este trabajo se ha utilizado para el control del impacto, es decir de la fase transitoria al pasar al movimiento restringido.

## 3 SISTEMA DE CONTROL EN TIEMPO REAL

Actualmente, la tendencia predominante es la implementación de los algoritmos de control sobre sistemas digitales. Sin embargo, a menudo la implementación de estos algoritmos se lleva a cabo de una forma equivocada debido a un incorrecto conocimiento de los conceptos de sistema de tiempo real. Por otro lado, un buen entendimiento de estos conceptos permitirá una reducción y optimización de los recursos utilizados lo cual repercutirá en una reducción de costes.

El desarrollo de una aplicación software en ingeniería debería ser capaz de ejecutar algoritmos de control a muy bajo periodo de muestreo, debería ser fácil de usar, mientras que al mismo tiempo, fuera capaz de garantizar la calidad de unos determinados servicios, y todo ello a un bajo coste.

Recientes avances en la tecnología del software han permitido el diseño y desarrollo de los sistemas empotrados. En particular, los sistemas empotrados para el control de procesos es un campo que necesita

una especial atención y esfuerzo en mejorar la funcionalidad y servicios basados en la tecnología de computadores de propósito general.

El principal aspecto en el desarrollo de sistemas empotrados para el control de procesos, se deriva de la necesidad de soportar una ejecución de tiempo real, fácil acceso a dispositivos de I/O, y tolerancia a fallos [6], con especial énfasis en el mínimo hardware y software.

El uso de los lenguajes y herramientas apropiados para el desarrollo de los sistemas empotrados de control y sus aplicaciones es una de las principales características del diseño e implementación de los algoritmos de control. Desde este punto de vista, el desarrollo de soluciones con lenguajes de tiempo real, como Ada, pueden ser usados posteriormente como lenguajes de desarrollo para aplicaciones de control de tiempo real.

El uso de soluciones específicas a través de plataformas comerciales, requiere un alto coste tanto del desarrollo como en el mantenimiento de la aplicación, adicionalmente requiere un alto conocimiento del sistema. El uso de código libre, como puede ser Real-Time Linux, puede mejorar el desarrollo del sistema en muchos aspectos. Otra alternativa es el uso de código específico desarrollado para sistemas empotrados con mínimos requerimientos, como ORK [7], Marte OS [1], los cuales permiten una programación tanto en lenguaje Ada como en C. En este trabajo, se ha optado por Marte OS, como plataforma para la implementación de la solución de control.

MaRTE OS es un kernel de tiempo real para el desarrollo de aplicaciones empotradas de tiempo real. El cual sigue el estándar de tiempo real mínimo POSIX.13, facilitando la posibilidad de programación tanto en lenguaje C como en lenguaje Ada. Esto permite el desarrollo de aplicaciones de tiempo real tanto en C como en Ada. También puede ser desarrollada aplicaciones implementadas con ambos lenguajes, permitiendo esto la facilidad del manejo de tareas en Ada, con el acceso a bajo nivel que proporciona C.

MaRTE OS se compone de una plataforma de desarrollo (host), basada en un PC Linux con los compiladores gnat y gcc (ambos de libre distribución), y una plataforma hardware de ejecución basada en un PC 386 o superior (target). Esta última plataforma requiere de una disquetera, o similar, para la carga de los programas a ejecutar, pero no requiere ni de disco duro, ni de sistema operativo.

La comunicación con el sistema empotrado podrá realizarse tanto vía RS-232 como mediante Ethernet.

Esto permitirá el envío de comandos procedentes de otras aplicaciones así como la transmisión de otros datos o variables del sistema para su posterior análisis y monitorización.

## 4 DESCRIPCIÓN DE LA PLATAFORMA EXPERIMENTAL

El sistema está compuesto por un eje lineal de la compañía Quanser que tiene acoplado en un extremo un sensor de fuerza JR3.

El control se realiza desde el PC target que contiene las tarjetas de interfaz con el eje y con el sensor.

Aparte, son necesarios dos PC's: uno para compilar los programas en ADA bajo MARTE (el host) y otro para guardar y visualizar los datos experimentales.

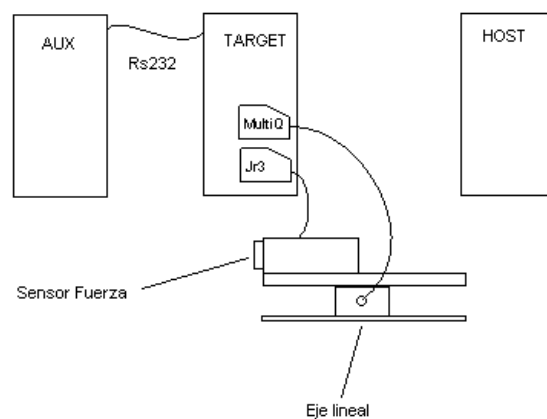


Figura 1: Esquema del sistema

A continuación se describen los componentes de forma más detallada:

*Sistema:*

- Quanser's MultiQ motion control board (Plataforma Quanser)

Como eje lineal tenemos la plataforma Quanser, que se desplaza mediante un motor de corriente continua que actúa gracias a una rueda dentada sobre un engranaje de cremallera. El motor es controlado mediante la tarjeta de adquisición de datos MultiQ. Para leer la posición relativa de éste, la plataforma lleva incorporado un encoder, que nos manda la información de la posición a la misma tarjeta para realizar el control en bucle cerrado.



Figura 2: Eje lineal con sensor de fuerza.

- Sensor de fuerza JR3:

El sensor es un dispositivo de aluminio con componentes analógicos y digitales. En la hoja de metal donde medimos la fuerza, existen unas galgas extensiométricas dispuestas en forma de cruz de tal manera que, midiendo la deformación de la hoja, podemos calcular la fuerza  $(x,y,z)$  y el momento  $(x,y,z)$  que actúa sobre ella. Esta información analógica es convertida a digital por un circuito electrónico en el propio sensor, y enviada en serie al receptor electrónico DSP, una tarjeta en el ordenador *Target*. Los datos que son mandados a la tarjeta son muestreados por el sensor a una frecuencia de 8 kHz.

*Host:*

Es un PC con un procesador Pentium II a 350 Mhz con 124 MB Ram. Como sistema operativo tiene Red Hat Linux 6.2 corriendo sobre un kernel 2.2.14. Este ordenador se va a utilizar para desarrollar todo el código del experimento. En este PC se tiene instalado el compilador de MarteOS 1.2. Con este software serán compilados los programas y volcados en un disquete.

Para el desarrollo de éstos programas se han utilizado las herramientas GPL (gcc, gnat, emacs) disponibles en la distribución anteriormente comentada.

*Target:*

Sobre este PC se ejecutará el programa compilado en el *Host*. Es un Pentium II a 350 Mhz con 262 MB de Ram

Funcionará a modo de sistema de control del proceso. Para ello posee dos tarjetas de adquisición de datos:

- Tarjeta de adquisición de datos de Quanser PCI MultiQ

Esta tarjeta es utilizada para obtener información del encoder y actuar sobre el motor de corriente continua. Tiene 16 entradas analógicas de 14 bits

con rangos programables de  $\pm 5$  y  $\pm 10$  Voltios, 4 salidas analógicas de 12 bits con un rango de  $\pm 10$  y 6 entradas para encoders ópticos.

- DSP Based Interface System

Esta tarjeta convierte los datos digitalizados que son recibidos de las galgas extensiométricas del sensor en medidas digitales de fuerza y momento. La adquisición de datos desde el sensor se realiza a una velocidad de 8 kHz. En caso de querer obtener la señal filtrada digitalmente, la frecuencia baja a 500 Hz, reduciendo  $\frac{1}{4}$  por cada filtro.

Éste arranca directamente desde el disquete y carga Marte como sistema operativo, funcionando como un sistema empotrado, utilizando así con nuestro programa todos los recursos disponibles en el PC. Ésto crea limitaciones en nuestro sistema, debido a que no podemos guardar datos en ficheros según el estandar POSIX. Los drivers de entrada y salida de las tarjetas han sido desarrollados en C y compilados con el programa principal. Para el almacenaje de los datos el elemento *Auxiliar*.

*Auxiliar:*

El *Target* se conecta vía RS-232 al PC *Auxiliar*. La información recibida se guarda en un fichero de datos ASCII mediante un programa desarrollado en C.

El PC Auxiliar tiene Windows 2000 como sistema operativo, de forma que es posible analizar los datos del fichero directamente desde Matlab. Esta última fase se puede hacer también desde Linux y procesar los datos en Scilab ([www.scilab.org](http://www.scilab.org)), teniendo así todo el conjunto de nuestro sistema compuesto por software Open Source.

---

## 5 RESULTADOS EXPERIMENTALES

Para los experimentos se ha implementado un regulador proporcional descrito en la ecuación [9]. No se utiliza la prealimentación descrita antes debido a que se ha verificado que la relación entre la acción de control y la fuerza no es lineal. Por lo tanto, no se alcanza la fuerza de referencia (100 g).

La velocidad se estimó mediante la fórmula :

$$v = \frac{x - x \cdot z^{-1}}{t_m} \quad (21)$$

De esta forma se puede utilizar la misma formulación para diferentes periodos de muestreo.

En movimiento libre se ha utilizado un regulador proporcional de velocidad con constante proporcional

0.00005 y un periodo de muestreo de 5 ms. La velocidad de referencia ha sido de 20000 pulsos de encoder/periodo de muestreo, lo que equivale aproximadamente a 25 m/s.

Para el control de fuerza se utilizó el regulador descrito antes con los siguientes parámetros:  $K_p=0.01$  y  $K_v=0.0001$ .

Se han hecho pruebas variando con tres diferentes periodos: 5 ms, 1 ms y 0,5 ms.

Los resultados están representados en las siguientes gráficas:

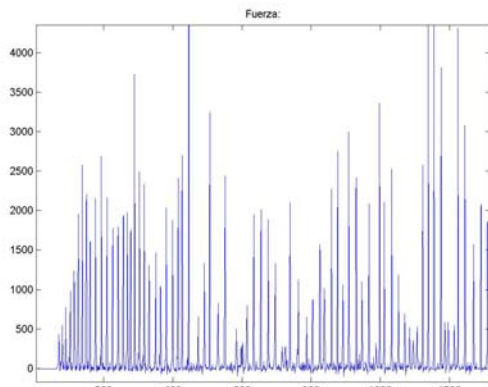


Figura 3: Fuerza con  $T_m=5ms$

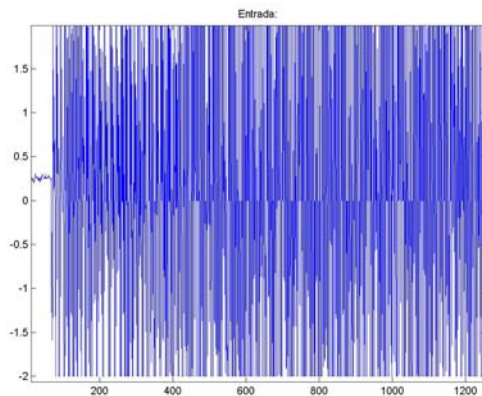


Figura 4: Acción de control con  $T_m=5ms$

Se ve que el sistema alcanza picos elevados de fuerza, pierde el contacto y posiblemente se vuelve inestable. La acción de control se satura continuamente.

Los resultados para el periodo de 1 ms son los siguientes:

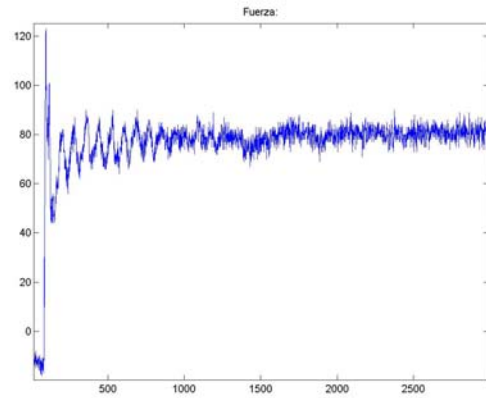


Figura 5: Fuerza con  $T_m=1ms$

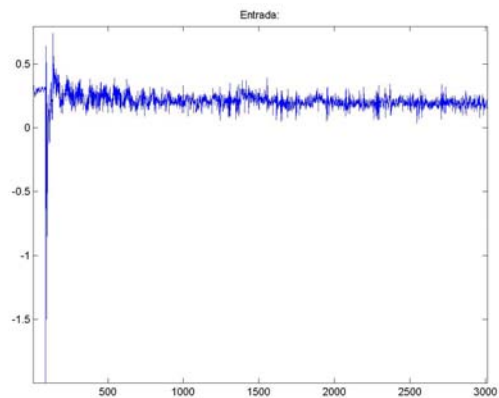


Figura 6: Acción de control con  $T_m=1ms$

Y con un periodo de 0,5 ms:

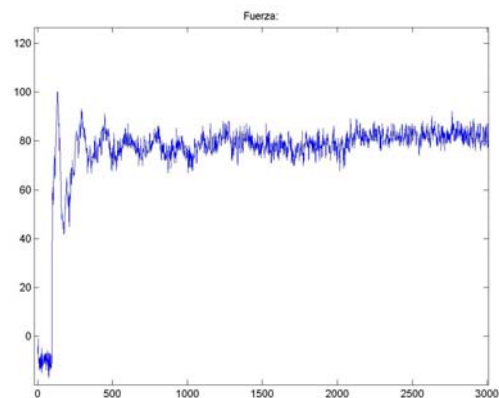


Figura 7: Fuerza con  $T_m=0.5ms$



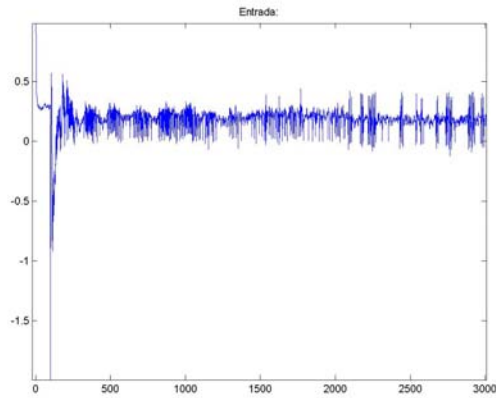


Figura 8: Acción de control con  $T_m=0.5\text{ms}$

Se ve que en ambos casos se reducen los picos de forma drástica, que se evitan las pérdidas de contacto y que el sistema es estable.

## 6 CONCLUSIÓN

Se ha implementado una plataforma experimental para el control de fuerza de una articulación con posibilidad de cambio de período de muestreo.

Para ello se ha utilizado un módulo lineal de Quanser y un sensor de fuerza de JR3.

El control se ha realizado bajo el sistema operativo de tiempo real MARTE con el lenguaje de programación ADA. Los drivers de las tarjetas se han desarrollado en C.

Se han realizado experimentos utilizando diferentes períodos el movimiento libre y el movimiento restringido, puesto que el cambio de en la dinámica del sistema es muy grande de un caso al otro.

Los resultados experimentales demuestran que el uso del mismo período para las dos fases puede tener consecuencias muy graves durante el movimiento restringido, como picos de fuerza muy elevados, pérdidas de contacto y hasta de estabilidad.

También se ha verificado que la reducción del período de muestreo en general mejora los resultados sobretodo en cuanto a sobreoscilación.

### Agradecimientos

Queremos agradecer a don Pedro Albertos y al Proyecto CICYT DPI2002-04432 por haber hecho posible este trabajo.

## Referencias

- [1] Aldea M. and González M (2001). "MaRTE OS: An Ada Kernel for Real-Time Embedded Applications". *Proceedings of the International Conference on Reliable Software Technologies, Ada-Europe*.
- [2] Chiaverini, S., Sciavicco, L., (1994), "Force/ Position Regulation of Compliant Robotic Manipulators", *IEEE Transactions on Automatic Control*, Vol. 39, n° 3.
- [3] Chiaverini, S., Siciliano, B., Villani L. L., (1993), "The Parallel approach to Force/ Position Control of Robotic Manipulators", *IEEE Transactions on Robotics and Automation*, Vol. 9, n° 4.
- [4] Ferretti, G., Magnani, G., Rocco, P., (1995), "On the Stability of Integral Force Control in Case of Contact with Stiff Surfaces", *Journal of Dynamics Systems, Measurement and Control*, Vol. 117.
- [5] Hogan, N., (1985), "Impedance Control: An approach to manipulation", *Journal of Dynamics Systems, Measurement and Control*, Vol. 107.
- [6] Kopetz H., (2001), "Real-Time Systems. Design Principles for Distributed Embedded Applications", Kluwer Academic
- [7] Puente Juan A., José F. Ruiz, and Juan Zamorano (2000). An open Ravenscar realtime kernel for GNAT. In Hubert B. Keller and Erhard Ploedereder, editors, *Reliable Software Technologies-Ada-Europe 2000*, number 1845 in LNCS, pages 5-15. Springer-Verlag.
- [8] Sciavicco, L., Siciliano, B., (1993), " Modeling and Control of Robot Manipulators", McGraw-Hill Company Inc.
- [9] Seraji, H., (1995), " Nonlinear and adaptive Control of Force and Compliance in Manipulators", *The International Journal of Robotics Research*.
- [10] Volpe, R. , Khosla, P., (1993a), " A Theoretical and Experimental Investigation of Impact Control for Manipulators", *International Journal of Robotics Research*, Vol. 12, n° 4.
- [11] Volpe, R. , Khosla, P., (1993b), "A Theoretical and Experimental Investigation of Explicit Force Control Strategies for Robot Manipulators"- *IEEE Transactions on Automatic Control* Vol. 38, n° 11.
- [12] Xu, Y, Hollerbach, M. J., Ma, D., (1995), "Force and Contact Transient Control Using Nonlinear PD Control"- *IEEE Control Systems Magazine*, Vol. 15 n° 1.