

PLATAFORMA DE EXPERIMENTACIÓN DE CONTROLADORES PARA ROBOTS INDUSTRIALES

R. Romero, I. Alcalá y F.R. Rubio

Dept Ingeniería de Sistemas y Automática

Escuela Superior de Ingenieros de Sevilla

e-mail:(romevi,ismael,rubio)@cartuja.us.es

Resumen

El objetivo del proyecto es el desarrollo de un equipo de experimentación para controladores de robots industriales sustituyendo el controlador original del robot manipulador. La gestión de funciones auxiliares necesarias para el manejo real de un robot manipulador se trata de forma separada a la implementación de controladores en sí, facilitando la prueba de nuevas estrategias de control. Con idea de simplificar el proceso de implementación de controladores, se programa en un ordenador el controlador a experimentar utilizando diagramas de bloques de SIMULINK. A continuación, el controlador se carga en una tarjeta de control en tiempo real de la compañía dSPACE que adquiere las medidas de posición y aplica las señales de control al robot manipulador. Mediante software es posible: supervisar el control implementado en la tarjeta dSPACE, modificar sus parámetros y seguir la evolución de variables relevantes del sistema. Lo cual permitirá investigar nuevos controladores en una plataforma real.

Palabras clave: Robot manipulador, Control por ordenador

1 INTRODUCCIÓN

El interés en el estudio de nuevas metodologías de control avanzado en robótica industrial ha motivado la creación de una plataforma que permita ensayar controladores de forma sencilla en un robot manipulador. La finalidad del proyecto consiste en probar distintas técnicas de control en un robot industrial, para lo cual resulta interesante centrarse en la síntesis de controladores y evitar problemas asociados a la implementación física en un sistema real (alarmas, protecciones, etc). Además, una mayor facilidad de programación y rapidez en la implementación de leyes de control permite reducir el tiempo de validación de las técnicas de control diseñadas.

Estas consideraciones han llevado al empleo de un sistema basado en una representación del controlador en diagrama de bloques que se almacena en

una tarjeta de control en tiempo real instalada en un PC dotado con la herramienta SIMULINK de MATLAB para la elaboración de los controladores que son cargados en la tarjeta de control.

La tarjeta de control utilizada ha sido la DS1103 de la compañía dSPACE [2] que consta de un procesador PowerPC, un DSP esclavo TMS320F240 y una memoria RAM de 32 MB. En la Tarjeta dSPACE se ejecuta el controlador diseñado y se gestionan las posibles alarmas y señales digitales de los circuitos electrónicos, así mismo lleva a cabo la aplicación de las señales de control a los motores de un Robot Manipulador.

El robot empleado es el brazo manipulador RM-10 para entornos industriales [4, 5]. El robot RM-10 tiene 6 grados de libertad, todos ellos de revolución y con accionamiento indirecto con motores sin escobillas. El robot dispone originalmente de controladores PID para cada motor de los que se ha prescindido considerando el objetivo del proyecto. En cambio, se ha conservado la etapa de potencia y los sensores resolver de posición. Además de las no linealidades en las ecuaciones dinámicas del robot debidas a acoplamientos entre los distintos enlaces, el RM-10 presenta fenómenos de holgura y fricción en los ejes que motivan su utilización como equipo de desarrollo de controladores.

Este documento está organizado de la siguiente manera: En la sección 2 se comentan las modificaciones hardware realizadas en el robot manipulador y se describen las tarjetas de circuito impreso que han sido necesarias. La sección 3 explica el proceso de implementación de un programa de control en tiempo real a partir de un modelo en SIMULINK, quedando descritos los bloques que componen el modelo en la sección 4. Los programas para modificar variables del controlador y supervisar el control están comentados en la sección 5. Por último, las conclusiones y posibles vías futuras de expansión de la plataforma de ensayos están comentadas en la sección 6.

2 MODIFICACIONES HARDWARE REALIZADAS

Para conseguir establecer la conexión entre el ordenador encargado del control y el robot manipulador se han implementado circuitos de interfaz en tarjetas electrónicas. Las tarjetas conectan el armario de control del robot y la tarjeta de control en tiempo real *dSPACE* instalada en el ordenador. En el diseño de las tarjetas se han aislado las señales del robot manipulador y de la tarjeta *dSPACE* mediante el empleo de optoacopladores y amplificadores analógicos de aislamiento.

Las funciones desempeñadas en las tarjetas aparecen en la figura 1 y comprenden: adquisición de las medidas de posición de los ejes, adquisición de posibles alarmas del robot, aplicación de señales digitales al armario del robot necesarias para realizar el control desde el ordenador y aplicación de la señal de control a cada regulador de velocidad de los motores del robot manipulador, así como la conmutación entre el controlador original del *RM-10* y el control por ordenador. Existen tres tipos de tarjeta de circuito impreso descritos en los siguientes apartados.

2.1 TARJETA DE CONMUTACIÓN DEL CONTROL

Antes de efectuar ninguna modificación en el robot manipulador *RM-10* el control de los ejes lo realizaba el armario de control del robot. Con objeto de poder controlar el robot manipulador desde el ordenador, se ha construido una tarjeta que conmuta la señal de los motores entre el armario de control del *RM-10* y el PC. Esta tarjeta electrónica permite la conmutación entre dos estados: posibilitar/impedir control desde PC. Para impedir el control desde el ordenador la tarjeta de conmutación del control interrumpe la alimentación de las tarjetas desarrolladas dejando el robot con su control original aunque se indique lo contrario desde el software de control del ordenador.

2.2 CIRCUITO RESOLVER/ENCODER Y AISLAMIENTO DE SEÑALES ANALÓGICAS

Para cada eje del robot manipulador es necesaria una tarjeta electrónica de este tipo. La tarjeta consta de dos módulos claramente diferenciados según la aplicación que realizan:

- El primer módulo realiza la conversión de la señal resolver del sensor de posición del eje a señal de un encoder incremental de 1024 pulsos por vuelta, para su lectura directa con

la tarjeta *dSPACE*. Inicialmente se acondiciona la señal aplicando filtros pasivos RC, una vez filtrada la señal, se procede a generar las señales encoder. La conversión se basa en el circuito integrado AD2S90 de Analog Devices, que a partir de la señal de referencia del sensor resolver, y de las señales inducidas seno y coseno, realiza la conversión a señal de encoder. La salida transformada a señal de encoder incremental se aísla antes de conectarse a la entrada de la tarjeta *dSPACE* encargada de leer señales de este tipo. El aislamiento se realiza mediante optoacopladores de alta velocidad y para ello es necesaria una etapa amplificadora ya que las salidas de encoder incremental del AD2S90 no son capaces de suministrar suficiente corriente para activar la etapa de optoaislamiento conectada a la tarjeta *dSPACE*.

- El segundo módulo se encarga de aislar las salidas analógicas de la tarjeta *dSPACE* generadas por los algoritmos de control en el ordenador. El aislamiento está basado en el circuito integrado ISO122P de la firma Burr-Brown que permite aislar señales analógicas con un rango $\pm 10 V$ y con un ancho de banda de 50kHz.

En la figura 2 se muestra el diagrama de bloques de esta tarjeta.

2.3 TARJETA DE ENTRADAS-SALIDAS DIGITALES

El robot manipulador utiliza una serie de entradas y salidas digitales que permiten conocer el estado de funcionamiento del sistema.

Las entradas digitales a la tarjeta *dSPACE* provenientes del robot se corresponden con alarmas que permiten detectar fallos en la operación del robot manipulador. Las principales alarmas son:

- EM6. Emergencia. Tiene que estar a uno cuando el robot funciona normalmente y en caso de darse una emergencia se desactiva.
- IFR. Se activa cuando los frenos del robot manipulador quedan liberados, permitiendo comprobar el estado de los mismos.
- IN8. Alarma de la fuente de alimentación por alcance de máxima temperatura.
- IN9. Alarma del fusible (fuente de alimentación).
- IN10. Alarma de pérdida de fase en la fuente de alimentación.

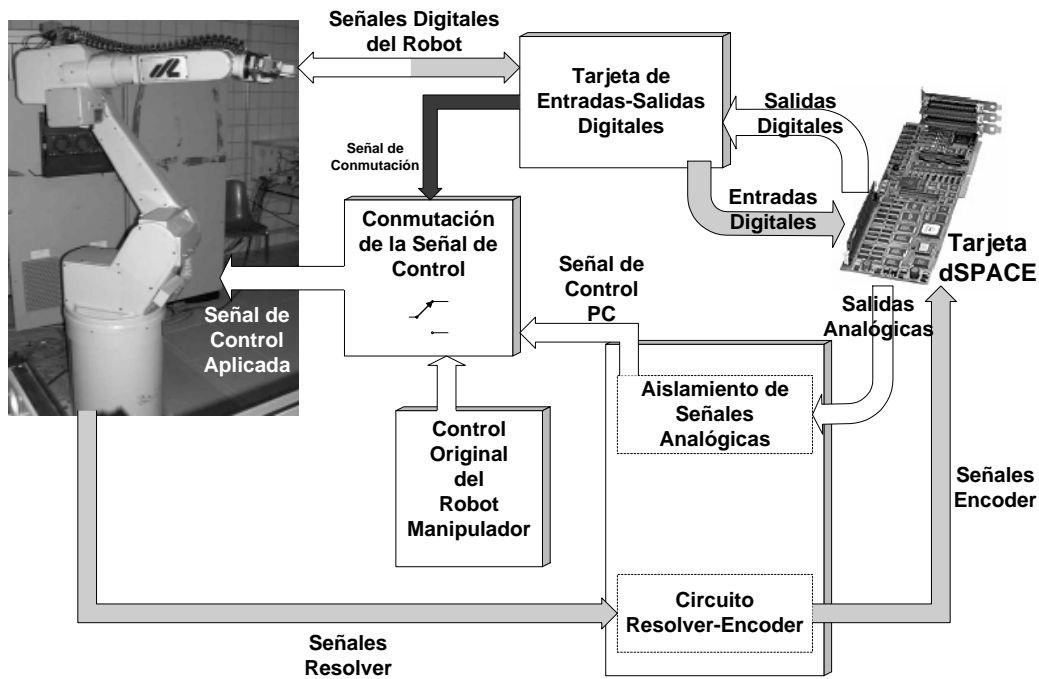


Figura 1: Conexión entre el Robot Manipulador *RM-10* y la tarjeta *dSPACE*.

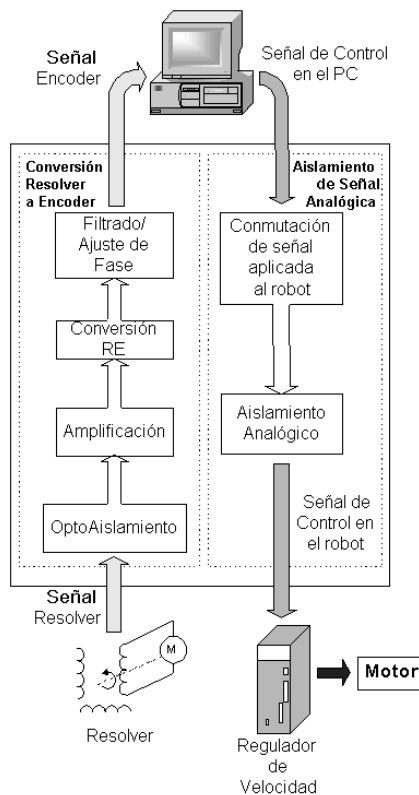


Figura 2: Esquema de la tarjeta de conversión resolver/encoder y aislamiento de señal de control analógica del motor.

En cuanto a las salidas digitales generadas en la tarjeta *dSPACE* se utilizan 3 señales que gobiernan la activación, frenado y limitación de potencia del robot manipulador:

ENABLE. Pone el robot en estado *activo* pudiendo darle consignas a los reguladores de velocidad de cada motor de los ejes del *RM-10*.

BRAKE. Cuando está activada, habilita los contactores de los frenos permitiendo el movimiento de los motores.

CIR5. Es la salida digital utilizada para limitar la potencia de los motores saturando la señal que les aplica a un 10% de la máxima señal de control que se puede aplicar al *RM-10*. Para poder aprovechar al máximo la capacidad de los motores tiene que estar desactivada.

Adicionalmente en los ejes más conflictivos (ejes 1, 2 y 3) se ha dotado al sistema de mayor seguridad con la introducción de una señal digital adicional (**VERIFICACIÓN CONTROL PC**) que comprueba la correcta conmutación de control del armario a control desde el ordenador.

3 IMPLEMENTACIÓN SOFTWARE DEL CONTROLADOR

En esta sección se explica el proceso de programación para probar el controlador diseñado.

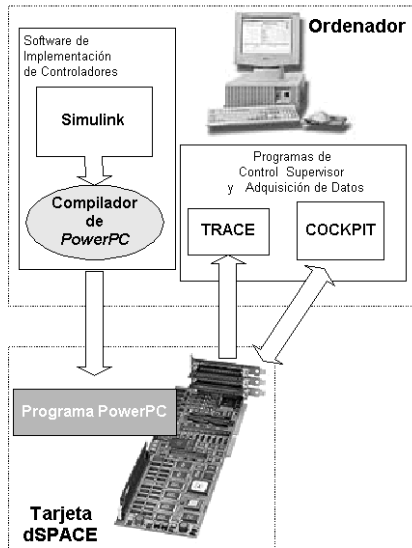


Figura 3: Esquema de programación y supervisión de controladores en la tarjeta *dSPACE*.

Dentro de este proceso hay que distinguir entre dos tipos de aplicaciones: aplicaciones utilizadas para implementar el controlador y aplicaciones encargadas de monitorización de resultados y sintonización de variables del controlador implementado. En la figura 3 aparece la relación entre las aplicaciones y la tarjeta de control en tiempo real *dSPACE*.

Las aplicaciones del primer tipo suponen el primer paso para implementar el control en el ordenador. Básicamente, se ha desarrollado un modelo de SIMULINK con distintos bloques (figura 4) que simplifica la introducción de un nuevo controlador. El controlador se construye con bloques de la biblioteca de SIMULINK dentro del denominado bloque de control de la figura 4, mientras que en el resto de bloques se desarrollan las tareas necesarias para la implementación física¹. La programación en bloques de SIMULINK proporciona un entorno familiar en la descripción de controladores que evita la utilización de lenguajes de programación de bajo nivel y la necesidad de un conocimiento por parte del ingeniero de control de las bibliotecas de la tarjeta *dSPACE*, lo cual incide de manera directa en un menor tiempo de implementación de controladores. Una vez introducido el controlador en el modelo de SIMULINK, se utiliza la herramienta *Real Time Workshop* MATLAB. Esta herramienta traduce el modelo a código C, lo compila y descarga de forma automática en la tarjeta *dSPACE*. La ejecución empieza cuando el modelo compilado está en la tarjeta *dSPACE*.

Finalmente, se utilizan las aplicaciones Cockpit y

¹Los bloques de SIMULINK del modelo se explican en la sección 4

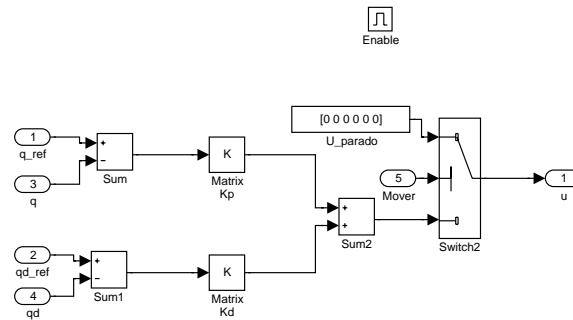


Figura 5: Ejemplo de controlador PD independiente para cada eje implementado en el bloque de control de SIMULINK en forma matricial.

Trace suministradas por *dSPACE* que pertenecen al segundo tipo de aplicaciones. El programa Cockpit de *dSPACE*, permite crear pantallas con controles para modificar en tiempo de ejecución los parámetros del modelo de SIMULINK y mostrar variables. Mientras que el programa Trace, se utiliza para supervisar los ensayos representando gráficamente la evolución de las variables del modelo.

4 BLOQUES DEL MODELO DE SIMULINK

Para establecer un entorno adecuado de programación del controlador en un bloque de SIMULINK, se ha dividido el modelo en distintos bloques encargados de funciones diferenciadas (ver figura 4). Esta separación en bloques consigue simplificar la programación del controlador haciéndola más transparente al emplear bloques que realizan funciones auxiliares independientes del controlador probado (protecciones, generación de trayectorias, alarmas, ...). Los bloques que componen el modelo de SIMULINK se describen en las subsecciones siguientes.

4.1 BLOQUE CONTROL

Está dedicado a la implementación de los controladores desarrollados. En este bloque de SIMULINK se incluyen los controladores aplicados desde el PC al Robot *RM-10*. En la figura 5, se puede apreciar un ejemplo de controladores PD en cada eje del Robot. El ejemplo de la figura 5 hace uso de bloques básicos de SIMULINK, aunque es posible utilizar bloques pertenecientes a *Toolbox* de control de MATLAB. El contenido del bloque de control puede ser sustituido por cualquier otro sin necesidad de modificar el resto de bloques (Máquina de estados, supervisor de alarmas, ...)

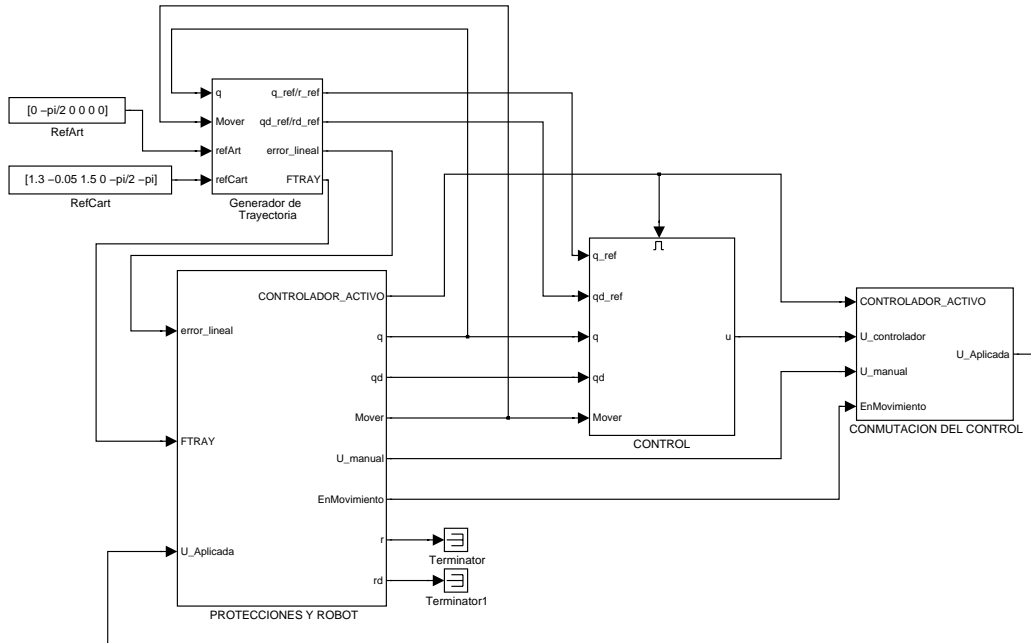


Figura 4: Modelo de SIMULINK con las funciones necesarias para el control del robot manipulador separadas en 4 bloques.

4.2 BLOQUE GENERADOR DE TRAYECTORIAS

Se encargará de suministrar las referencias de posición y velocidad de cada eje al bloque de control. Ha sido definido para permitir una representación tanto en el espacio articular como en el lineal. Genera trayectorias con un polinomio de orden cinco para obtener referencias suaves de posición y velocidad en cada eje del robot.

4.3 BLOQUE CONMUTACIÓN DEL CONTROL

Este bloque supone una medida de protección necesaria. En caso de que el controlador a prueba no consiga seguir la trayectoria de referencia es necesario utilizar otro controlador para detener el robot manipulador. Cuando el bloque de protecciones detecta este tipo de situación, cambia el controlador experimental por un PID incluido en el bloque de protecciones y robot. El PID está suficientemente probado y pasa a ser el encargado de parar el robot. El bloque de conmutación cambia la señal del bloque de control por la señal del controlador PID, que se envía al bloque de SIMULINK de salidas analógicas de la tarjeta *dSPACE*.

Además de situaciones de emergencia, la conmutación del controlador también se realiza cuando se quiere mover el robot en modo manual

4.4 BLOQUE PROTECCIONES Y ROBOT

Incluido en este bloque está la adquisición de las medidas de posición de los ejes, adquisición de entradas digitales y aplicación de salidas digitales al robot manipulador para poder moverlo

La parte dedicada a las protecciones del sistema, se encarga de comprobar el correcto funcionamiento y gobernar el bloque de conmutación. Las funciones de protección más relevantes están desarrolladas en el Supervisor de Alarmas y en la Máquina de Estados.

El Supervisor de Alarmas. En la figura 6 se puede ver la implementación de este bloque. Tiene como misión comprobar la existencia de alarmas y comunicarlo a la *Máquina de Estados* que se encargará de evolucionar al estado *Encendido* accionando los frenos y deshabilitando el bloque de control en caso de emergencia. Las alarmas detectadas se registran en variables indicadoras monitorizadas por los programas de supervisión (descritos en el apartado 5).

Dentro de este bloque Supervisor de Alarmas se han incluido las siguientes protecciones software:

- Límite de ángulo en las articulaciones: genera una alarma cuando el ángulo de una articulación queda fuera de su rango. Los rangos fijados para cada eje pueden leerse en la tabla

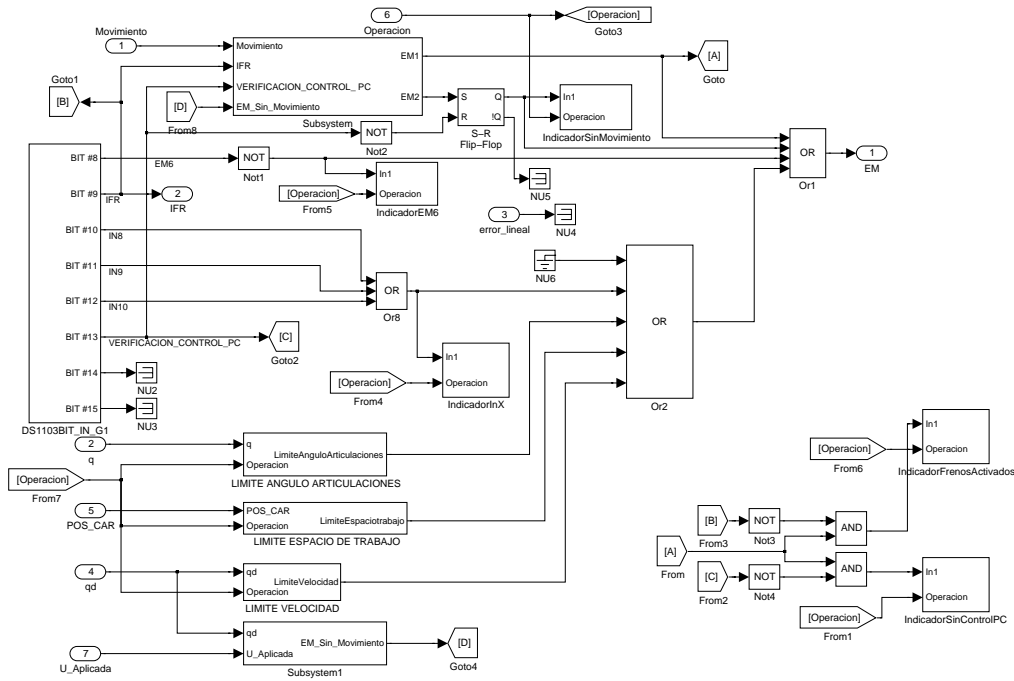


Figura 6: Construcción del Bloque de SIMULINK para supervisión de alarmas.

1.

- Límite de espacio de trabajo: esta emergencia se activará si la garra del robot se sale del espacio de trabajo definido.
- Límite de velocidad: fija la velocidad máxima de las articulaciones.
- Sin Movimiento: Alarma que se activa en caso de no detectarse movimiento en algún eje cuando se le aplica un par suficientemente grande. Esta situación, puede deberse a una incorrecta lectura por parte del encoder, o a una inadecuada desactivación de los frenos.
- Sin Control PC: aparecerá esta alarma si se han quitado los frenos del robot y la tarjeta de conmutación no ha activado el control desde PC.
- Frenos Activados: produce una emergencia en caso de intentar controlar desde PC con los frenos activados.
- Error lineal: activa una alarma si el error cometido entre la posición real del robot y la posición de referencia es excesivo. Indica que el controlador experimental no es capaz de llevar al robot manipulador por la trayectoria de referencia.

La Máquina de Estados. Este bloque se encarga de efectuar las transiciones entre los distintos estados del sistema y habilita al bloque de

Eje del Robot	Ángulo mínimo	Ángulo máximo
1	-260	260
2	-100	0
3	-50	50
4	-140	140
5	-80	80
6	-140	140

Tabla 1: Límites de los ángulos de las articulaciones del robot *RM-10* en grados.

control en caso de que el supervisor de alarmas no detecte un comportamiento anormal. Los distintos estados de funcionamiento están descritos a continuación:

ENCENDIDO. En el estado de reposo del sistema, el control está habilitado y los frenos están activos.

OPERACIÓN. El robot está habilitado y en espera de recibir señales de control en los reguladores de velocidad. los frenos siguen activos.

MOVIMIENTO MANUAL. Permite el control en velocidad de los motores del *RM-10* por separado mediante unos pulsadores localizados en el panel del programa de control supervisor que se verá en la sección 5. No tiene aplicación desde el punto de vista de implementación de controladores, pero es impor-

tante para situar el robot en posiciones iniciales

MOVIMIENTO TRAYECTORIA. Es el estado que hay que habilitar para utilizar los controladores desarrollados. El controlador que se está probando sigue la referencia generada en el generador de trayectorias a partir de la posición inicial de los ejes. Es posible permitir que deje de controlar al finalizar la trayectoria o bien que continúe manteniendo el robot estático en la posición final.

MOVIMIENTO FRENADO. durante la permanencia en este estado se pretende conseguir un correcto frenado del RM-10 tanto al finalizar un movimiento manual correcto como al producirse alguna alarma durante un movimiento (manual o automático).

ESPERA FRENOS. Se ha introducido este estado intermedio para posibilitar una situación de espera que asegure la entrada efectiva de los frenos del robot antes de finalizar el control.

5 CONTROL SUPERVISOR Y ADQUISICIÓN DE DATOS EN EL ORDENADOR

Para llevar a cabo las tareas de supervisión durante los ensayos con el robot manipulador, se ha hecho uso de los programas Cockpit y Trace que proporciona *dSPACE*.

Cockpit es un programa SCADA que permite, mediante un editor gráfico, diseñar pantallas de explotación de variables del modelo así como su modificación en tiempo real. En este proyecto se ha desarrollado un panel de control que permite:

- visualizar medidas, como por ejemplo: lecturas de encoders, medidas de posición, error en posición y señal de control. También presenta la máquina de estados e indicadores de alarma de posibles fallos detectados.
- Modificar variables del modelo de Simulink. A través de elementos de edición se pueden variar parámetros del controlador. Mediante pulsadores se controla en el panel el posicionamiento manual de cada eje en sentido positivo o negativo, la activación del robot pasando a estado habilitado y el comienzo del seguimiento de la trayectoria con el controlador experimental al activar el modo automático.

En la figura 7 aparece la composición de la pantalla del panel de control.

En la aplicación Trace de *dSPACE* se representa la evolución de cualquier variable definida previamente en el modelo de SIMULINK. Esta herramienta permite representar gráficamente valores relevantes del ensayo y comprobar el funcionamiento del controlador.

En la figura 8 puede observarse la pantalla donde se representan los valores de posición angular, velocidad angular y señal de control para los ejes 4 a 6. En concreto, en dicha figura, aparecen señaladas la evolución de la velocidad angular y la señal de control aplicada del eje 6. Con la pantalla de selección se puede elegir, para representar gráficamente, cualquier variable del modelo de SIMULINK.

6 DESARROLLOS FUTUROS

La biblioteca Clib de la tarjeta *dSPACE* abre nuevas posibilidades de ampliación y mejora del equipo. En esta biblioteca están incluidas funciones en C que permiten la lectura y escritura (en tiempo de ejecución) de variables de la tarjeta de control *dSPACE* desde el ordenador. En esta línea de trabajo, se está empezando a desarrollar un sistema de seguimiento visual para el brazo manipulador mediante la incorporación de un sistema de visión. Asimismo, se pretende hacer uso de esta plataforma para la implementación de diversos controladores en vías de investigación en los campos del control robusto, no lineal, etc. que incluyen el estudio de modelos de fricción.

Agradecimientos

Este trabajo ha sido financiado por el proyecto CICYT TAP98-0541 y por el proyecto FEDER 1FD97-C

Referencias

- [1] John J. Craig, *Introduction to robotics: Mechanics and control*, Addison Wesley, 1989.
- [2] *dSPACE, Matlab-dspace interface libraries*, Manual, 1999.
- [3] K.S. Fu, R.C. Gonzalez, and C.S.G. Lee, *Robótica, control, detección, visión e inteligencia*, McGraw Hill, 1989.
- [4] C. Pérez, *Simulación y control de un robot manipulador*, Escuela Superior de Ingenieros, Universidad de Sevilla, 1999.
- [5] System Robot, *Manuale d'uso rm10*, 1991.

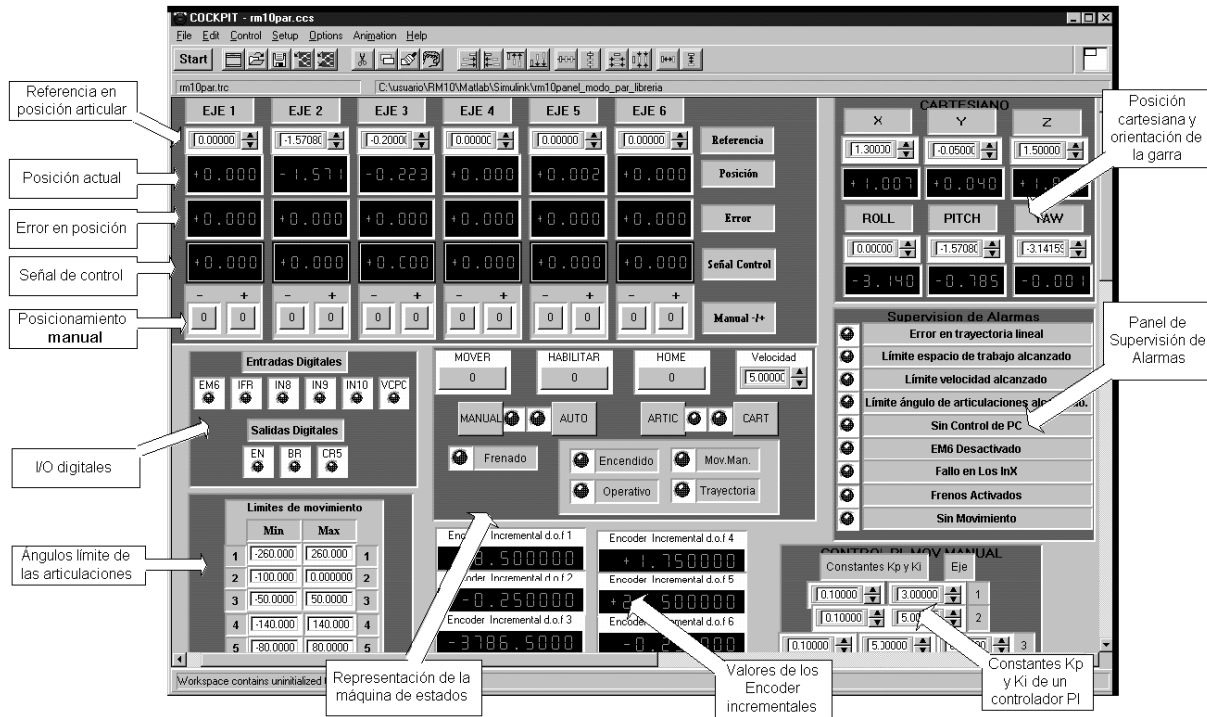


Figura 7: Programa SCADA para la supervisión de ensayos con el Robot Manipulador de experimentación.

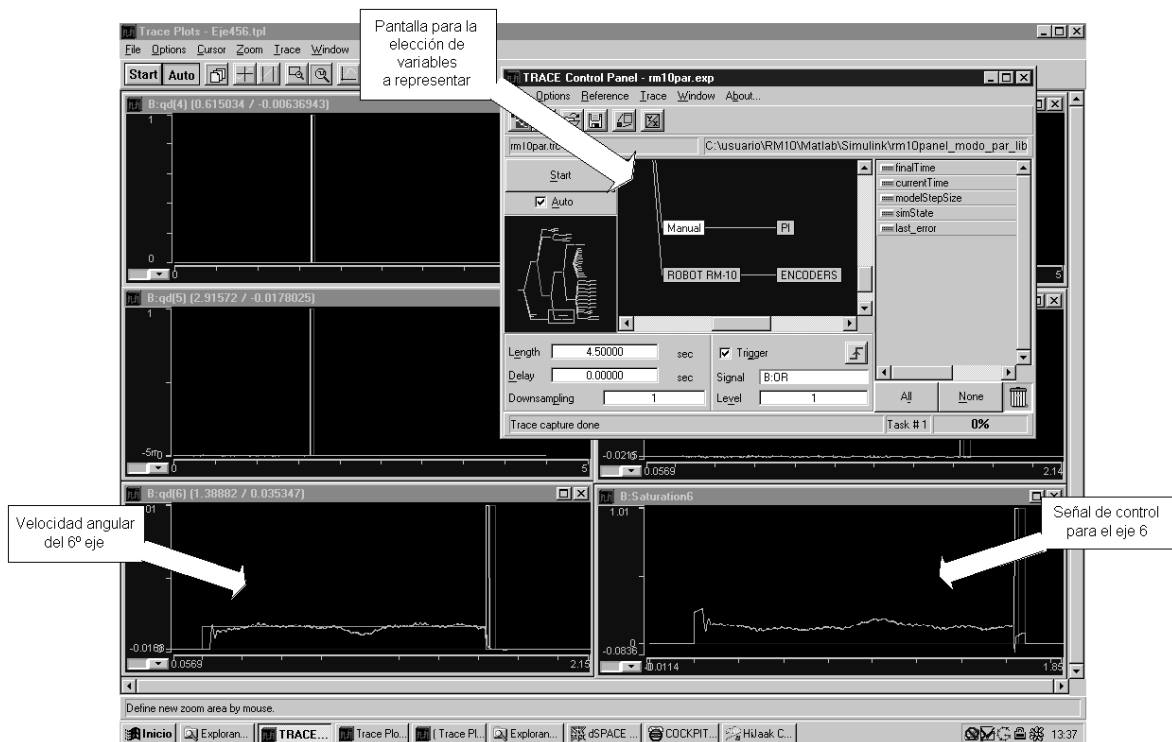


Figura 8: Aplicación para visualizar la evolución de variables definidas en el modelo de SIMULINK.