

Proyecto de Fin de Carrera
Integración de un sensor de Fuerza y Torsión al
efector final de un robot manipulador

Autor: Álvaro Payán Rodríguez

Septiembre 2005

Índice

1	Introducción.	4
1.1	Contenidos y objetivos del proyecto	4
1.1.1	Resumen de objetivos	4
1.1.2	Contenidos y organización de la documentación.	5
1.2	Introducción al control por realimentación de esfuerzos	6
1.3	Equipo utilizado en el proyecto	7
1.3.1	Aplicaciones Software	7
1.3.2	Material Hardware	8
2	Sensor de Fuerza/Torsión.	9
2.1	Introducción	9
2.2	Descripción de los componentes del sensor	10
2.2.1	Transductor	10
2.2.2	Cable del transductor	12
2.2.3	Controlador	12
2.3	Software	15
2.3.1	Lista de comandos	15
2.4	Características necesarias para una comunicación serie	19
2.5	Estudio de velocidades de comunicación	19
2.5.1	Velocidad de comunicación puerto serie	21

<i>ÍNDICE</i>	2
2.6 Datos necesarios para una comunicación directa con el transductor . . .	23
3 Protocolo de comunicación mediante el puerto serie.	28
3.1 Introducción	28
3.2 Conexionado	28
3.3 Algoritmo de comunicación	30
3.3.1 Configuración	32
3.3.2 Intercambio de datos	34
4 Comunicación analógica.	38
4.1 Introducción	38
4.2 Uso de la tarjeta dSPACE para implementar el protocolo de comunicación	38
4.3 Descripción de los esquemas de Simulink	39
4.4 Integración en el modelo del RM-10	42
5 Placa de protección y conexionado.	44
5.1 Introducción	44
5.2 Diseño de la placa de protección	44
5.2.1 Protecciones analógicas	45
5.2.2 Protecciones digitales	50
5.2.3 Diseño final de la placa	53
5.3 Conexionado	54
5.3.1 Conexión fuente de alimentación del controlador-tarjeta de protecciones	54
5.3.2 Conexión transductor-tarjeta de protecciones	58
5.3.3 Conexión fuente de alimentación del PC-tarjeta de protecciones	58
5.3.4 Conexión tarjeta dSPACE-Tarjeta de protecciones	59

<i>ÍNDICE</i>	3
6 Pantallas creadas para ControlDesk.	60
6.1 Introducción	60
6.2 Pantalla de pruebas para la comunicación por el puerto serie	61
6.3 Pantalla de pruebas para la comunicación analógica	62
6.4 Integración en la pantalla general de control del RM-10	63
7 Acoplamiento al efector final	66
7.1 Introducción	66
7.2 Descripción de la fresadora EMCO PC MILL 125	67
7.2.1 Carros de desplazamiento	70
7.2.2 Tambor Portaherramientas	70
7.2.3 Zona de programación y control	71
7.3 Diseño del acoplamiento para el sensor de fuerza al robot	73
7.3.1 Piezas de unión entre el sensor y el manipulador	76
7.3.2 Pieza adicional para poder añadir una varilla de contacto al transductor	80
8 Pruebas de funcionamiento.	84
8.1 Introducción	84
8.2 Prueba de Bias y de medición de F_z	85
8.3 Comprobación de escala en F_z	88
9 Conclusiones y desarrollos futuros.	92

Capítulo 1

Introducción.

1.1 Contenidos y objetivos del proyecto

1.1.1 Resumen de objetivos

El objetivo del proyecto es añadir un sensor de Fuerza y Torsión al efector final de un robot manipulador. En este caso se ha trabajado con la plataforma del robot RM-10 y todo el trabajo se ha desarrollado aplicado a su entorno de programación y de control.

Haciendo uso de la tarjeta de control en tiempo real modelo DS1103 de la compañía dSPACE, conseguiremos mantener una comunicación directa con el propio sensor que nos permitirá obtener datos de Fuerza y Torsión y usarlos para llevar a cabo un control con realimentación de esfuerzos.

En este proyecto se describen varios protocolos de comunicación con el sensor para el intercambio de datos, cada uno de ellos con unas características distintas. Además, se implementarán dichos protocolos en Matlab (Simulink) con el objetivo de añadir los datos de Fuerza y Torsión a todos los esquemas ya desarrollados para el control del manipulador RM-10. Por otra parte también se describen los esquemas de la tarjeta de protección creada para la conexión dSPACE-SENSOR y las modificaciones llevadas a cabo en el efector final del manipulador para el acoplamiento físico del transductor.

Como bibliografía fundamental usada para la realización del proyecto, conviene destacar el Proyecto de Fin De Carrera de Rafael Romero Vicente [1], que si bien no incluye nada acerca de estrategias de control con realimentación de esfuerzos, ha servido como guía para conocer el entorno de control del RM-10, y ha permitido integrar todo

el protocolo para la medida de Fuerzas y Torsiones en dicho entorno.

1.1.2 Contenidos y organización de la documentación.

La documentación del proyecto se desglosa en los siguientes ítems:

- **Memoria descriptiva.** La presente memoria describe todos los planteamientos teóricos adoptados y los resultados prácticos obtenidos durante la realización del proyecto
- **Apéndice A: Códigos fuente.** Consiste en el listado de los códigos fuente de todas las *s-functions* creadas
- **Apéndice B.** Contiene los siguientes apartados:
 1. Diagramas de bloques creados en Simulink.
 2. Esquemas eléctricos de la tarjeta de protección.
- **CD-ROM.** En este soporte se incluyen:
 1. Memoria del proyecto y Apéndices.
 2. Programas software desarrollados en Matlab y ControlDesk
 3. Imágenes, fotos, vídeos.
 4. Otra documentación.

Concretamente, la memoria actual se encuentra estructurada de forma que el capítulo 1 introduce al lector en el contenido del proyecto y en el motivo por el cual se hace necesario llevar a cabo una medición de Fuerzas y Torsiones. En el capítulo 2 se lleva a cabo una descripción del sensor utilizado para el proyecto y de todas sus características. Por su parte, los capítulos 3 y 4 describen los distintos algoritmos implementados para llevar a cabo la comunicación con el sensor y el capítulo 5 describe la tarjeta de protección creada para la conexión entre el sensor y la tarjeta de comunicación. El capítulo 6 describe las pantallas de explotación creadas para controlDesk. El capítulo 7 comenta la forma en la que se ha acoplado físicamente el transductor al efector final del manipulador. Por último el capítulo 8 muestra los resultados experimentales obtenidos en algunas pruebas realizadas. El capítulo 9 comprende tanto las conclusiones finales como las futuras líneas de trabajo que se proponen.

1.2 Introducción al control por realimentación de esfuerzos

Aunque el objeto de este proyecto no sea el control por realimentación de esfuerzos en sí, es importante destacar que todo el trabajo desarrollado se ha hecho con el fin de disponer de medidas de fuerza y torsión para poder implementar en un futuro controles de este estilo. Sin entrar en aspectos teóricos e implementaciones de controladores, conviene al menos comentar brevemente el motivo por el que existen aplicaciones en las que resulta necesario el conocimiento de las fuerzas existentes en el sistema para poder llevar a cabo un control óptimo sobre dicho sistema. En cualquier caso, para mayor información sobre técnicas de control mediante realimentación de esfuerzos, se remite a [2], [3] y [4].

Muchas tareas de automatización requieren la interacción entre el manipulador y su entorno. Por ejemplo tareas como excavaciones, demoliciones, empujar algún objeto, rasparlo, afilarlo, pulirlo, retorcerlo, etc. Todas estas capacidades requieren intrínsecamente que el manipulador presente un controlador de fuerza.

Hoy en día, el control de fuerza comprende un campo de investigación muy potente debido a su importancia para muchas tareas y a la sobreexplotación que han sufrido otras estrategias de control cuyos campos de mejoras están prácticamente agotados. Hasta la fecha dos acercamientos principales han sido propuestos para el control de fuerza: *control explícito de fuerza* y *control de impedancia*.

Un control de impedancia con realimentación de fuerza contiene un controlador explícito de fuerza pero sin necesidad de tomar medidas de dichas fuerzas. No es necesario realimentar las fuerzas existentes en el sistema porque se hace uso del concepto de impedancia. Este concepto permite caracterizar de alguna forma al entorno con el que se interacciona y estimar, ajustando algunos parámetros, las fuerzas aplicadas.

El control explícito de fuerza involucra la medida directa de los valores de fuerza, buscando siempre el objetivo de situar los lazos de realimentación lo más cerca posible de las salidas. Dentro del control explícito de fuerza se pueden distinguir además dos tipos distintos: *basado en fuerza* y *control de fuerza con bucle interior en posición*. La primera de ellas es la más común y suele emplear algún tipo de control PID ó alguna serie de filtros. Las técnicas de control con bucle interior en posición, tal y como su nombre sugiere, contienen un bucle exterior con realimentación de fuerzas que proporciona referencias en posición para el controlador del bucle interior basado en posiciones.

1.3 Equipo utilizado en el proyecto

Para la realización del proyecto, la pieza fundamental de la que se ha hecho uso ha sido el **sensor de Fuerza/Torsión** de la compañía ATI. Las características técnicas de dicho sensor se describirán en el capítulo 2. Por otra parte también ha sido muy importante la utilización del robot industrial RM10, de la firma italiana System Robot, y todo su entorno que comprende un ordenador personal al que conectamos una tarjeta DSP controladora y una gran variedad de armarios, instrumentación y herramientas disponibles en los laboratorios del departamento de Ingeniería de Sistemas y Automática de la Escuela Superior de Ingenieros de Sevilla.

Adicionalmente se ha hecho uso de un conjunto de aplicaciones software que han permitido el diseño de tarjetas electrónicas, programación, monitorización de ensayos, etc.

1.3.1 Aplicaciones Software

Las principales aplicaciones utilizadas para el desarrollo del proyecto han sido:

1. Microsoft Visual C++ v6.0 para el desarrollo de las s-functions utilizadas en los correspondientes bloques de Simulink.
2. Orcad Release 9. Utilizado para el diseño y elaboración de la placa electrónica de protección creada para las conexiones con la tarjeta dSPACE.
3. Matlab v6.0 que incorpora la herramienta de simulación Simulink v3.0 y las *toolboxes* de control. Adicionalmente se hace uso de los complementos Real Time 22 1 Introducción Workshop y Real Time Interface Library para la compilación de modelos y la descarga en la tarjeta dSPACE. Como guía fundamental de apoyo se ha hecho uso de [5].
4. Software dSPACE con su aplicación Controldesk. Dicha aplicación integra otras dos aplicaciones que se comentan a continuación:
 - Cockpit. Permite establecer pantallas de monitorización que posibilitan la modificación en tiempo real de parámetros de la aplicación cargada en la tarjeta.
 - Trace. Representa la evolución de cualquier variable definida previamente en el controlador en ejecución.

Los manuales usados para todas las dudas planteadas acerca de dicha herramienta Software son [6], [7], [8], [9], [10] y [11].

1.3.2 Material Hardware

Como se ha comentado anteriormente, además del sensor de Fuerza y Torsión se ha hecho uso de una serie de elementos hardware pertenecientes a la plataforma creada para el robot RM10 y que conviene comentar. Todos estos elementos se adquirieron en su día y se configuraron de forma adecuada para hacer funcionar el robot y su entorno. Para poder integrar la medida de fuerzas y torsiones no ha sido necesario modificarlos sino simplemente se ha hecho uso de ellos añadiéndoles además las aplicaciones desarrolladas en el presente proyecto.

1. Sensor de fuerza y torsión de ATI que se describe en el capítulo siguiente.
2. Robot Manipulador RM10.
3. Ordenador PC compatible con microprocesador Pentium II a 350 MHz, 256 Mb de memoria RAM y 20 Gb de disco duro que incorpora una tarjeta de red mediante la cual puede comunicarse con otros ordenadores.
4. Tarjeta procesadora de señales analógicas y digitales DS1103 de la compañía dSPACE que fue instalada en el ordenador anterior y dedicada al control del robot manipulador. Algunas de sus características principales son:
 - Procesador IBM PowerPC a 333 MHz.
 - DSP esclavo con funciones de entrada y salida.
 - Memoria SRAM local de 2 Mb y DRAM global de 32 Mb.
 - Entradas y salidas digitales: se han usado tres canales de salida digital.
 - Entradas analógicas (rango $\pm 10V$): 2 unidades ADC 16-bit, multiplexadas.
 - Salidas analógicas (rango $\pm 10V$): se ha hecho uso de dos canales de salida analógica de 14-bit.
5. Herramientas y dispositivos de instrumentación facilitados en los laboratorios del departamento de Automática y Robótica.
6. Otros elementos adicionales, tales como cableado diverso (clasificado por colores), conectores de distintos tipos, componentes electrónicos de aislamiento galvánico, etc.

Capítulo 2

Sensor de Fuerza/Torsión.

2.1 Introducción

El presente proyecto se ha realizado utilizando un sensor de fuerza/torsión de ATI. En concreto, se ha hecho uso de un sensor modelo *gamma*. Como se ha comentado en el capítulo anterior, dicho sensor ha sido el elemento fundamental para la realización del presente proyecto, y por tanto, el objetivo de este capítulo es realizar una descripción de todas las características necesarias para trabajar con dicho sensor. En estas características se incluyen aquellas relacionadas con el propio hardware del sistema y aquellas relacionadas con la comunicación.

El problema de la comunicación es de vital importancia en nuestro caso. La propia compañía que fabrica el sensor proporciona un software de usuario con una interfaz muy simple y fácil de usar. El problema de este software es que únicamente permite visualizar las medidas de fuerza y torsión mediante una serie de gráficos dinámicos basados en barras. Para poder hacer uso de este software únicamente se tiene que conectar el controlador de nuestro sensor al puerto serie de algún PC donde dicho software esté instalado.

El problema surge cuando se pretende realizar un control en tiempo real, ya que es necesario disponer de las medidas de fuerza y torsión mediante alguna estructura de datos. Para obtener estos datos es necesario implementar un software adicional que permita una comunicación directa con el sensor mediante la cual se le podrán solicitar los datos deseados. Existirán dos posibles formas de comunicación con éste:

1. Haciendo uso del controlador del propio sensor. Podremos conectarnos a dicho controlador mediante el puerto serie de un PC y crear algún software que nos permita solicitar medidas al propio controlador e interpretar los datos recibidos

por el puerto serie.

2. Obviando el uso del controlador y llevando a cabo una comunicación directa con el transductor. Para implementar este método habrá que conocer el significado de todas las señales de comunicación del propio sensor y habrá que trabajar a nivel físico con las líneas de información. Es decir, haciendo uso de los canales de entrada y salida de la tarjeta dSPACE deberemos trabajar directamente con señales de tensión.

Los manuales fundamentales usados como bibliografía para entender el funcionamiento del sensor han sido [12] y [13].

2.2 Descripción de los componentes del sensor

El sistema completo asociado al sensor está formado por los siguientes componentes:

- Transductor
- Cable del transductor
- Controlador

A continuación pasamos a describir cada uno de estos elementos en detalle.

2.2.1 Transductor

El transductor es una estructura compacta que convierte fuerza y torsión en una señal analógica de la cual hace uso el controlador (ver figura 2.1). Viene provisto de unos pines de sobrecarga que lo protegen contra cualquier daño provocado por sobrecargas inesperadas. Usualmente el transductor se acopla en la muñeca de un robot manipulador entre dicho robot y su efector final.

El modelo *gamma* del cual disponemos está compuesto de galgas de silicio que proporcionan una medida 75 veces superior a las de las galgas convencionales de aluminio. Gracias a ello, se requiere una amplificación mucho menor que permite obtener una señal con una distorsión debida al ruido muy cercana a cero. En concreto se usan

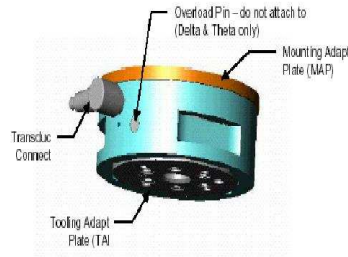


Figura 2.1: Transductor

6 galgas cada una con una resistencia nominal entre 650Ω y 950Ω . Es necesario que las señales de excitación (SG_{HI} , SG_{LO}) sean capaces de proporcionar una carga de 200Ω . En la figura 2.2 se muestra el esquema eléctrico del transductor.

Cada galga mide una porción de la carga total del transductor. A su vez, como podemos observar las galgas se dividen en dos partes formando un divisor de tensión que permite obtener la señal que representa la carga aplicada. Para que este proceso funcione correctamente es necesario que las señales de excitación empleadas sean muy estables y con un nivel muy bajo de ruido, ya que cualquier desviación en la señal de alimentación nos llevaría a cambios bastante importantes en las señales de salida del sensor.

El transductor dispone además de un circuito termistor que permite compensar temperaturas en casos en los que las condiciones de trabajo sean extremas. Sin embargo, no entramos en este documento en detalle acerca de la compensación de temperatura debido a que en nuestras condiciones de trabajo (con temperaturas normales), no resulta necesaria dicha compensación. En cualquier caso, se recomienda acudir a los manuales del sensor para obtener más información sobre este aspecto.

Las tablas de calibración del transductor para el sistema inglés y el sistema métrico y las propiedades físicas del transductor se muestran en la figura 2.3.

Por último, es importante destacar que las galgas están situadas adecuadamente para poder compartir la información de fuerza y torsión aplicadas al sensor. Por este motivo, es relativamente fácil saturar el transductor si nos salimos de un cierto rango de operación. En la figura 2.4 se muestran gráficas que nos permiten obtener el rango de nuestro sensor en el sistema Inglés y en el sistema Internacional. En las gráficas superiores se muestran combinaciones de fuerzas en el eje x y/o y con torsiones sobre el eje z. En las gráficas inferiores se representan combinaciones de fuerzas en el eje z con torsiones alrededor de los ejes x y/o y.

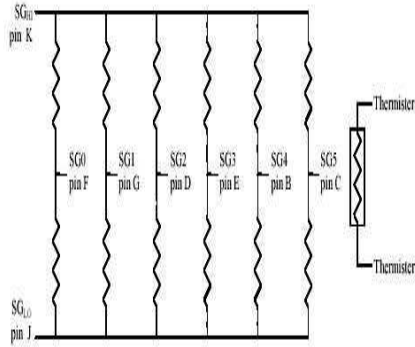


Figura 2.2: Esquema Transductor

2.2.2 Cable del transductor

Para la conexión entre el transductor y el controlador se hace uso de un cable flexible eléctricamente apantallado para protegerlo de transmisiones e interferencias externas. En el extremo que se conecta al controlador posee un conector de 15 pines estándar de dos filas, mientras que en el otro extremo (el que se conecta al transductor) dispone de un conector especialmente protegido de doce pines (véase figura 2.5).

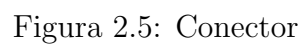
2.2.3 Controlador

La función principal del controlador es convertir la señal eléctrica que procede de las galgas del transductor en una señal de fuerza y torsión en componentes cartesianas. Para el controlador usado para el presente proyecto, la comunicación puede realizarse mediante el puerto serie ó mediante la E/S discreta. En el caso de que queramos prescindir del controlador en el protocolo de comunicación deberemos conocer cuales son sus funciones y emularlas mediante algún software creado y mediante las entradas y salidas analógicas y digitales de nuestra tarjeta de comunicación.

Calibration	English (US)			Metric (SI)		
	US-7.5-25	US-15-50	US-30-100	SI-32-2.5	SI-65-5	SI-130-10
Rated Sensing Ranges						
Fx, Fy	±7.5 lb	±15 lb	±30 lb	±32 N	±65 N	±130 N
Fz	±25 lb	±50 lb	±100 lb	±100 N	±200 N	±400 N
Tx, Ty, Tz	±25 in-lb	±50 in-lb	±100 in-lb	±2.5 Nm	±5 Nm	±10 Nm
Resolution (16-bit)						
Fx, Fy	1/2560 lb	1/1280 lb	1/640 lb	1/640 N	1/320 N	1/160 N
Fz	1/1280 lb	1/640 lb	1/320 lb	1/320 N	1/160 N	1/80 N
Tx, Ty	1/1280 in-lb	1/640 in-lb	1/320 in-lb	1/8000 Nm	3/16000 Nm	1/3200 Nm
Tz	1/1280 in-lb	1/640 in-lb	1/320 in-lb	1/8000 Nm	3/16000 Nm	1/3200 Nm
Counts Value (16-bit)						
Fx, Fy, Fz	2560 / lb	1280 / lb	640 / lb	640 / N	320 / N	160 / N
Tx, Ty, Tz	2560 / in-lb	1280 / in-lb	640 / in-lb	8000 / Nm	5333.28 / Nm	3200 / Nm
Tool Transform Factor	0.01 in/unit	0.01 in/unit	0.01 in/unit	0.8 mm/unit	0.6 mm/unit	0.5 mm/unit

Physical Properties	US	Metric
Stiffness (Calculated)		
X-axis and Y-axis force (Kx, Ky)	52×10 ³ lb/in	9.1×10 ⁶ N/m
Z-axis force (Kz)	100×10 ³ lb/in	18×10 ⁶ N/m
X-axis and Y-axis torque (Ktx, Kty)	93×10 ³ in-lb/rad	11×10 ³ Nm/rad
Z-axis torque (Ktz)	140×10 ³ in-lb/rad	16×10 ³ Nm/rad
Resonant Frequency (Measured)		
Fx, Fy, Tz	1.4 kHz	
Fz, Tx, Ty	2.0 kHz	
Maximum Single-axis Load		
Fx, Fy	±270 lb	±1200 N
Fz	±910 lb	±4100 N
Tx, Ty	±690 in-lb	±79 Nm
Tz	±730 in-lb	±82 Nm
Weight		
Transducer with standard aluminum MAP plates	0.56 lb	250 g
Transducer with ring/plug TAP	0.75 lb	325 g
Transducer with ring/plug MAP	0.8 lb	375 g
Material		
Transducer	Aircraft Aluminum	
Standard mounting adapter	Aircraft Aluminum	
Ring/Plug adapters	Aircraft Aluminum	

Figura 2.3: Calibración y propiedades físicas



2.3 Software

El uso del controlador nos permite obtener una interfaz bien definida de comunicación con el sensor. Para ello, disponemos en el manual del sensor de una lista con todos los comandos que podemos enviar al controlador y las respuestas correspondientes de éste. La comunicación con el controlador se realiza con una conexión estándar RS-232 (puerto serie), ya que en nuestro caso no disponemos de otros puertos opcionales.

El procedimiento de comunicación consiste por tanto en llevar a cabo una conexión con el controlador mediante su puerto serie y hacer uso de una serie de comandos definidos. La mayoría de estos comandos consisten en 3 caracteres ASCII. Cuando el controlador recibe un comando responderá con el comando de control $\langle ACK \rangle$ (decimal 6) si el comando es válido ó un $\langle NAK \rangle$ (decimal 21) si el comando es incorrecto. En caso de que se envíe un $\langle NAK \rangle$, se mandará a continuación un mensaje de error seguido de $\langle CR \rangle \langle LF \rangle$ y el carácter ">". Por el contrario, si un comando válido es recibido, el controlador responderá con los datos que correspondan seguidos de $\langle ACK \rangle \langle CR \rangle \langle LF \rangle$ ">" que indicarán que la respuesta se ha completado.

Antes de pasar a describir los comandos más importantes, conviene comentar que el software del controlador dispone además de una bandera de $\langle error \rangle$. Dicha bandera puede tener los valores 0, 1, 2 ó 3. Normalmente su valor es 0 y pasa a valer 1 si las fuerzas aplicadas al sistema exceden el rango (saturación). El valor 2 indica que ha ocurrido un error en el transductor, como por ejemplo, que se ha roto una galga ó que se ha desconectado el cable. Si $\langle error \rangle$ contiene el valor 3 será debido a que ha ocurrido un error de saturación y de transductor simultáneamente. Cuando ocurre un error con el sensor podremos observar los siguientes pasos.

- Se encenderá el LED indicativo de error en el frontal de la caja del controlador.
- Por el puerto serie se enviará un mensaje de error que continuará hasta que el motivo del error desaparezca ó hasta que se ejecute el comando "SM 0".
- La bandera $\langle error \rangle$ adquirirá el valor 1, 2 ó 3.

2.3.1 Lista de comandos

Comandos de selección de modo de comunicación

El modo de comunicación puede ser binario ó ASCII. El modo ASCII tiene la ventaja de la facilidad a la hora de leer los caracteres pero por otra parte no es capaz de proporcionar una tasa de envío tan rápida debido a que hace uso de un mayor número de bytes. Los siguientes comandos permiten seleccionar el modo de comunicación.

- **CD A:** Comunicación modo ASCII.
- **CD B:** Comunicación modo binario.

Comandos de selección de tipo de dato

El software del sistema nos permite seleccionar el tipo de dato que deseamos obtener. A continuación se muestran los distintos comandos de selección y se describe para cada uno de ellos el tipo de datos que se obtiene.

- **CD D:** "Data Decimal Gage". Los datos proporcionados son directamente los valores de tensión que se tiene en cada galga. Cada uno de ellos se representa mediante un valor decimal dentro del rango -2048 a 2047.

Si nos centramos en el estudio del modo ASCII, podemos describir el formato de la línea de datos que proporciona el controlador cuando dichos datos son solicitados.

ASCII: $\langle error \rangle, XXXXXX, XXXXXX, XXXXXX, XXXXXX,$
 $XXXXXX, XXXXXX \langle CR \rangle \langle LF \rangle$

Como se puede observar, la línea consta de 45 bytes. En primer lugar se envía la bandera $\langle error \rangle$ seguida de una coma y, a continuación, los seis valores de galgas cada uno de ellos representados en 6 Bytes y separados por comas. Finalmente se envía un retorno de carro y un "Linefeed".

- **CD H:** "Data Hexadecimal Gage". El formato es idéntico al anterior con la salvedad de que los datos de las galgas se representan ahora en valores Hexadecimales. Debido a este motivo la línea de datos cambiaría sensiblemente pasando ahora a constar de 37 bytes, ya que cada campo de datos pasaría a tener una longitud de 4 bytes.
- **CD R:** "Data Resolved". El controlador proporciona datos de fuerza y torsión en un número decimal ("counts") que habría que escalar adecuadamente para obtener unidades del Sistema Internacional (ver figura 2.3).

La línea de datos que proporciona el sensor tendría el siguiente formato.

ASCII: $\langle error \rangle, XXXXXX, XXXXXX, XXXXXX, XXXXXX,$
 $XXXXXX, XXXXXX \langle CR \rangle \langle LF \rangle$

Como se puede observar el formato de la línea es igual a la de las anteriores con la salvedad de que ahora los valores de los datos representan los valores de fuerza y torsión (counts) y no los valores de las tensiones en las galgas. El orden de estos datos es: Fx, Fy, Fz, Tx, Ty, Tz.

Comandos de petición de datos

- **QR:** Solicita al controlador una línea de datos. Dicha línea se ajustará al formato descrito en el apartado anterior. Además se añaden otros caracteres en la respuesta.

Usuario: $QR \langle CR \rangle$

Respuesta: $\langle ACK \rangle \langle \text{línea de datos} \rangle \langle CR \rangle \langle LF \rangle \langle ACK \rangle \langle CR \rangle \langle LF \rangle \rangle$

Como se puede observar, si trabajamos en modo "Data Resolved", la respuesta del sensor constará de 54 bytes, ya que la línea de datos contiene 45 bytes.

- **QS:** Solicita un flujo continuo de datos hasta que el usuario ejecute otro comando ó envíe $\langle CR \rangle$.

Usuario: $QS \langle CR \rangle$

Respuesta: $\langle ACK \rangle \langle \text{línea de datos 1} \rangle \langle \text{línea de datos 2} \rangle \dots$

Usuario introduce $\langle CR \rangle$ (en otro caso el flujo de datos continúa)

Respuesta: $\langle ACK \rangle \langle CR \rangle \langle LF \rangle \rangle \langle CR \rangle \langle LF \rangle \rangle$

Comandos de Bias

El controlador F/T tiene la capacidad de almacenar 3 **Bias** distintos. El uso de dicha capacidad es de gran ayuda para eliminar los efectos de la gravedad (peso del transductor) y otras fuerzas que estén actuando sobre el efector final. Cuando existe algún valor **Bias** almacenado el controlador usa este valor como punto de referencia. A las lecturas de fuerza y torsión se les restará este valor antes de ser transmitidas.

- **SB**: Almacena un dato **Bias** que es almacenado en una cola de tipo LIFO. Si se ejecuta el comando cuando la cola está llena, se sustituirá el dato más reciente, permaneciendo intactos los dos anteriores.
- **SU**: Permite borrar el último **Bias** de la cola. En caso de que hubiese otro **Bias** almacenado, éste pasaría a estar nuevamente en uso.
- **SZ**: Elimina todos los **Bias** almacenados en la cola. Este comando es ejecutado cada vez que se enciende el controlador ó se lleva a cabo un reset.

Comentarios adicionales

A pesar de que tan sólo se han comentado algunas instrucciones, cabe destacar que existen muchos otros comandos para comunicarse con el controlador. Estos comandos no se han comentado en detalle porque para las aplicaciones requeridas no son de gran interés, no obstante se va a realizar al menos una pequeña descripción de ellos.

- Datos de comprobación de error (**CD E**, **CD U**): Existen comandos que permiten añadir bytes de información adicionales cuando una trama de información es solicitada. Estos bytes permiten comprobar si ha habido algún error con alguno de los datos.
- Comandos de selección de salida (**CA b**, **CP a**): Permiten seleccionar los puertos de comunicación. En concreto permiten habilitar ó deshabilitar la salida analógica (es opcional y nuestro controlador no dispone de ella) y también seleccionar entre puerto serie ó paralelo (nuestro controlador sólo dispone de puerto serie).
- Otros comandos para gestión de la comunicación (**CF d**, **CL b**, **CV h**): Estos comandos permiten configurar algunos parámetros de la transmisión como habilitar y deshabilitar algunas opciones adicionales (Opciones de Bias u opciones monitor) ó configurar el envío de "linefeed", $\langle LF \rangle$, seguido de cualquier retorno de carro. Además, para aplicaciones en las que no sean necesarios todos los datos de fuerza/torsión, el comando **CV h** nos permite seleccionar cuáles de los seis datos (F_x , F_y , F_z , T_x , T_y , T_z) deseamos obtener. Esto nos permitirá aumentar la tasa de envío en caso de que no vayamos a usar esos seis valores.
- Petición de la matriz de calibración (**QC**): Muestra por pantalla la matriz de calibración almacenada en el controlador y que está actualmente en uso.
- Petición de los valores de picos (**SP**, **SC**, **QP**): Permiten obtener los valores máximos y mínimos de fuerza torsión aplicados dentro del intervalo de tiempo que se desea estudiar.

- Envío de mensaje de error (**SM b**): permite habilitar ó deshabilitar el envío de un mensaje de error en caso de que se produzca un error.
- Comandos para el puerto de entrada/salida discreto: El puerto discreto permite una monitorización del funcionamiento del sensor que se lleva a cabo en paralelo a la comunicación adicional. Esta monitorización permite aplicaciones como comprobar el estado de ciertas líneas u ordenarle al controlador que envíe un cierto mensaje en caso de que se produzca una determinada condición.
- Marco de trabajo (**TF d, TL, TD d**): Estos comandos permiten cambiar el origen de coordenadas que se toma como referencia en el transductor mediante ciertos desplazamientos en el espacio.
- Macro para arrancar (**ZC 0,"s"**): Permite definir una secuencia de instrucciones que se ejecutarán secuencialmente cada vez que el sistema se encienda ó se aplique un reset.

2.4 Características necesarias para una comunicación serie

Para llevar a cabo una comunicación serie con el sensor de fuerza es necesario usar un cable serie estándar de 9 pines y configurar una comunicación en el otro extremo con transmisión de 8 bits sin paridad y un bit de parada. La tasa de baudios debe coincidir con la establecida en el controlador. En nuestro caso usaremos la más rápida disponible, es decir, 38400 baudios.

El puerto serie RS-232 estándar consta de los pines que podemos ver en las figuras 2.6 y 2.7.

En cuanto a la comunicación por el puerto serie no hay mucho más que destacar, ya que es un interfaz normalizado y únicamente debemos preocuparnos de configurar bien la velocidad y el tipo de datos que se intercambian. Por lo demás, únicamente hay que hacer uso de los comandos detallados en el apartado anterior.

2.5 Estudio de velocidades de comunicación

Teniendo en cuenta que el objetivo final será implementar un protocolo de comunicación para disponer de los datos proporcionados por el sensor en tiempo real, y

PIN	SEÑAL
1	N/C
2	Receive
3	Transmit
4	Reserved
5	Ground
6	N/C
7	N/C
8	N/C
9	N/C

Figura 2.6: Asignación de pines I/O RS-232

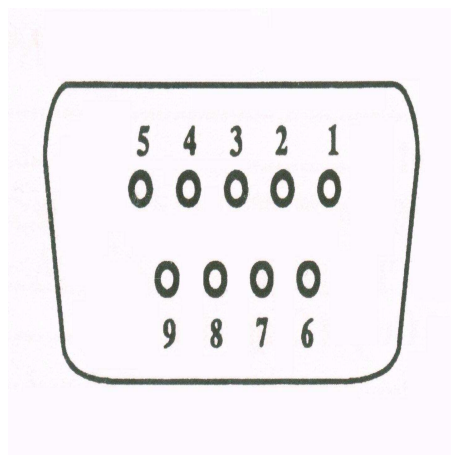


Figura 2.7: Puerto serie disponible en el controlador

de esta forma llevar a cabo algún tipo de control, debemos analizar la rapidez con la que podremos disponer de esos datos y comprobar que se obtiene una tasa de muestreo adecuada para la aplicación requerida en cada momento.

2.5.1 Velocidad de comunicación puerto serie

Existen dos posibilidades para llevar a cabo la comunicación de forma continua con el controlador y poder disponer de los datos proporcionados por éste.

Comando QS

La primera opción, y que a priori parece más adecuada, es la utilización del comando **QS** que proporciona un flujo continuo de datos. Haciendo uso de este comando, es necesario enviar una única instrucción al controlador y a partir de ese instante esperar a que éste envíe los datos de fuerza y torsión a la máxima tasa que sea capaz de proporcionar. Esta opción debemos descartarla por la necesidad de sincronizar la lectura de los datos recibidos. Continuamente estaremos recibiendo datos por el puerto serie e iremos almacenando esos datos en un buffer. El problema surge al extraer los datos de dicho buffer, ya que necesitaremos identificar el comienzo de la línea de datos. Además deberemos asegurarnos de que hemos almacenado suficientes datos para que al menos exista una línea completa. Recordemos que el formato de la línea de datos tenía la forma:

$\langle error \rangle, XXXXXX, XXXXXX, XXXXXX, XXXXXX,$
 $XXXXXX, XXXXXX \langle CR \rangle \langle LF \rangle$

Como se puede observar, el tamaño del buffer de lectura debe ser al menos de 89 bytes para asegurarnos de que al menos disponemos de una línea completa (45 bytes). De esta forma, tendríamos que esperar continuamente a recibir 89 bytes para poder disponer de los datos correspondientes a una medida. Si evaluamos el tiempo empleado para esta operación teniendo en cuenta que trabajamos a una tasa de 38400 baudios (bit/seg):

$$\frac{1}{38400 \text{ seg}} \cdot 8 \frac{\text{bytes}}{\text{bit}} \cdot 89 = 18.54 \text{ mseg}$$

Comando QR

La segunda opción que se propone es la utilización del comando **QR**. Recordemos que este comando proporciona una única línea de datos y que por lo tanto no tendríamos problema alguno con la sincronización. Conociendo el formato de la respuesta del controlador, acto seguido al envío del comando **QR** tomaremos del buffer de lectura los bytes correspondientes a la respuesta y podremos conocer con certeza, dentro de esta línea, la posición de los bytes correspondientes a datos.

Los bytes recibidos por el puerto serie serían por tanto los correspondientes a la respuesta del controlador al comando **QR**, es decir, la secuencia:

$\langle ACK \rangle \langle \text{línea de datos} \rangle \langle CR \rangle \langle LF \rangle \langle ACK \rangle \langle CR \rangle \langle LF \rangle \rangle$

Como se puede observar, la respuesta consta en total de 54 bytes (la línea de datos está formada por 45 bytes). Por lo tanto, el periodo de refresco de los datos sería:

$$\frac{1}{38400} \frac{\text{bit}}{\text{seg}} \cdot 8 \frac{\text{bytes}}{\text{bit}} \cdot 54 = 11.25 \text{mseg}$$

Para aplicaciones como realimentación por control de esfuerzos, este tiempo es demasiado elevado. El tiempo de muestreo es algo crítico en estas situaciones debido sobre todo al instante en que pasa a existir contacto con un cuerpo. En caso de recibir esa información con cierto retardo, el algoritmo de control no funcionará. Por lo tanto existe la necesidad de intentar disminuir el tiempo de refresco de los datos. Debemos plantearnos la posibilidad de que las necesidades de nuestra aplicación requieran una velocidad aún mayor, y por tanto deberíamos buscar algún método para conseguir un periodo de muestreo de al menos 1ms .

Dado que las velocidades de transmisión no se pueden reducir más haciendo uso del puerto serie y dado que nuestro controlador no dispone de puerto analógico ni paralelo (ambos son opcionales), el único medio del que disponemos para reducir los tiempos de transmisión sería obviar por completo el controlador y llevar a cabo una comunicación directa con el transductor. Para ello es necesario realizar un estudio detallado del funcionamiento electrónico de dicho transductor y de la funcionalidad de cada uno de sus pines de comunicación.

2.6 Datos necesarios para una comunicación directa con el transductor

Para poder llevar a cabo una comunicación directa con el transductor obviando la función del controlador debemos en primer lugar conocer la función de cada uno de los pines de comunicación de dicho transductor. Como se ha comentado en apartados anteriores, el transductor dispone de un conector hembra de 15 pines. En la figura 2.8 se muestra la asignación de cada uno de los citados pines.

Como se puede observar los pines 6, 13 y 14 no son utilizados. El resto se comentan a continuación:

- **Alimentación del transductor:** El transductor ha sido diseñado para trabajar con una tensión de $+15V$ y $-15V$, aunque podría funcionar con tensiones tan bajas como $9V$. La fuente de tensión utilizada para la alimentación debe ser de muy bajo ruido y bajo rizado. Para solventar todos los posibles problemas que pueden aparecer a la hora de elegir ó diseñar una fuente de alimentación, podríamos optar por hacer uso de la fuente del controlador del sensor. De esta forma nos aseguramos de que se cumplen todos los requisitos y de que la señal de alimentación es adecuada. Únicamente sería necesario buscar algunos puntos de conexión en el controlador de los cuales sea factible sacar señales de alimentación que irían directamente al transductor.

Como características de la fuente de alimentación del controlador podemos decir que es una fuente lineal de $\pm 15V$ con una precisión de $\pm 0.6V$ y una regulación típica de línea de $13mV$. Su regulación de voltaje de salida es $10\mu V/V_0$ a una frecuencia $10Hz \leq f \leq 100KHz$.

- **Alimentación de las galgas:** Además de la señal de alimentación del transductor, se debe proporcionar otra señal de alimentación que servirá para las galgas. Dicha señal debe ser de $+10.0V$ y también debe poseer características como bajo ruido y bajo rizado. La señal de retorno de tierra debe ser conectada a la señal de tierra de alimentación del transductor.

Una vez más podríamos hacer uso de la fuente de alimentación del controlador para obtener esta señal de alimentación. De esta forma no tendríamos que preocuparnos de cumplir ninguna especificación porque sabemos que la señal tomada es válida.

- **Líneas de multiplexión:** Mediante estas tres líneas digitales de entrada al transductor podemos seleccionar la galga cuya tensión queremos obtener a la salida. El hecho de que las señales de salidas se obtengan mediante multiplexión dificulta ligeramente el proceso para conocer las fuerzas y torsiones, ya que será

D-Sub pin del conector	Símbolo	Señal	Descripción
2	+V	Externa	Alimentación positiva del transductor
9	AGnd	Externa	Alimentación del transductor (tierra)
1	-V	Externa	Alimentación negativa del transductor
4	+SG	Externa	Alimentación positiva de las galgas
11	-SG	Externa	Retorno de tierra de las galgas
8	A0	Externa	Selección MUX línea A0(Bit menos significativo)
15	A1	Externa	Selección MUX línea A1
7	A2	Externa	Selección MUX línea A2(Bit más significativo)
5	+Bias	Externa	Señal analógica de Bias no invertida(-5V...+5V)
12	+Bias	Externa	Señal analógica de Bias invertida(+5V...-5V)
3	+Sig	Transductor	Salida de datos no invertida (-5V...5V)
10	-Sig	Transductor	Salida de datos invertida (+5V...-5V)

Figura 2.8: Asignación de pines transductor

necesario realizar un barrido de todas las galgas (variando las señales de multiplexión), y acto seguido multiplicar entonces por la matriz de calibración que permitirá obtener datos de fuerza/torsión a partir de las tensiones en las galgas.

Si nos referimos a valores de tensión, debemos saber que todo lo que esté por debajo de $0.8V$ será interpretado como un 'cero' lógico y todo lo que esté por encima de $2.4V$ será interpretado como un 'uno' lógico. Los valores de tensión aplicados nunca deberán salirse del rango de las tensiones de alimentación del transductor.

El código Gray usado para la multiplexión se muestra en la figura 2.9.

Galga seleccionada	Código Grey A2 A1 A0	Voltage C B A
G0	000	LLL
G1	001	LLH
G2	011	LHH
G3	010	LHL
G4	110	HHL
G5	111	HHH
Termistor	101	HLH
No usado	100	HLL

Figura 2.9: Código de multiplexión de líneas

La señal analógica de salida cambiará en el instante en el que la señal de selección lo haga. La frecuencia para el completo establecimiento de la señal de salida es $8000Hz$, ó lo que es lo mismo, $0.125mseg$.

- **Señal de Bias:** Para cada galga seleccionada en cada momento, deberemos alimentar por los pines de Bias con una tensión distinta para cada galga. Esto se lleva a cabo para eliminar los posibles valores de offset que puedan aparecer. Para nuestro caso (sensor FT 3559) los valores de tensiones necesarias para cada galga se muestran en la figura 2.10
- **Datos analógicos de salida:** La salida de datos analógicos proporciona las tensiones que caen en las galgas de forma diferencial. Será necesario medir para

PAR DE GALGAS SELECCIONADO	+BIAS VOLTAGE	-BIAS VOLTAGE
G0	1.35 V	-1.35 V
G1	2.84 V	-2.84 V
G2	2.72 V	-2.72 V
G3	3.04 V	-3.04 V
G4	2.61 V	-2.61 V
G5	0.88 V	-0.88 V
Termistor	0.41 V	-0.41 V

Figura 2.10: Tensiones de bías necesarias para cada galga

cada galga seleccionada la tensión que existe entre **+SIG** Y **-SIG**, esta diferencia estará dentro del rango ($-10V$, $10V$)

Una vez conocida la función de cada uno de los pines, se puede establecer un protocolo de comunicación para poder obtener los datos en un PC haciendo uso de una tarjeta de adquisición de datos (en nuestro caso usaremos dSPACE).

El protocolo será muy simple y consistirá simplemente en realizar un barrido de todas las galgas para ir almacenando sus valores de tensión. Una vez terminado un ciclo (se han leído las tensiones en todas las galgas) se *procesan* los datos obtenidos y se pasa al siguiente ciclo. De esta forma se continúa indefinidamente.

El procesamiento de los datos mencionado anteriormente correspondería a las operaciones que es necesario realizar para obtener los valores de Fuerza/torsión en función de los valores de tensiones en galgas leídos. Este proceso se reduce únicamente a una multiplicación matricial de la forma:

$$FT = [C \cdot SG]$$

donde:

- **FT** son los datos de fuerza y torsión calibrados. $FT = [F_x, F_y, F_z, T_x, T_y, T_z]$.
- **SG** es la matriz (vector) que contiene los datos de las tensiones en las galgas.
- **C** es la matriz de calibración. Para nuestro caso (sensor FT3559) la matriz de calibración tiene la siguiente expresión:

$$C = \begin{bmatrix} -0.007753 & 0.004112 & -0.018775 & 0.650968 & -0.028382 & -0.652283 \\ -0.010183 & -0.749223 & -0.004976 & 0.381134 & 0.006373 & 0.374143 \\ -1.183820 & 0.017224 & -1.174941 & 0.012184 & -1.178809 & -0.014301 \\ 0.016062 & -0.352207 & 1.340955 & 0.165291 & -1.334350 & 0.158848 \\ -1.553802 & 0.005106 & 0.804629 & -0.306570 & 0.785211 & 0.326036 \\ 0.014600 & 0.810695 & -0.020290 & 0.799408 & 0.027209 & 0.788916 \end{bmatrix}$$

Será necesario tener en cuenta en todo momento, además de la velocidad máxima a la que trabaja el sensor, la velocidad máxima a la que puede trabajar el equipo de adquisición de datos. Si este equipo es suficientemente potente podremos disponer de medidas de fuerza y torsión en intervalos de muestreo del orden de $1msg$, que es un valor bastante adecuado para llevar a cabo un control mediante realimentación de fuerza.

Capítulo 3

Protocolo de comunicación mediante el puerto serie.

3.1 Introducción

Tal y como se ha expuesto a lo largo de los apartados anteriores, el problema de la comunicación con el sensor no está resuelto con el software que proporciona la propia compañía que lo fabrica. Cuando el objetivo es la implementación de un control con realimentación de fuerzas, debemos crear nuestras propias funciones que nos permitan disponer de las medidas del sensor en alguna estructura de datos que podrá ser leída para llevar a cabo el control con realimentación.

Se han comentado en el capítulo anterior las distintas posibilidades en las que puede pensarse para implementar un protocolo de comunicación y obtener medidas del transductor, pero no se ha entrado en detalle en la implementación de ninguna de ellas. En este capítulo se detallará un protocolo que mediante el puerto serie nos permite comunicarnos con el controlador proporcionado con el sensor. Veremos que la principal ventaja de este método será la simplicidad en el protocolo, ya que este se reducirá al intercambio de una serie de comandos (descritos en el capítulo anterior) entre nuestro puesto de trabajo (PC) y el controlador del sensor, que será el encargado de interpretar las señales eléctricas del transductor.

3.2 Conexionado

En la figura 3.1 se muestra un esquema de conexiones del sistema completo. Como puede apreciarse, el transductor se conecta únicamente al controlador. El controlador por tanto será el único encargado de intercambiar señales eléctricas con el transductor e interpretar esas señales como medidas de Fuerza y Torsión.

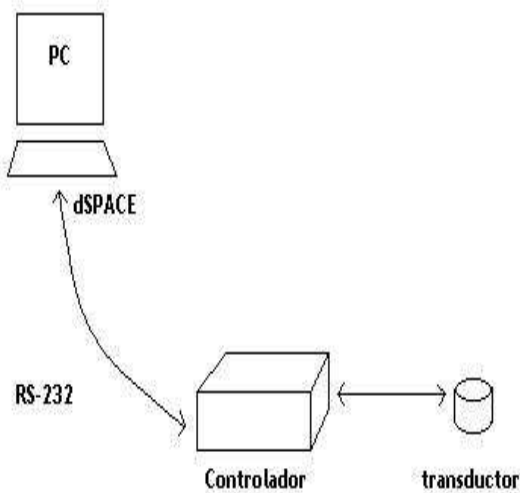


Figura 3.1: Conexionado de los componentes

La comunicación por parte del usuario (PC mediante la tarjeta dSPACE), es una comunicación directa con el controlador mediante el puerto serie. Mediante esta conexión se enviarán una serie de comandos predeterminados, que permitirán obtener un flujo de datos que interpretados de forma adecuada resultarán ser datos de fuerza y torsión.

Para llevar a cabo las conexiones descritas no necesitaremos añadir ningún interfaz a los componentes de nuestro sistema. En primer lugar, la tarjeta de dSPACE dispone de un puerto serie y por tanto únicamente es necesario identificar los pines correspondientes y añadir un conector de salida. Y en segundo lugar, el controlador dispone también de un puerto serie configurable y de un conector DB15 de doble fila que permitirá realizar la conexión con el transductor.

3.3 Algoritmo de comunicación

Cuando hablamos del algoritmo de comunicación, nos referimos al conjunto de órdenes que se intercambiarán de forma periódica para obtener datos de Fuerza/Torsión del sensor. En nuestro caso, se ha creado un algoritmo en el que se pueden diferenciar dos partes, una primera que correspondería a la configuración del sensor y una segunda fase que se encargaría únicamente de pedir datos y acto seguido recibir e interpretar dichos datos.

Si estudiamos las pantallas creadas en Simulink para implementar el protocolo de comunicación, observamos en primer lugar una *macro* con el nombre **sensor** que almacena todos los bloques necesarios para tal efecto. Dicha *macro* dispondrá de tres salidas que corresponderán a los datos leídos del sensor, es decir, bit de error, datos de fuerza y datos de torsión. Esta pantalla la podemos observar en la figura 3.2.

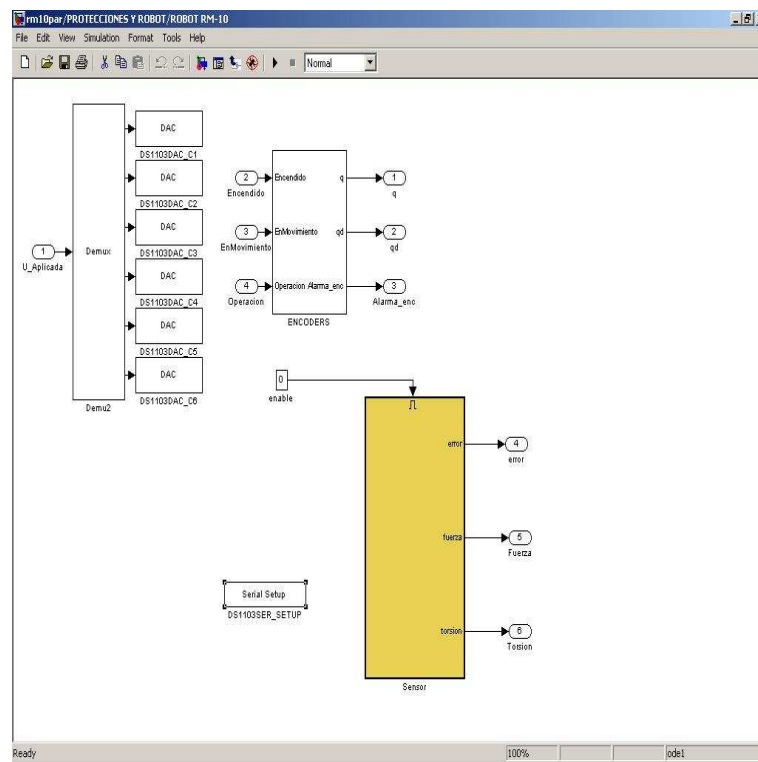


Figura 3.2: Macro principal del protocolo

Como podemos ver en dicha figura, aparece acompañando a nuestra *macro* un bloque de Simulink con el nombre **Serial Setup**. Se puede intuir que este bloque hace las funciones de configuración del puerto serie en el lado de la tarjeta dSPACE que estamos tratando. Esta configuración debe coincidir en ambos extremos de la comunicación (tarjeta dSPACE y controlador del sensor), por tanto los valores que

impondremos son:

- Protocolo RS232.
- Tasa de Baudios de 38400 Baud.
- 8 bits de datos y un bit de parada.
- Sin paridad.
- Cola de recepción FIFO de 256 bytes de tamaño.

Si entramos en la macro para estudiar la implementación del protocolo observaremos una ventana de Simulink como la que se muestra en la figura 3.3. En dicha ventana destacan tres bloques principales:

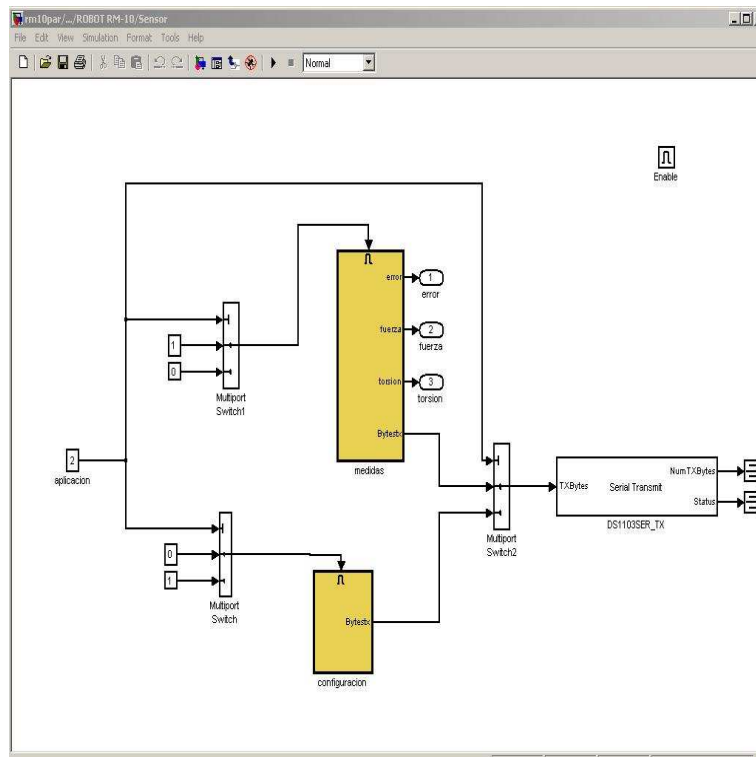


Figura 3.3: Contenido de la Macro principal

- **medidas:** Macro donde se implementa la fase del protocolo que se encarga de pedir los datos de fuerza y torsión al sensor y de interpretar las tramas de respuesta de éste. Como se puede observar, las salidas de dicha macro son las medidas finales de fuerza y torsión.

- **configuracion:** Macro donde se lleva a cabo la configuración de los parámetros de comunicación del sensor.
- **Serial Transmit:** Bloque de simulink que representa el buffer de escritura en el puerto serie de la tarjeta dSPACE. Cualquier byte que se introduzca en su entrada será enviado por el puerto serie.

Como se ha visto anteriormente, el protocolo consta de dos fases diferenciadas. Conviene resaltar que aunque estén implementadas en bloques separados, en ningún momento podrán ser ejecutadas en paralelo. Es decir, si en algún momento se desea reconfigurar algún parámetro como puede ser por ejemplo almacenar un valor de Bias, para poder solicitar esta operación deberemos interrumpir el intercambio de datos de medidas del sensor. Esta exclusión mutua se consigue en Matlab mediante los bloques de habilitación (*enable*). Como se ha visto en la figura 3.3, dispondremos de una variable interna llamada *aplicacion* que permitirá seleccionar una de las dos fases de trabajo.

A continuación se describe la implementación de cada una de las dos fases comentadas.

3.3.1 Configuración

En la fase de configuración se facilita el acceso a algunas funciones adicionales proporcionadas por el controlador del sensor. En concreto, en esta fase podremos hacer uso de las opciones de:

- **Bias:** Podremos almacenar en cualquier instante el valor de las fuerzas y torsiones para usarlos como Bias, es decir, para restárselos a las medidas siguientes. Esto permitirá eliminar los efectos de offset debidos a las fuerzas gravitatorias y otras fuerzas externas que puedan aparecer.
- **Elección del tipo de dato:** Otra de las facilidades que proporcionaba el controlador era la posibilidad de elegir el tipo de dato que queríamos recibir. Haciendo uso de esta opción se ha dado la posibilidad al usuario de obtener los datos en medidas de fuerzas y torsiones ó directamente en valor decimal de la tensión leída en cada galga, es decir, sin haber multiplicado previamente por la matriz de calibración.

Para explicar como se ha llevado a cabo la implementación de las funciones de configuración debemos atender a la figura 3.4 donde se muestra la ventana de Simulink

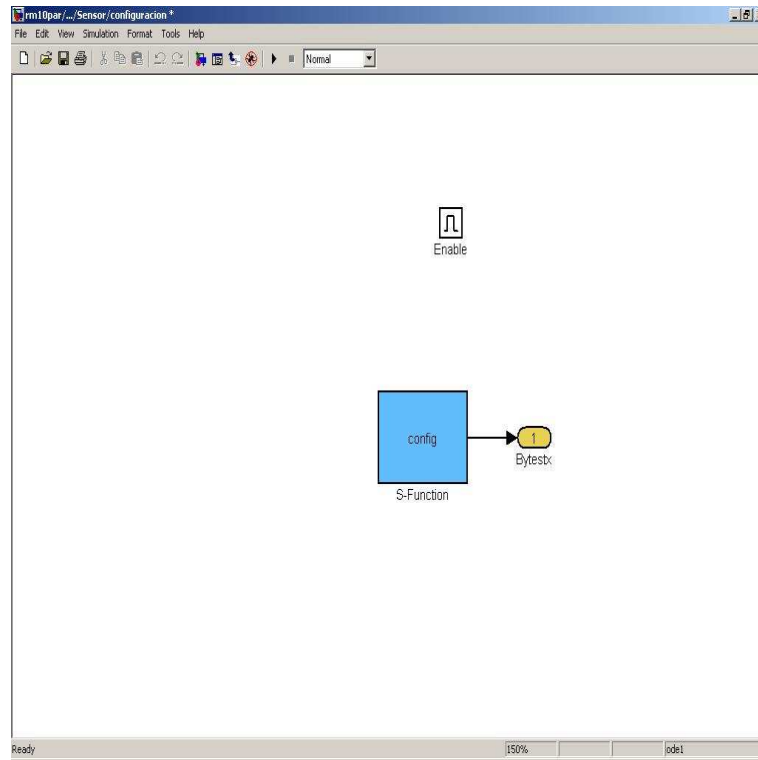


Figura 3.4: Macro de configuración

que encontraremos si entramos en el bloque correspondiente a la macro de Configuración.

Como se puede observar en dicha figura, la parte de configuración consta de un sólo bloque que corresponde a una *s-function* con el nombre **config**. Dicha *s-function* tiene tres parámetros de entrada que podrán ser modificados en cualquier instante desde la pantalla de control creada en *control-desk*. Estos parámetros son:

- **Tiempo de muestreo:** Se podrá configurar el valor del tiempo de muestreo con el que trabajará la función, es decir, podremos seleccionar cada cuanto tiempo deseamos que se ejecute la función.
- **Bias:** Será una variable que podrá tomar los valores 0, 1 ó 2. Si su valor es '1', se almacenará un nuevo valor de Bias que será usado en medidas posteriores. Si vale '2' se eliminará un valor de Bias de la cola LIFO de Bias. Si su valor es '0' no se llevará a cabo ninguna operación de Bias.
- **Tipo de dato:** Variable entera que podrá valer 1 ó 2 (cualquier otro valor será ignorado). En caso de que su valor sea '1' se enviarán las órdenes correspondientes al controlador para que envíe los datos como medidas de fuerzas y torsiones. En caso de valer '2', se le pedirán los datos al controlador como valores decimales de las tensiones medidas en las galgas.

Para llevar a cabo todas estas operaciones, en la *s-function* creada se ha hecho uso de una variable global llamada *sec* que servirá para llevar a cabo una secuencia ordenada de instrucciones. Al final de cada ejecución de la función, la variable *sec* se verá incrementada en una unidad hasta llegar a su valor máximo (3) en el que volverá a establecerse a cero. Por tanto la función constará de 4 fases que se irán ejecutando secuencialmente una detrás de otra cada vez que se produzca una interrupción por el tiempo de muestreo. Al final de cada fase, además de actualizarse el valor de dicha variable, se escribirán en la salida del bloque de la *s-function* tres bytes que serán enviados por el puerto serie al controlador.

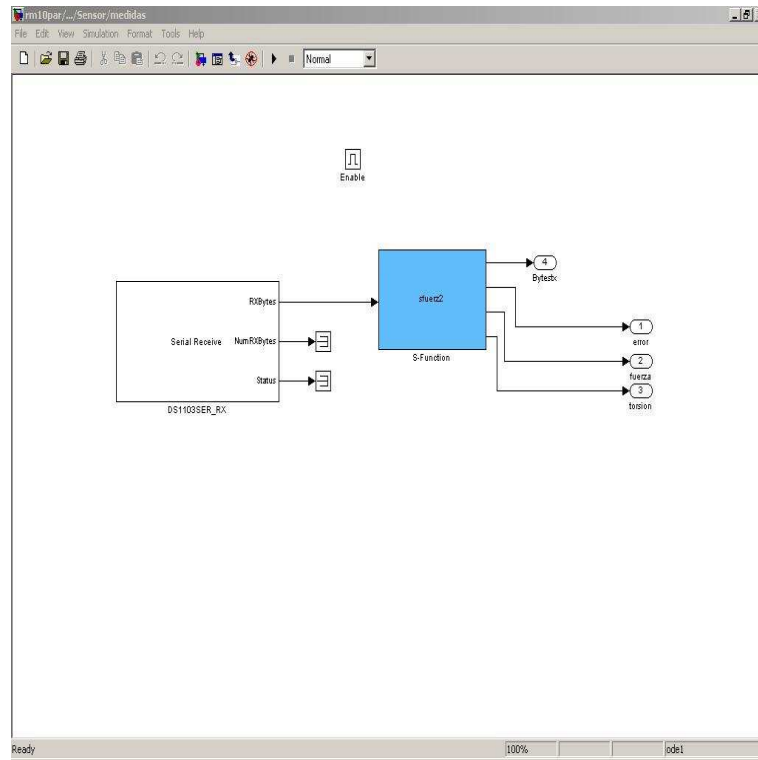
- En la primera fase (*sec*=0) no se llevará a cabo ninguna operación, simplemente se enviarán tres retornos de carro al controlador.
- En la segunda fase se trabajará con la opción de Bias. Se comprobará el valor del parámetro de Bias descrito antes y en función de su valor se enviarán al controlador los bytes necesarios para solicitar la operación deseada.
- La tercera y la cuarta fase servirán para la configuración del tipo de dato. Es necesario hacer uso de dos fases porque al final de cada una de ellas se enviarán por el puerto serie únicamente 3 bytes, y para configurar el tipo de datos hacen falta 4 bytes. Los dos bytes sobrantes serán retornos de carro que se ignorarán en el controlador.

3.3.2 Intercambio de datos

La fase de intercambio de datos es la encargada de mandar órdenes a través del puerto serie para pedir tramas de datos. Además, esperará las respuestas del controlador e interpretará dichas respuestas para facilitar los datos de Fuerza y Torsión. Esta fase será la que permanezca activa durante la mayor parte de la ejecución, ya que una vez configurados los datos inicialmente, lo normal es que se dejen invariables.

Si entramos en la macro de Intercambio de datos (**medidas**) nos encontraremos una pantalla como la que se muestra en la figura 3.5. En dicha figura destacan dos bloques principales: **Serial Receive** y **sfuerz2**. Ambos bloques se describen a continuación.

Serial Receive

Figura 3.5: Macro **medidas**

Bloque de las librerías de la tarjeta dSPACE para Simulink que representa el buffer de entrada para el puerto serie. Es decir, existirá una cola de entrada, cuyo tamaño configuraremos a 256 bytes, donde se irán almacenando los bytes recibidos por el puerto serie. Conforme se vayan leyendo esos bytes, la cola se irá vaciando.

sfuerz2

Este bloque corresponde a la *s-function* creada que se encargará de gestionar las comunicaciones por el puerto serie en la fase de intercambio de datos que estamos tratando. A continuación se comentan los elementos que la componen.

1. *Entradas:* Como se puede observar, su entrada está conectada al bloque Serial Receive descrito anteriormente, es decir, en su entrada dispondrá en todo momento de los bytes leídos por el puerto serie. La dimensión del puerto de entrada será de 54 bytes, que corresponde al tamaño de la trama de respuesta del controlador ante una petición de datos.
2. *Salidas:* Se puede observar en la figura que el bloque que tratamos consta de cuatro salidas.

- *Bytestx*: Salida conectada al buffer del puerto serie para el envío de datos. Esta salida tiene dimensión 3 y en ella se escribirán todos los bytes correspondientes a comandos que se desean hacer llegar al controlador. Dichos bytes se almacenarán en el buffer de salida del puerto serie y posteriormente se enviarán.
- *error*: En esta salida se escribirá el valor del bit de error contenido en la trama de datos.
- *fuerza y torsion*: Estas dos salidas tomarán como valores los datos de fuerza y torsión leídos en cada trama de datos recibida. Cabe destacar que ambas salidas constan de tres dimensiones, una para cada componente de fuerza y torsión en el espacio cartesiano:

$$F_x, F_y, F_z, T_x, T_y, T_z$$

3. *Parámetros*: La función consta únicamente de un parámetro que será el tiempo de muestreo. Este parámetro es de vital importancia en nuestra aplicación porque establecerá el tiempo de refresco de los datos de fuerza y torsión. El valor del tiempo de muestreo estará limitado por la velocidad de comunicación del puerto serie. Como se comento en el estudio de velocidades del capítulo 2, para el tipo de comunicación que estamos tratando dispondremos de una trama de datos cada $11.25mseg$ y por tanto ese será el mínimo valor de muestreo que podremos usar para que el protocolo de comunicación funcione correctamente.

El código de la función se muestra en el apéndice A. El algoritmo programado consiste básicamente en pedir una trama de datos al controlador y posteriormente recorrer byte a byte dicha trama de datos identificando cada uno de los campos que en ella aparecen.

En la primera ejecución de la función habrá que tener en cuenta que aún no dispondremos de datos en la entrada, por tanto, la única operación que se llevará a cabo será la escritura en la salida del comando de petición de datos al controlador.

$$QR < CR >$$

Como se puede comprobar, la petición de datos consta de tres bytes, y es por este motivo por el que se ha establecido con dimensión 3 la salida de bytes de la *s-function*.

Una vez ejecutada al menos una vez la función, dispondremos en todo momento en la entrada de nuestro bloque de una trama de datos. Esta trama de datos será la que podremos recorrer byte a byte con el fin de obtener datos de los campos que corresponden a fuerzas y torsiones. Para llevar a cabo esta operación es necesario conocer el formato de la trama de datos. Este formato se mostró en el capítulo 2. Recordemos que tenía la forma:

$\langle ACK \rangle \langle \text{línea de datos} \rangle \langle CR \rangle \langle LF \rangle \langle ACK \rangle \langle CR \rangle \langle LF \rangle \rangle$

Siendo relevante únicamente la parte de $\langle \text{línea de datos} \rangle$ cuyo formato es:

$\langle error \rangle, XXXXXX, XXXXXX, XXXXXX, XXXXXX,$
 $XXXXXX, XXXXXX \langle CR \rangle \langle LF \rangle$

Por tanto, dado que conocemos el formato de la trama de la que disponemos, únicamente habrá que recorrerla identificando los caracteres de separación entre los distintos campos y los valores numéricos de dichos campos. Acto seguido se actualizarán las salidas de error, fuerza y torsión con los valores leídos y además se volverá a solicitar otra trama de datos por el puerto serie haciendo uso del mismo comando detallado anteriormente.

Capítulo 4

Comunicación analógica.

4.1 Introducción

En este capítulo se describe el protocolo de comunicación implementado para llevar a cabo una comunicación directa con el transductor del sensor. En el capítulo 2 se realizó una descripción de la función de cada pin del transductor y también del tipo de comunicación que es necesario establecer con él. Recordemos que nos encontramos frente a un transductor del cual necesitamos obtener 6 voltajes que corresponderán a las caídas de tensión en las galgas, y recordemos también que para conocer estas tensiones es necesario realizar un barrido en código gray porque estas señales se obtienen mediante multiplexión.

4.2 Uso de la tarjeta dSPACE para implementar el protocolo de comunicación

Como se ha comentado en la introducción del presente capítulo, las características de comunicación del transductor no van a volver a ser descritas debido a que esta tarea ya se llevó a cabo en el capítulo 2. Ahora nos limitaremos únicamente a describir cómo se ha implementado el protocolo de comunicación en el PC de control haciendo uso de la tarjeta de comunicaciones dSPACE y de la herramienta *RTW* de Simulink para control en tiempo real.

En primer lugar debíamos seleccionar los canales de la tarjeta que usaríamos

para la comunicación con el sensor. La elección de dichos canales se detalla a continuación. Sobre ella cabe mencionar que ha sido motivada en gran medida por los canales que ya estaban en uso para otras aplicaciones.

1. Como señales analógicas de lectura por parte de la tarjeta dSPACE se hizo uso de los canales analógicos de entrada *ADCH2* y *ADCH4*. Ambos canales tienen un rango de lectura de $\pm 10V$ e internamente se lleva a cabo una conversión a digital de 16 bits cada $4\mu seg$.
2. Para la función de señales de Bias que es necesario hacer llegar al transductor se ha hecho uso de los canales analógicos de salida de la tarjeta *DACH7* y *DACH8*. Ambos canales realizan una conversión interna D/A de 14 bits cada $6\mu seg$ y proporcionan señales analógicas dentro de un rango de $\pm 10V$ y $\pm 25mA$.
3. Por último, para las 3 señales digitales de multiplexión se ha hecho uso de los canales digitales de salida de la tarjeta *IO16*, *IO17* e *IO18*. Todos ellos proporcionan un nivel de tensión propio de un *TTL* en un rango de intensidades de $\pm 10mA$.

El resto de señales que es necesario hacer llegar al transductor corresponden a las señales de alimentación. Todas ellas han sido sacadas de la fuente de alimentación del controlador. Por tanto, dicho controlador va a ser usado únicamente para obtener las tensiones de alimentación, en ningún momento se va a hacer uso de su funcionalidad.

En el siguiente apartado se describen los esquemas creados en Simulink para gestionar todos los canales de entrada y salida de la tarjeta, de forma que se intercambien las señales necesarias para poder conocer las fuerzas y torsiones aplicadas sobre el transductor.

4.3 Descripción de los esquemas de Simulink

Los esquemas creados en Simulink son relativamente simples. Únicamente hay que conocer los bloques de librería de la tarjeta dSPACE que se usan para gestionar los canales de comunicación, y las dos *s-functions* creadas para gestionar dichos canales.

La pantalla principal que contiene todos los bloques es la que se muestra en la figura 4.1. Como puede apreciarse, aparecen una serie de bloques de Simulink que se exponen a continuación.

- *DS1103MUXADC*: Este bloque proporciona acceso a los canales de lectura de la tarjeta. Como parámetros de comunicación necesarios, únicamente es necesario

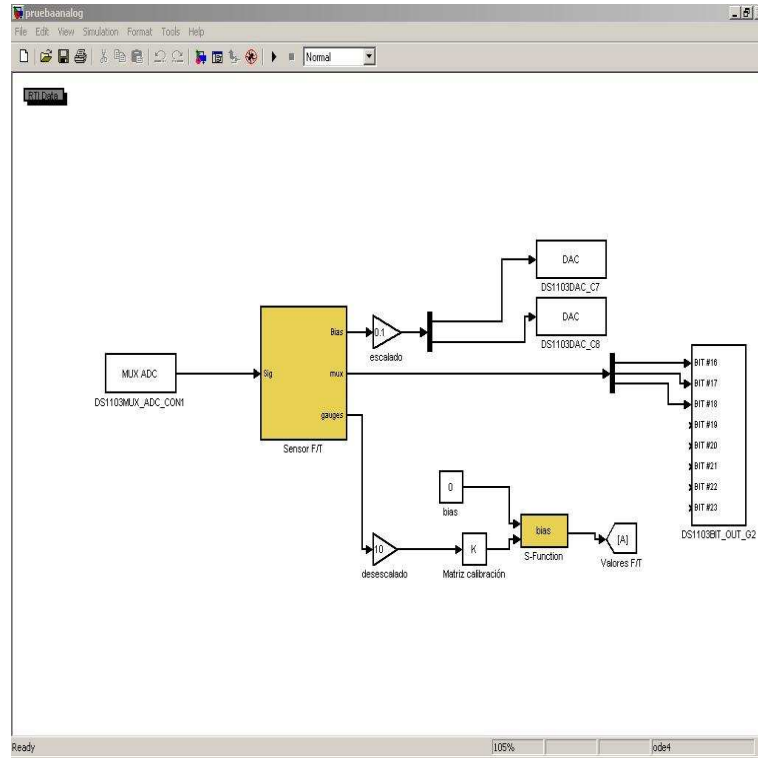


Figura 4.1: Pantalla principal del modelo

indicar los números de los canales analógicos que se desean leer, es decir, los canales 2 y 4.

- **DS1103DAC**: Bloque que accede directamente a las salidas analógicas de la tarjeta. A su entrada lee valores en el rango ± 1 que traduce en tensiones a la salida de $\pm 10V$ por los canales analógicos. De ahí que sea necesario un escalado. Los únicos parámetros que son necesarios configurar son el número del canal que se desea leer, el voltaje inicial que aparecerá en dicho canal y la tensión que quedará en la línea una vez que acabe la ejecución del programa.
- **DS1103BITOUT**: Este bloque permite acceder a los canales digitales de la tarjeta. A su entrada leerá un '1' ó un '0' y esto provocará que en el canal digital correspondiente aparezca un nivel lógico alto o bajo respectivamente. Para configurar este bloque únicamente es necesario indicar el nivel lógico inicial y final que se desea que aparezca en el canal.

Como se puede apreciar, todos los bloques comentados anteriormente no tienen ninguna complicación. El problema ahora consiste en programar una *s-function* que se encargue de gestionar el protocolo de comunicación. Esta *s-function* es la que aparece con el nombre **sensor F/T** y su código fuente lo podemos encontrar en el apéndice A que se entrega como anexo de este documento.

La función **sensor F/T** se ejecuta en seis pasos que se repiten indefinidamente. En cada paso se actualizan las salidas digitales para hacer un barrido en código Gray y de esta forma ir seleccionando cada galga. A su vez también se actualizan las salidas analógicas para enviar la tensión de bias correspondiente a cada galga y por último, también lee el valor de la entrada que corresponderá siempre a la caída de tensión correspondiente a la galga que se seleccionó en la ejecución inmediatamente anterior. Nuestra *s-function* además irá almacenando los valores leídos en cada galga y una vez que haya hecho un barrido de las seis, actualizará la salida correspondiente con estos seis valores leídos.

Vista desde el punto de vista funcional, la *s-function* tiene la misión de sacar por su tercera salida los seis valores de tensiones leídos en las seis galgas. El problema que esto conlleva es que los canales de entrada proporcionan un rango de lectura de $\pm 10V$ pero en el esquema de Simulink, el bloque *DS1103MUXADC* proporciona salidas en un rango de ± 1 . La solución a este problema es muy simple y como se puede observar en el esquema del modelo, únicamente consiste en multiplicar por 10 las señales leídas.

Una vez multiplicadas todas las señales de tensiones leídas por 10 estaremos ya en condiciones de multiplicar por la matriz de calibración y de esta forma obtener finalmente los datos de fuerzas y torsiones medidos por el transductor.

El último bloque que conviene comentar es el que aparece con el nombre de **Bias**. Este bloque en teoría no es necesario, pero se ha incluido porque añade una funcionalidad que será de gran utilidad para futuros experimentos con el sensor. Idealmente podríamos pensar que si no ejercemos ninguna fuerza sobre el sensor, éste debería medir valores nulos de fuerzas y torsiones. Sin embargo, debido a la fuerza de la gravedad y a la propia masa del transductor, esto no será así. En posición de reposo el transductor medirá ciertos valores de fuerzas y torsiones que en teoría deberíamos obviar. Para este fin se ha creado la *s-function* de nombre **Bias**, que básicamente es un bloque de Simulink que dispone de una entrada de selección que nos permite activar su funcionamiento. En el momento en que dicha entrada pase de valor cero a valor uno, la *s-function* almacenará el valor de las medidas de fuerzas y torsiones y a partir de ese momento a su salida en vez de proporcionar directamente los valores leídos por el sensor, a estos valores les restará los que ha almacenado en el momento de su activación. Si en algún instante interesa desactivar esta operación, únicamente será necesario volver a poner su entrada de selección a cero. De esta forma, en cualquier instante posterior que vuelva a activarse, la funcionalidad será la misma.

Los valores finales de fuerza y torsión que deberán usarse para llevar a cabo un control en tiempo real, serán aquellos que se proporcionan a la salida del bloque Bias. Una vez más, el código de dicha *s-function* lo podremos encontrar en el apéndice A.

4.4 Integración en el modelo del RM-10

Anteriormente se ha detallado cómo se ha implementado el protocolo de comunicación con el transductor haciendo uso de Simulink. No obstante, esta explicación se ha llevado a cabo sobre el modelo inicial que se creó, es decir, sobre un modelo que únicamente contiene la parte de comunicación con el sensor. El siguiente paso consistiría en añadir todo este modelo a aquel usado para controlar el robot *RM10*. Aunque no sea objeto del presente proyecto, conviene aclarar que el modelo de control del RM-10 es aquel que contiene toda la configuración para la lectura de los encoders, permite acceder a las señales de control del manipulador, etc. En este modelo se podrá implementar cualquier estructura de control con realimentación de posiciones y velocidades, ya que ambas estarán disponibles en el esquema de Simulink. Es por tanto en este modelo donde debemos integrar la arquitectura diseñada para la comunicación con el sensor de forma que al igual que se disponen de posiciones y velocidades, en futuros experimentos se pueda disponer también de fuerzas y torsiones.

La opción elegida para incluir la parte de comunicación con el sensor en el modelo general ha sido incluirlo todo en una macro con el nombre **SENSOR FUERZA-TORSION**. Esta macro constará de dos salidas de dimensión 3 cada una. La primera salida proporcionará las fuerzas y la segunda las torsiones. Por último se deben añadir dos mantenedores de orden cero (Zero-Order Hold) que serán necesarios debido a los distintos tiempos de muestreo con los que se va a tratar. En la figura 4.2 se muestra el aspecto de la ventana de Simulink donde se ha decidido añadir la macro del sensor. Esta ventana es la misma donde se incluyen las macros de lecturas de los encoders y de actuación de las señales de control.

Con la integración de la estructura de comunicación con el sensor en nuestro modelo general se ha cumplido el objetivo general del presente proyecto, que era básicamente permitir que las señales de fuerza y torsión estuviesen disponibles para en un momento dado poder implementar controladores con realimentación de fuerza y torsión.

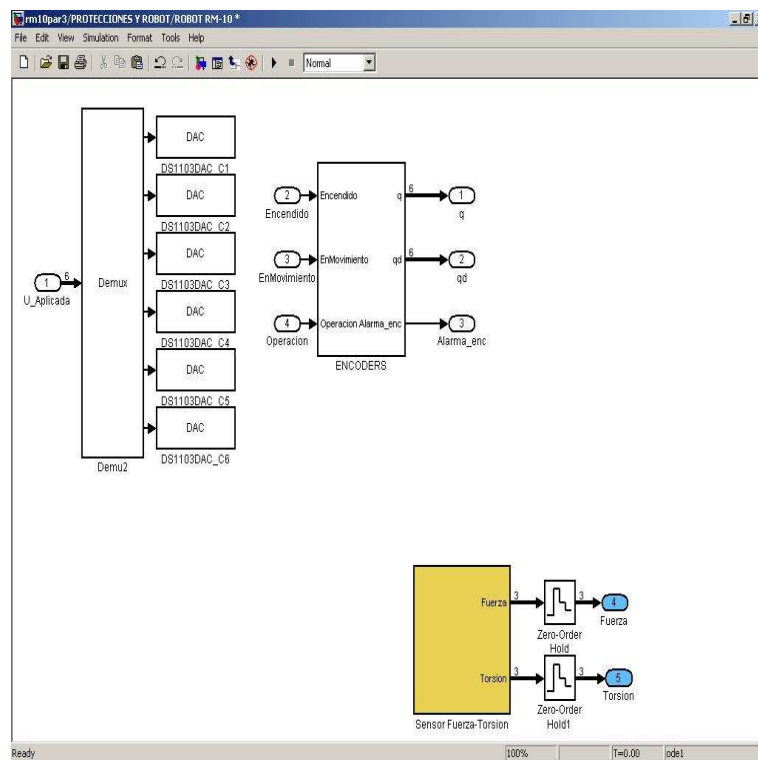


Figura 4.2: Integración de la estructura de comunicación en el modelo general

Capítulo 5

Placa de protección y conexionado.

5.1 Introducción

En el capítulo anterior se ha descrito el protocolo desarrollado para llevar a cabo una comunicación directa con el transductor del sensor haciendo uso de los canales de entrada y salida analógicos y digitales de la tarjeta de comunicaciones dSPACE. No obstante, a la hora de conectar dichos canales de la tarjeta con los pines correspondientes del conector del transductor hay que tener mucho cuidado. Debemos recordar que estamos tratando con componentes de un elevado valor económico y que por tanto debemos proteger al máximo con el fin de no dañarlos.

Debido a que tanto la tarjeta dSPACE como el transductor del sensor son elementos de muy elevado coste, se ha optado por hacer uso de protecciones con aislamiento galvánico para la comunicación en ambos sentidos. De esta forma se protegen ambos elementos frente a cualquier señal de sobretensión ó de sobreintensidad que pudiera aparecer en el circuito debido a algún fallo eléctrico.

A lo largo de este capítulo se describirá tanto la placa de protección creada para llevar a cabo la conexión como todos los conectores y cables que han sido necesarios para conseguir montar el sistema completo.

5.2 Diseño de la placa de protección

A la hora de diseñar la placa destinada a aislar galvánicamente ambos ex-

tremos de la comunicación, lo primero que hay que tener claro es el número de canales analógicos y digitales que hay que proteger y el rango de tensiones permitido para cada uno de ellos. Recordemos que en nuestro caso los canales utilizados van a ser:

- 2 canales de lectura analógicos de la tarjeta (*ADCH2* y *ADCH4*) que corresponden a las señales de datos que salen del sensor (+Sig y -Sig). Ambas deben estar protegidas de forma que su voltaje no supere el rango permitido por la tarjeta dSPACE, es decir, deben estar comprendidas entre $-10V$ y $+10V$.
- 2 canales de escritura analógicos de la tarjeta (*DACH7* y *DACH8*) que corresponden a las señales BIAS de voltaje que es necesario hacer llegar al transductor. Ambas deben estar dentro del rango $\pm 5V$ que es el rango que acepta el transductor.
- 3 canales de salida digitales de la tarjeta (*IO16*, *IO17* y *IO18*) que corresponden a las señales de multiplexión que es necesario hacer llegar al transductor.

Como se puede observar, en total harán falta 3 protecciones para canales digitales, dos protecciones para canales analógicos en el rango $\pm 5V$ y dos protecciones para canales analógicos en el rango $\pm 10V$. Debido a que las técnicas utilizadas para aislar canales analógicos son muy distintas a las técnicas para canales digitales, debemos tratar cada uno por separado, y una vez terminados los diseños agrupar ambos en la misma placa.

5.2.1 Protecciones analógicas

Para lograr aislamiento galvánico en los canales de comunicación analógicos se ha optado por hacer uso del amplificador de aislamiento *ISO122P* del fabricante BURR-BROWN. Dicho amplificador de alta precisión permite proteger en un rango de entrada que puede alcanzar hasta los $1500V_{rms}$ proporcionando a la salida una señal que no se saldrá del rango $\pm 10V$. Este amplificador además permite trabajar con anchos de banda de hasta $50KHz$.

La señal es transmitida digitalmente a través de una capacidad diferencial de barrera de $2pF$. Con una modulación digital, las características de la barrera no afectarán a la integridad de la señal que resultará ser de una alta calidad.

En la figura 5.1 se puede apreciar la asignación de pines del integrado. Como se puede observar, serán necesarias dos fuentes de alimentación independientes (V_{s1} y V_{s2}) para proporcionar energía a ambos extremos del integrado asegurando el aislamiento galvánico. Las alimentaciones requeridas en ambos extremos deben estar comprendidas

entre $\pm 4.5V$ y $\pm 18V$ y deben ser capaces de proporcionar una corriente de $\pm 5mA$ en V_{s1} y $\pm 5.5mA$ en V_{s2} .

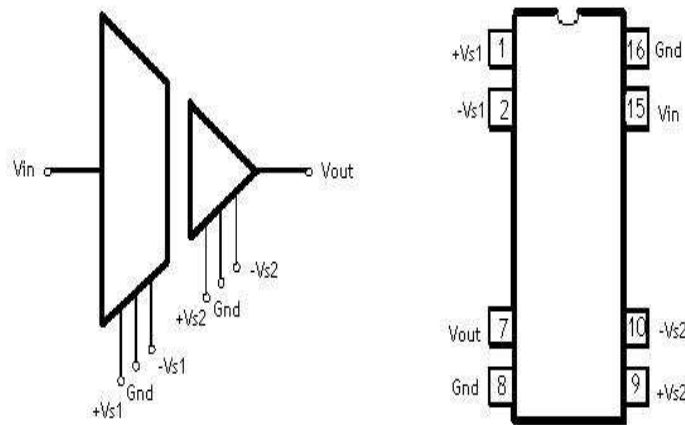


Figura 5.1: Diagrama de conexiones ISO122P

En nuestra aplicación usaremos alimentaciones de $\pm 12V$ que obtendremos en un extremo a partir de la fuente de alimentación del PC y en el otro extremo de la fuente de alimentación del controlador. El motivo por el que haremos uso de la fuente del controlador es que como se ha comentado en el capítulo anterior, esta es la fuente que usaremos para alimentar el transductor, y por tanto, podemos aprovechar que en cualquier caso era necesario sacar tensiones de dicha fuente para no tener que hacer uso de ninguna otra.

El problema que se plantea al hacer uso de la fuente de alimentación del controlador es que dicha fuente proporciona tensiones de $\pm 15V$ mientras que las tensiones que vamos a utilizar deben ser de $\pm 12V$. Para solucionar este problema haremos uso de los reguladores de tensión *LM7812* y *LM7912* que nos permitirán obtener las tensiones deseadas a partir de las del controlador. El esquema utilizado para la incorporación de los reguladores de tensión se muestra en la figura 5.2. Nótese la adición de capacidades que permitirán estabilizar las tensiones de entrada y salida y hacerla más estable frente a transitorios.

Con lo comentado hasta ahora podría pensarse que ya está resuelto el apartado de las protecciones analógicas. Sin embargo queda aún por resolver un pequeño problema. El amplificador de aislamiento elegido proporciona protección en el rango $\pm 10V$

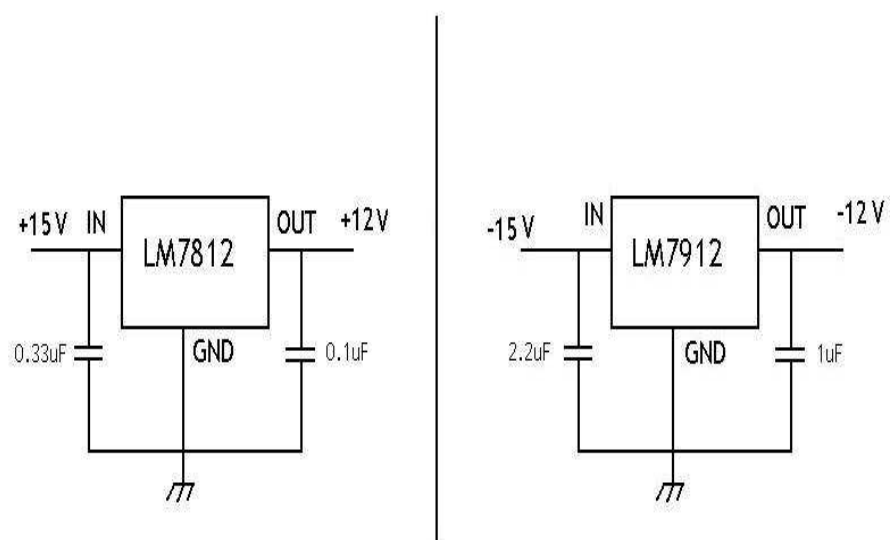


Figura 5.2: Esquema de conexiones para los reguladores de tensión

pero recordemos que las señales BIAS de entrada al sensor no debían exceder del rango $\pm 5V$. Por lo tanto es necesario buscar otro amplificador de aislamiento ó modificar de alguna forma el circuito ideado para cumplir las especificaciones deseadas.

La solución tomada para reducir el rango de tensiones fue el empleo de dos diodos zener que entrasen en conducción siempre que la señal de entrada excediese el rango permitido. Teniendo en cuenta que disponemos de señales de alimentación de valor $\pm 15V$ el circuito ideado para los canales que estamos tratando es el que se muestra en la figura 5.3.

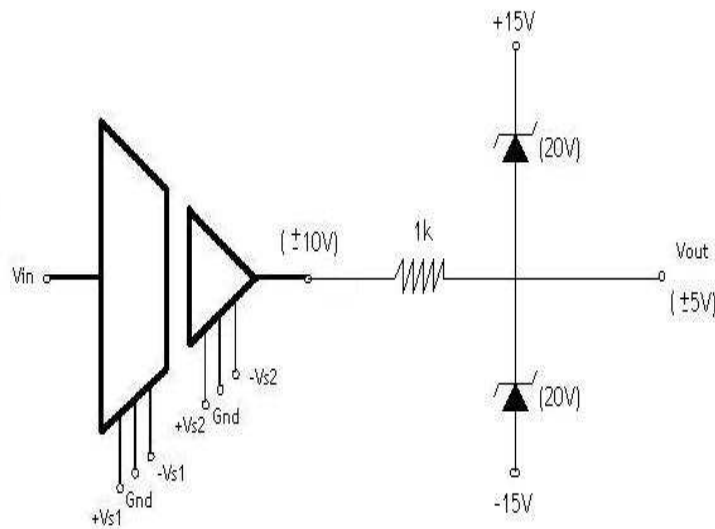


Figura 5.3: Circuito empleado para protección en el rango $\pm 5V$

Como se puede observar, el esquema de protección para el rango $\pm 5V$ se basa en el esquema utilizado para el rango anterior. Hace uso del mismo amplificador de aislamiento (*ISO122P*) para conseguir que la señal esté comprendida entre $\pm 10V$ y se le añaden dos diodos zener para reducir a la mitad ese rango. Como se puede observar en la figura, ambos diodos tendrán una tensión de ruptura de 20 V. De esta forma, suponiendo que la señal a la salida del amplificador supera una tensión de $+5V$, en el zener de abajo se sobrepasará la tensión de ruptura y éste entrará en conducción inversa, impidiéndose de esta forma que la tensión aumente por encima de los $+5V$. Si por el contrario suponemos que la señal alcanza una tensión de $-5V$ a la salida del amplificador, podremos deducir que en el diodo de arriba se sobrepasaría la tensión de ruptura y entraría de igual forma en conducción inversa. De esta forma, la tensión no podría seguir disminuyendo y estaría su mínimo en $-5V$ tal y como se deseaba.

La elección de las tensiones de ruptura en los diodos viene impuesta por la alimentación usada. Se ha optado una vez más por hacer uso de las alimentaciones proporcionadas por el controlador y de esta forma se han obtenido las tensiones de rupturas necesarias en los zener para comprender la señal dentro del rango deseado.

En la figura 5.3 podemos ver además la adición de una resistencia a la salida del amplificador de aislamiento. Esta resistencia es necesaria para cuando los diodos zener entren en conducción, ya que será la resistencia donde caerá todo el voltaje que esté por encima del rango permitido. La elección del valor de esta resistencia se ha hecho teniendo en cuenta dos aspectos:

- Su valor debía de ser suficientemente pequeño para que en el caso de tener una señal aceptable en el rango deseado, no se produjera en ella una caída de tensión apreciable que conllevaría a que la señal de entrada no coincidiera con la de salida. Esto se consigue teniendo en cuenta la impedancia de entrada de la cual dispone el transductor, cuyo valor se midió experimentalmente y resultó ser de $400K\Omega$. Por lo tanto, en caso de que la señal estuviese dentro del rango permitido y los diodos en consecuencia estuviesen en corte, el esquema equivalente sería el que se muestra en la figura 5.4, en el que se puede apreciar que la impedancia de entrada del transductor es la resistencia dominante, y por tanto, es en ella en la que va a caer prácticamente toda la tensión.

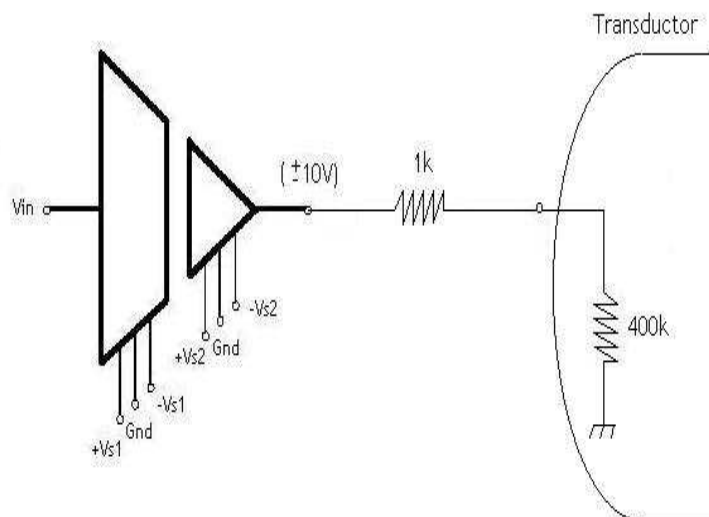


Figura 5.4: Circuito equivalente para ambos diodos en corte

- El segundo aspecto a tener en cuenta era el valor máximo de la corriente que circularía por nuestra resistencia. Si se elegía una resistencia demasiado pequeña, en el caso extremo en el que la señal tuviera un valor de $+10V$ ó $-10V$, la intensidad necesaria para hacer que los $5V$ que exceden del rango impuesto cayeran en la resistencia, podría resultar ser de un valor demasiado elevado para los zener. Por lo tanto, debíamos elegir una resistencia tal que en el caso extremo no necesitase una corriente demasiado elevada para disipar el voltaje que superase el rango deseado.

$$I = \frac{V}{R} = \frac{5V}{1k\Omega} = 5mA$$

Como se puede observar, el valor de la intensidad en el peor caso no es demasiado elevado y resulta ser bastante adecuado para que los diodos zener lo soporten sin ningún problema.

Finalmente, cabe comentar que en el apéndice B se encuentran las versiones finales implementadas de todos los esquemas eléctricos comentados anteriormente.

5.2.2 Protecciones digitales

Para llevar a cabo el aislamiento de los canales digitales usaremos un *fototransistor optoacoplador*. La ventaja de este tipo de dispositivos es que permite implementar un aislamiento galvánico completo entre ambos extremos de la comunicación con un diseño eléctrico muy sencillo y un coste económico bastante reducido.

En concreto haremos uso de optoacoplador *4N25* de *Fairchild-semiconductor*. El esquema de este dispositivo se muestra en la figura 5.5, y como puede observarse consta de un fotodiodo emisor y un transistor que lleva a cabo la tarea de recepción, y que entrará en conducción cada vez que el diodo emita luz. Es decir, en función del estado del fotoemisor, emitiendo o no emitiendo, el transistor del otro extremo entrará en conducción ó quedará cortado dando lugar a un '0' ó un '1' lógico.

Este componente no es tan autónomo como el amplificador de aislamiento usado en las protecciones analógicas en el sentido de que no proporciona la funcionalidad de aislamiento galvánico digital por sí solo, sino que es necesario acompañarlo de algún circuito que fije los rangos de tensiones e intensidades que van a aparecer.

Para diseñar nuestro circuito debemos tener en cuenta en primer lugar que estos canales serán usados para las líneas de multiplexión del transductor. Estas líneas estarán referidas siempre a la línea de tierra de la alimentación, y una tensión por encima de $2.4V$ será considerada como un '1' lógico mientras que una tensión por

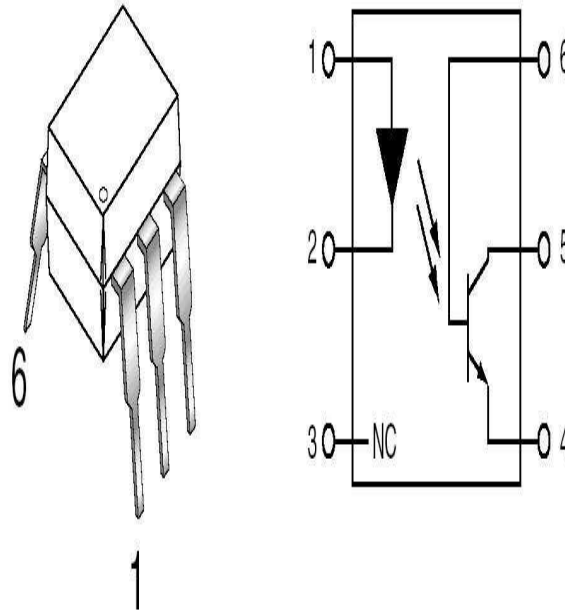


Figura 5.5: Esquema del optoacoplador 4N25

debajo de $0.8V$ será considerada como un '0' lógico. En las tensiones intermedias no se debe trabajar porque no está determinada la asignación que llevará a cabo el transductor. En cualquier caso además, sabemos que nunca se deberán sobrepasar las tensiones de alimentación.

Teniendo en cuenta todo esto, y sabiendo que el rango de tensiones de salida de la tarjeta dSPACE para sus canales digitales es el propio de las tensiones en transistores TTL, se diseñó un circuito que se puede observar en la figura 5.6.

Si observamos la figura, vemos que en la parte izquierda (lado de la tarjeta dSPACE) únicamente se le añade una resistencia de 250Ω . Si la salida de la tarjeta presenta un nivel bajo ($0V$) el fotodiodo permanecerá en corte y no emitirá luz. Si por el contrario la salida de la tarjeta presenta un nivel alto, circulará una determinada corriente por el diodo que hará que éste emita luz y el transistor que funciona como detector entre a su vez en conducción. Como puede intuirse, el valor de la resistencia ha sido elegido para regular la intensidad que circulará por el fotoemisor de forma que dicha intensidad esté dentro del rango admitido por dicho diodo.

Centrándonos ahora en el otro extremo (parte izquierda), vemos que en el caso en el que el transistor no conduzca, el voltaje a la entrada de multiplexión del sensor será nulo y por tanto, se interpretará como un cero lógico. Por el contrario, en los casos en los que el transistor conduzca, la caída de tensión que se producirá en la entrada de

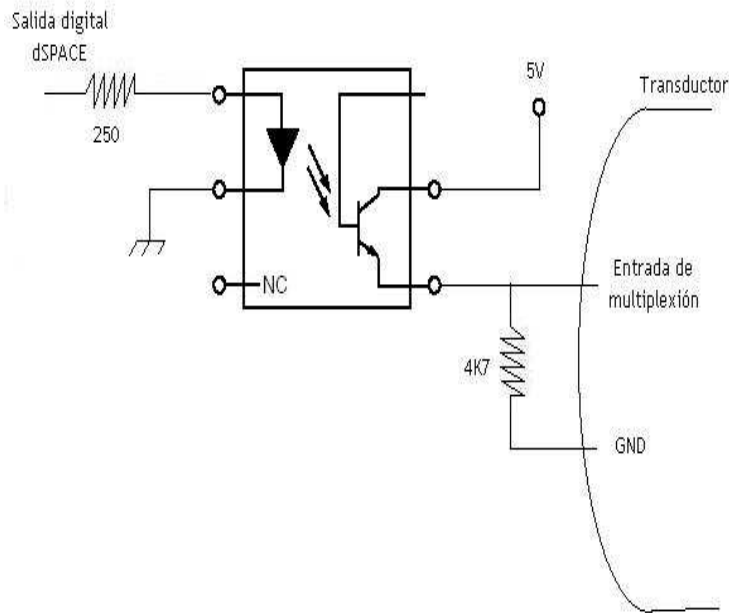
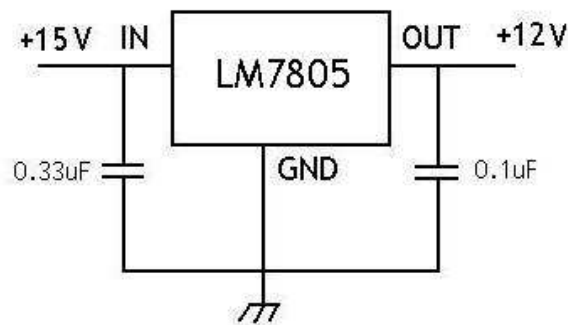


Figura 5.6: Circuito diseñado para protecciones digitales

multiplexión será interpretada como un uno lógico. En este caso además, la resistencia añadida de valor $4K7\Omega$ quedará en paralelo con la impedancia de entrada del sensor. El valor de dicha resistencia ha sido elegido para que en este caso, al ser una resistencia de menor valor que la impedancia de entrada, la mayor parte de la corriente caiga en ella de forma que no se inyecte toda la corriente procedente del transistor en el sensor.

Por último, cabe comentar que debido a los umbrales de tensión interpretados por el sensor como unos y ceros, se ha optado por alimentar el emisor del transistor con una tensión de $5V$. Para obtener esta tensión una vez más hemos recurrido a la fuente de alimentación del controlador. Para poder obtener una tensión de $+5V$ a partir de los $+15V$ que proporciona el controlador se ha hecho uso del regulador de tensión *LM7805*. El esquema de conexiones para dicho regulador coincide con los empleados con los reguladores comentados en apartados anteriores y lo podemos ver en la figura 5.7.

Todas las conexiones descritas para este canal digital habría que repetirlas de forma idéntica para los tres canales digitales de comunicación. El pequeño circuito de alimentación (que incluye el regulador a $+5V$) es válido para alimentar los 3 optoacopladores. Los diseños finales de los circuitos se encuentran en el apéndice B.

Figura 5.7: Esquema para el regulador de tensión *LM7805*

5.2.3 Diseño final de la placa

En cuanto al diseño final de la placa únicamente cabe comentar que se ha optado por implementar en una misma placa todas las protecciones (analógicas y digitales). En el apéndice B se puede ver el esquemático final diseñado en ORCAD, donde se puede apreciar que la placa consta de 4 amplificadores de aislamiento *ISO122P* para los canales analógicos y tres optoacopladores *4N25* para los canales digitales. Asimismo, la placa contiene los tres reguladores de tensión comentados en la sección anterior (*LM7805*, *LM7812* y *LM7912*) y a su vez contiene todos los componentes auxiliares que son necesarios y que han sido comentados en el apartado anterior.

En cuanto a los componentes usados en la placa para poder conectar las distintas señales necesarias, podemos decir que se han usado borneras estándar en las cuales únicamente es necesario introducir los cables correspondientes y acto seguido atornillarlos para fijarlos. En total la placa consta de 5 borneras dobles a cada lado que comprenden un total de 20 señales que es necesario conectar. En la figura 5.8 se puede observar una foto de la placa acabada y en la figura 5.9 encontramos la lista de las señales correspondientes. La numeración para llevar a cabo la asignación de los pines consiste en usar la letra 'D' para las bornas que aparecen en la parte inferior en

la foto y la letra 'I' para las que aparecen en la superior. El número 1 correspondería a la borna más a la izquierda y el 10 a la situada más a la derecha.

Como se puede observar además en la figura 5.8, se ha optado por introducir la placa en una caja de plástico con el fin de aislarla del exterior y a su vez, con el fin de facilitar las tareas de conexionado. Para hacer uso de la placa de protecciones, no será necesario conocer la correspondencia eléctrica de cada una de las señales que entran o salen de la placa, ya que haciendo uso de la caja de aislamiento y de una serie de conectores estándar, se ha desarrollado una interfaz en la que todas las conexiones son a través de conectores *DB9* y *DB15* muy fáciles de conectar y desconectar. En la siguiente sección se describe el conexionado realizado y por tanto se describe también cómo es la interfaz de conexiones desarrollada.

5.3 Conexionado

Como se ha comentado en la sección anterior, para facilitar el conexionado de todas las señales que intervienen en la comunicación entre el sensor y la tarjeta de comunicaciones dSPACE, se ha optado por hacer uso de una serie de conectores estándar. Gracias a esto, futuros usuarios del sistema no deberán conocer la implementación eléctrica del sistema, sino únicamente deberán llevar a cabo una serie de conexiones inmediatas que se describen a continuación.

5.3.1 Conexión fuente de alimentación del controlador-tarjeta de protecciones

Para llevar a cabo la conexión entre la fuente de alimentación del controlador y la tarjeta de protecciones se ha hecho uso de conectores *DB9*. El motivo por el que se ha usado este tipo de conectores ha sido que era necesario tomar de la fuente 4 señales distintas, y por tanto, se ha elegido un conector estándar muy conocido buscando que sobren algunos pines por si en un futuro es necesario hacer uso de alguno de ellos para transmitir alguna otra señal.

Los pasos llevados a cabo para realizar las conexiones han sido:

1. Fue necesario en primer lugar abrir la caja del controlador y haciendo uso de un polímetro, buscar algún punto donde fuese viable sacar las señales de alimentación necesarias. Una vez encontrados estos puntos de conexión, se cablearon hasta un

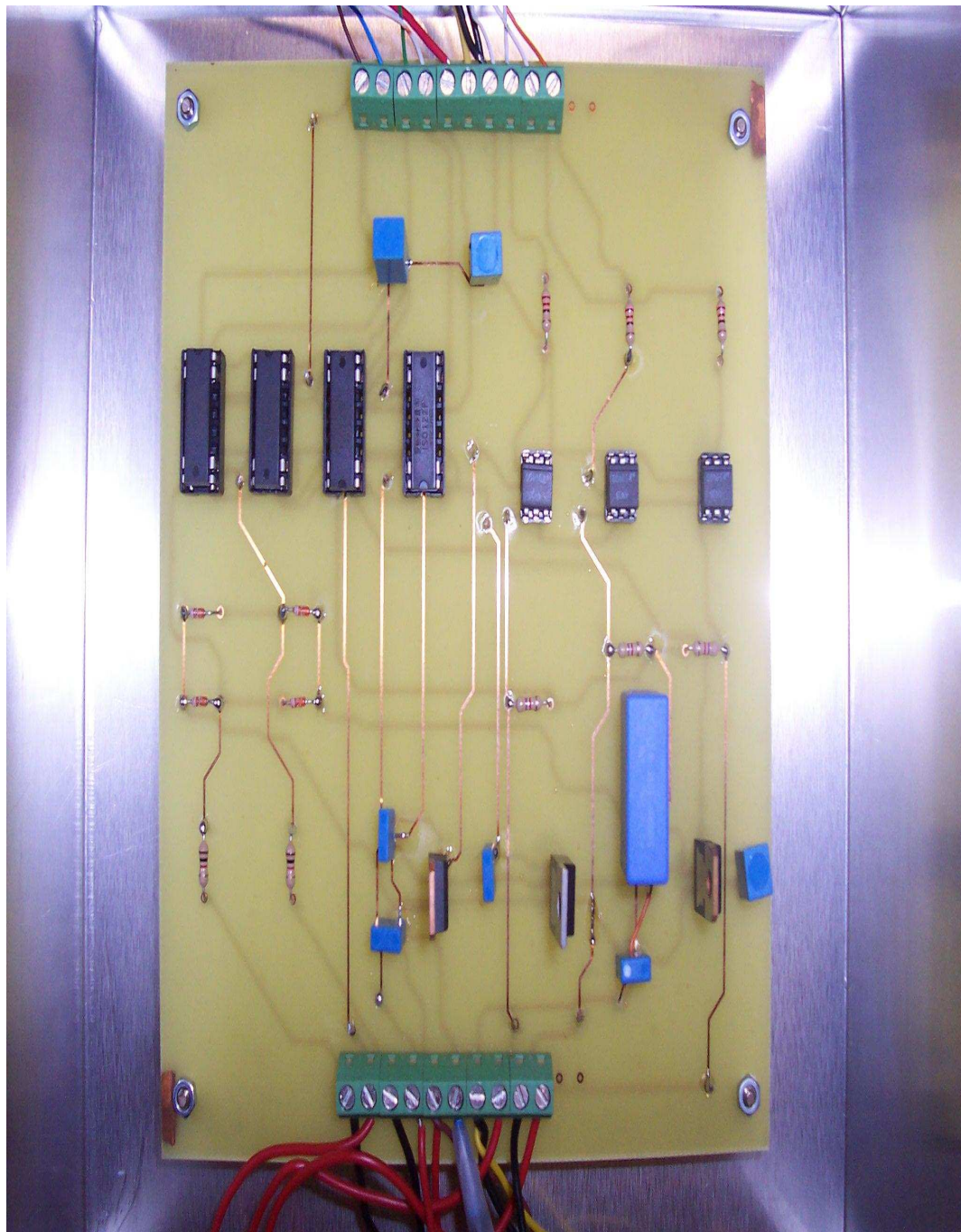


Figura 5.8: Aspecto final de la placa de protecciones

(PARTE DEL SENSOR)		(PARTE DE DSPACE)	
BORNA	ASIGNACIÓN	BORNA	ASIGNACIÓN
D1	+Bias	I1	DACH7
D2	-Bias	I2	DACH8
D3	+Sig	I3	ADCH2
D4	-Sig	I4	ADCH4
D5	+15V	I5	+12PC
D6	GND	I6	-12PC
D7	-15V	I7	GND
D8	MUX1	I8	I/O16
D9	MUX2	I9	I/O17
D10	MUX3	I10	I/O18

Figura 5.9: Correspondencias entre conectores y señales

conector hembra *DB9* que se añadió en la parte trasera de la caja del controlador, y que serviría posteriormente como punto de conexión externo para obtener las señales de alimentación (ver figura 5.10).

2. En segundo lugar se cablearon los pines de la tarjeta de protección correspondientes a las alimentaciones del controlador hasta un conector *DB9* situado en la caja de aislamiento de dicha tarjeta.
3. Finalmente se creó un cable que serviría como alargadera para realizar la conexión entre la caja del controlador y la caja de aislamiento de la tarjeta de protección.

Es importante destacar la necesidad de cuidar al máximo la asignación de pines elegidos del conector. La elección tomada debemos tenerla en todo momento presente y respetarla al cablear ambos interfaces (controlador y tarjeta de protección). En la figura 5.11 se muestra la asignación de pines impuesta en los conectores *DB9*.

En la figura 5.12 se muestra el aspecto final de la conexión entre el controlador y la tarjeta.



Figura 5.10: Cableado interno de las señales de alimentación del controlador

PIN	SEÑAL
1	+15V
5	-15V
6	GND
9	+10V

Figura 5.11: Asignación de pines para interfaz controlador-Tarjeta de protección



Figura 5.12: Conexionado final controlador-Tarjeta

5.3.2 Conexión transductor-tarjeta de protecciones

Para llevar a cabo la conexión con el transductor únicamente es necesario cablear los pines correspondientes de la tarjeta de protección hasta un conector *DB15* de dos filas situado en el frontal de la caja de aislamiento de la tarjeta siguiendo la asignación de pines impuesta por el fabricante del sensor y que se comentó en el capítulo 2 (figura 2.8).

El cable que servirá de unión entre el transductor y el conector añadido en la caja de la tarjeta será el que servía en el sistema inicial para conectar el transductor con el controlador. El aspecto final del conexionado se muestra en la figura 5.13.

5.3.3 Conexión fuente de alimentación del PC-tarjeta de protecciones

Para tomar las señales de alimentación del PC se ha hecho uso de un conector

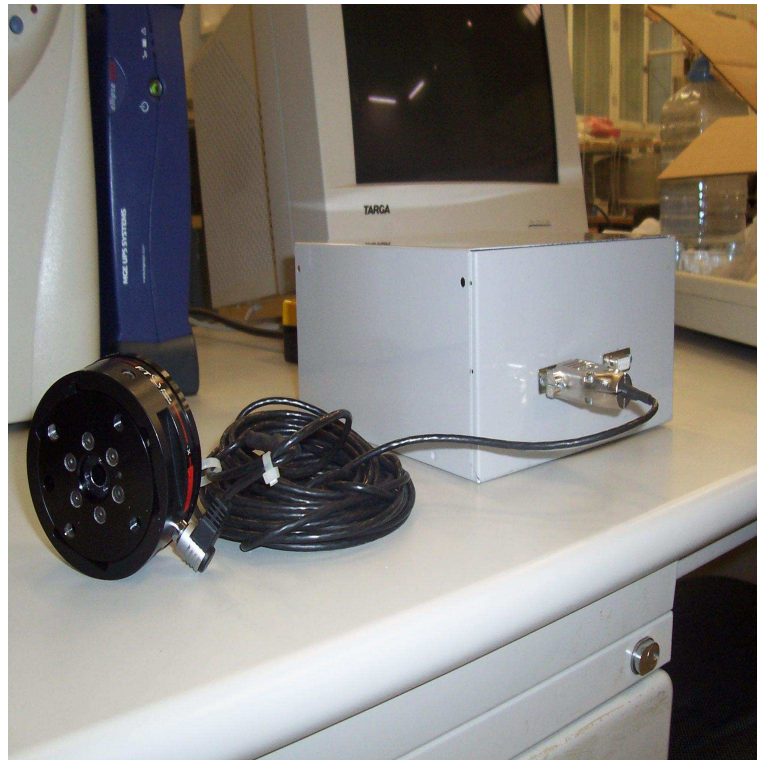


Figura 5.13: Conexionado final Transductor-Tarjeta

de alimentaciones estándar de 4 pines. Resultó necesario cablear las señales correspondientes a alimentaciones en ambos extremos así como crear un cable que sirviera como alargadera para la conexión.

5.3.4 Conexión tarjeta dSPACE-Tarjeta de protecciones

El procedimiento para llevar a cabo las conexiones entre la tarjeta dSPACE y la tarjeta de protección es idéntico al comentado en los apartados anteriores. En primer lugar fue necesario cablear en el extremo de la tarjeta dSPACE haciendo uso de un conector *DB15* de dos filas. En segundo lugar fue necesario cablear en el extremo de la tarjeta de protecciones y finalmente, se creó un cable que serviría como alargadera.

El principal inconveniente de este conexionado particular fue una vez elegidos los canales analógicos y digitales de la tarjeta dSPACE, hacer uso del manual de dicha tarjeta para identificar los pines que era necesario cablear.

Capítulo 6

Pantallas creadas para ControlDesk.

6.1 Introducción

En el presente capítulo se describen las pantallas creadas en ControlDesk que sirven como interfaz de usuario para poder visualizar los datos correspondientes a la ejecución de los modelos creados en Simulink.

En primer lugar se hablará de las pantallas de pruebas creadas para probar los algoritmos de comunicación con el sensor. Una de ellas sirvió para probar la comunicación por el puerto serie y la otra para probar la comunicación directa con el transductor. En ambos casos fue de gran ayuda crear estas pantallas de prueba porque facilitó enormemente las funciones de depuración.

Por último se describirán brevemente las modificaciones que se llevaron a cabo en la pantalla global de ControlDesk creada para controlar todas las señales correspondientes al RM-10. Esta pantalla global está asociada al modelo global de Simulink citado en el capítulo anterior y una vez más podemos interpretar los cambios realizados como una integración del sensor en el modelo general del que se disponía al comenzar este proyecto.

6.2 Pantalla de pruebas para la comunicación por el puerto serie

Como se ha ido comentando a lo largo del presente documento, el primer protocolo de comunicación que se implementó fue a través del puerto serie. Para ello hacíamos uso del puerto serie que incorpora la tarjeta dSPACE y lo conectábamos al controlador del sensor que a su vez se comunicaba con el transductor.

Aunque el protocolo de comunicación no parece ser complicado a priori, fueron necesarias muchas pruebas hasta conocer exactamente los bytes que se volcaban sobre la línea serie cada vez que se enviaba un comando. Pese a que en el manual de usuario del sensor se describen las tramas de datos, existen algunos caracteres de control que no se aclaran por completo. Por tanto la única solución para hacer funcionar la comunicación fue crear una pantalla de pruebas en ControlDesk que nos permitiera visualizar en todo momento los bytes que se estaban volcando sobre la línea serie y la interpretación de la trama de datos que el protocolo programado estaba llevando a cabo.

El aspecto final de la pantalla de pruebas creada es el que se muestra en la figura 6.1. A continuación se comentan cada uno de los elementos que componen esa pantalla.

- *Datos enviados por el puerto serie:* En el capítulo 3 en el que se describía el protocolo de comunicación mediante el puerto serie se explicaba que en cada instante de muestreo se enviarían un total de 3 bytes por el puerto serie. En la esquina superior izquierda de la figura 6.1 se pueden observar los tres displays que se han añadido a la pantalla de pruebas para poder monitorizar estos bytes.
- *Datos recibidos por el puerto serie:* Justo debajo de los displays descritos en el párrafo anterior se añadieron otra serie de displays que nos permiten visualizar las tramas de datos completas que se reciben por el puerto serie.
- *Elementos de control:* Como se puede comprobar existe una serie de botones actuadores que nos permiten cambiar en línea el valor de ciertos parámetros de la comunicación. Todos estos parámetros fueron descritos en el capítulo 3 y nos limitamos ahora únicamente a nombrarlos: *Tipo de dato, selección de modo, Enable, Bias*.
- *Elementos de visualización de los resultados:* Para comprobar finalmente que el protocolo de comunicaciones funciona correctamente se añadieron unos displays que permitieran conocer en todo momento las fuerzas y torsiones que se estaban midiendo. Adicionalmente se incluyeron una serie de gráficas que nos permitían además representar esos datos.

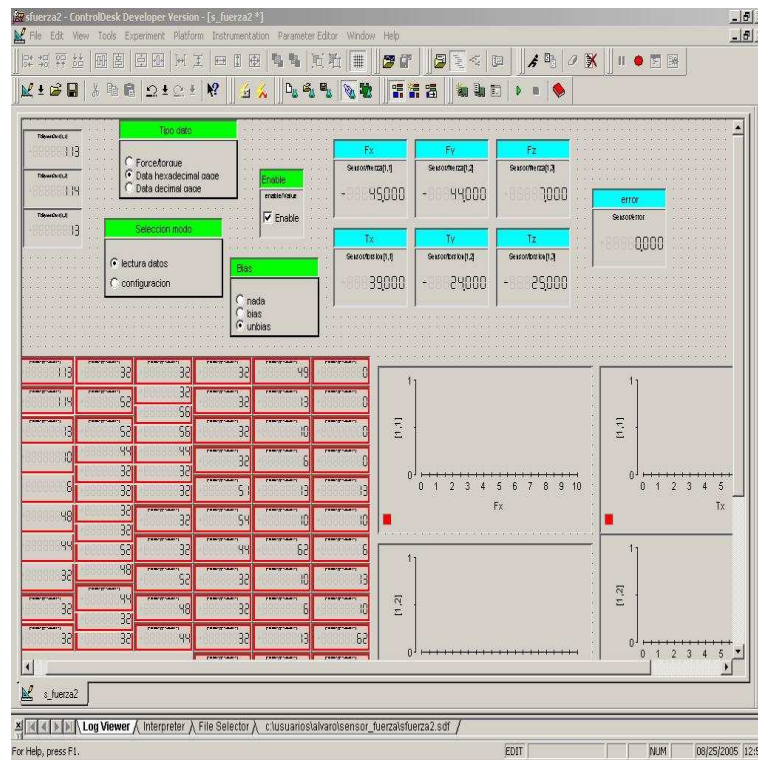


Figura 6.1: Pantalla de ControlDesk para probar el protocolo serie

Resulta importante comentar que gracias a esta pantalla de pruebas se pudo comprobar además el hecho de que el protocolo de comunicaciones no funciona correctamente para periodos de muestreo inferiores a $2mseg$.

6.3 Pantalla de pruebas para la comunicación analógica

En el caso de la comunicación analógica la pantalla de pruebas no resultó ser tan decisiva en el sentido de que no fue necesario realizar muchas pruebas para comprobar que el protocolo funcionaba. Nos referimos en este caso a un protocolo mucho más definido y trabajado sobre el que no existían tantas ambigüedades, y por tanto, la pantalla de pruebas sirvió únicamente para asegurarnos que como era de esperar, el protocolo funcionaba correctamente.

En la figura 6.2 se muestra el aspecto de la pantalla de explotación creada. Como se puede observar esta pantalla es mucho más simple que la comentada en el apartado anterior. En este caso únicamente ha sido necesario visualizar los valores de las tensiones leídas por los canales analógicos, los valores que se escribían sobre los canales analógicos (BIAS) y digitales (Multiplexión) de salida, el valor de las tensiones

resultante en cada galga y por último las fuerzas y torsiones resultantes tras multiplicar por la matriz de calibración. Adicionalmente se incluyó el valor de las fuerzas y torsiones que se ven a la salida de la función de Bias para comprobar su correcto funcionamiento.

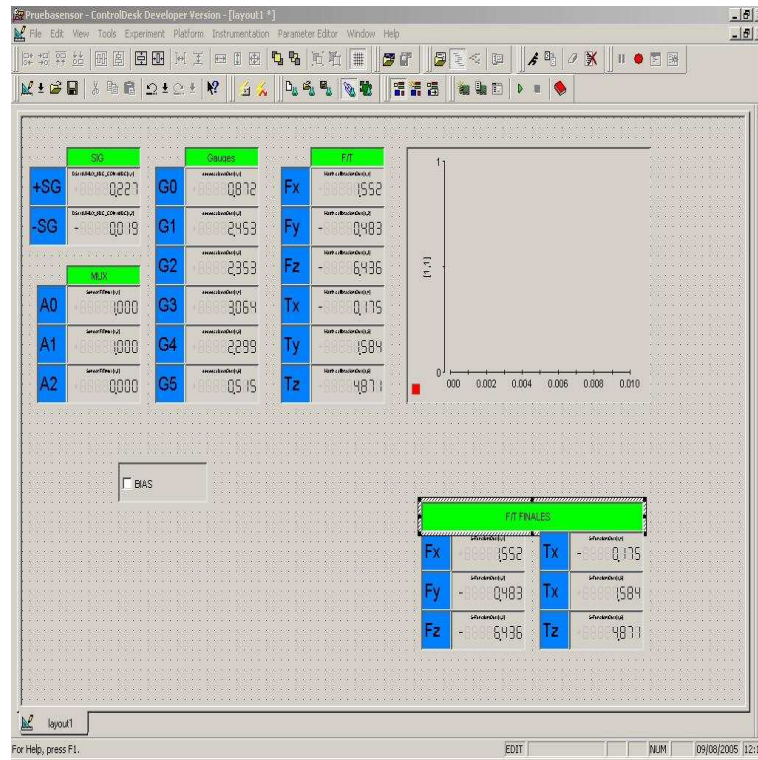


Figura 6.2: Pantalla de ControlDesk para probar el protocolo de comunicación analógica

6.4 Integración en la pantalla general de control del RM-10

Una vez comprobado el correcto funcionamiento de los protocolos de comunicación, el siguiente paso era modificar de alguna forma la pantalla general de la que se disponía para el control del *RM* – 10 con el fin de visualizar las fuerzas y torsiones que se estaban ejerciendo sobre el sensor.

En primer lugar se hizo lo propio para la comunicación por el puerto serie. En este caso además de la visualización de fuerzas y torsiones era necesario añadir elementos que nos permitiesen actuar sobre los parámetros de configuración de la configuración. Sobre esta pantalla no se desea comentar nada en detalle porque finalmente

se va a hacer uso siempre de la comunicación analógica que resulta ser más rápida. En cualquier caso, en la figura 6.3 se muestran las modificaciones incluidas sobre la pantalla de explotación general.

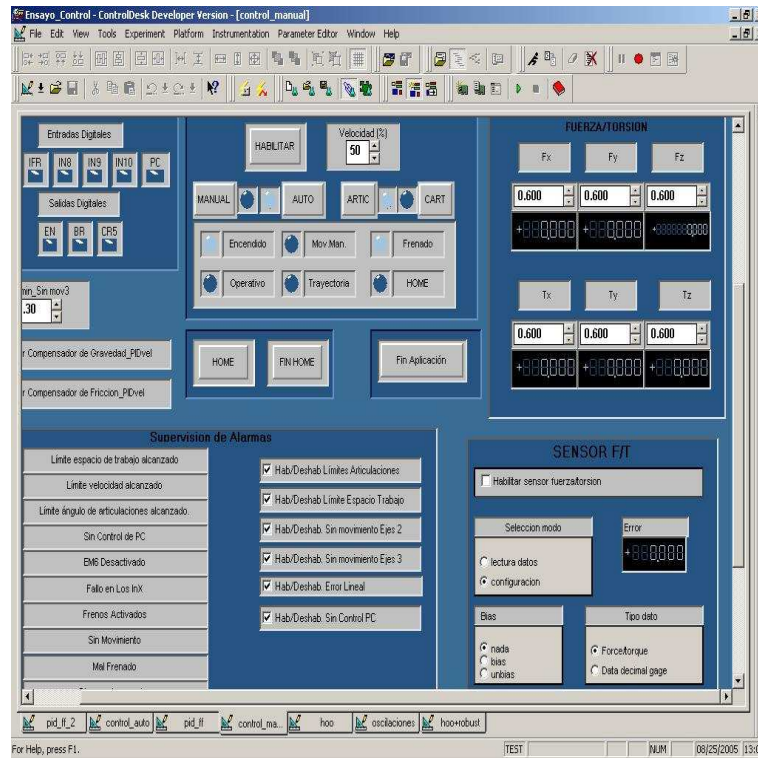


Figura 6.3: Integración del protocolo serie en la pantalla de explotación global

Finalmente, en la figura 6.4 se muestra el aspecto final de la parte de la pantalla de explotación global donde se han introducido los cambios relativos al sensor de esfuerzo. Debido a la simplicidad de las operaciones realizadas únicamente es interesante mostrar la figura y comentar que se ha añadido una pestaña de control que permite actuar sobre la función de Bias comentada en el capítulo 4 en la descripción del protocolo analógico.

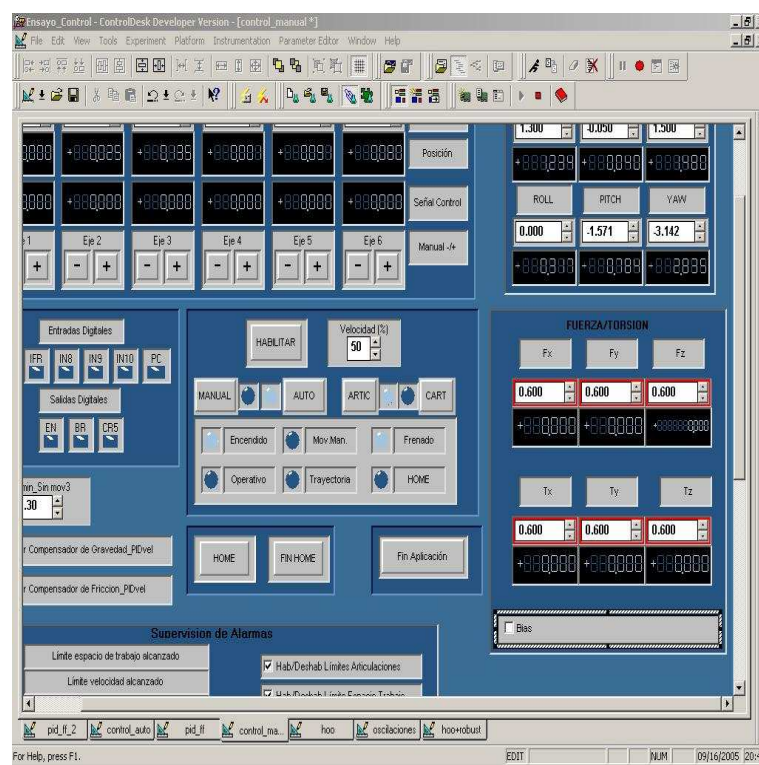


Figura 6.4: Integración del protocolo de comunicación analógica en la pantalla de explotación global

Capítulo 7

Acoplamiento del transductor al efector final del manipulador.

7.1 Introducción

Una vez resueltos todos los problemas de comunicación con el sensor y una vez que disponemos de las medidas de fuerza y torsión de forma correcta, el siguiente paso es acoplar el transductor del robot a la muñeca del RM-10. Para ello es necesario quitar la garra de la que disponía el manipulador y diseñar una serie de piezas que permitan la sujeción del transductor del sensor a la muñeca del robot.

Para la fabricación de las piezas de acoplamiento decidimos hacer uso de la máquina de control numérico de la que disponemos en el Laboratorio del Departamento de Ingeniería de Sistemas y Automática. Otra opción habría sido encargarla a alguna empresa de fabricación, pero debido a su elevado coste decidimos tomar el primer camino.

Una vez estudiado el diseño del transductor y del efector final del RM-10 se pensó en el número de piezas que nos harían falta y se estudió una aproximación de lo que podría ser el diseño de estas piezas. Aprovechando que los alumnos de la asignatura *Laboratorio de Informática y Robótica industrial* tenían impuesto en el plan de dicha asignatura el uso de la máquina de control numérico, optamos por plantearles la fabricación de las piezas como trabajo final de la asignatura.

De esta forma, se le pasaron las especificaciones a los alumnos encargados del proyecto de la asignatura y una vez aprendieron a hacer uso de la máquina de control numérico, se pusieron a trabajar para conseguir fabricar las piezas.

A continuación se hace una breve introducción al equipo empleado para la creación de las piezas y se describe además someramente el proceso de fabricación así como los resultados obtenidos.

7.2 Descripción de la fresadora EMCO PC MILL 125

La máquina que queremos describir se trata de una máquina herramienta controlada mediante CNC desarrollada para el ámbito de la didáctica fundamentalmente, aunque perfectamente válida para uso industrial en la fabricación de pequeñas piezas. Se trata de una máquina herramienta de tres ejes (existen máquinas de CNC que tienen la posibilidad de incorporar un cuarto eje adicional para determinados trabajos).

En la figura 7.1 se muestra el aspecto general del frontal de la máquina herramienta, en ella podemos comprobar dos zonas diferenciadas de la misma, la zona de mecanizado y la zona de control y programación.

La parte de la izquierda (figura 7.2) es la zona de mecanizado, que como su propio nombre indica se trata de la zona de la máquina donde se lleva a cabo el mecanizado de las piezas a desarrollar. Los elementos que componen dicha zona se describen a continuación:

1. Elemento de iluminación de la zona de mecanizado.
2. Elemento para lubricación de las piezas y herramientas durante las operaciones de mecanizado para refrigerar las mismas, disminuir el desgaste de la herramienta y facilitar la eliminación de virutas de la zona de mecanizado. En esta máquina se encuentra anulado y en su lugar se encuentra acoplado un dispositivo de aire comprimido para desalojar las virutas de la zona de mecanizado.
3. Puerta de seguridad accionada mediante cilindro neumático de doble efecto y válvula 5/3 centro cerrado. El cilindro neumático se encuentra provisto de detectores magnéticos que verifican la condición de puerta cerrada para permitir la puesta en marcha del centro de mecanizado.
4. Husillo de la fresa y torreta revolver portaherramientas. El husillo de la fresa es accionado mediante motor de alterna cuya velocidad es variable desde el control. En este centro de mecanizado, vemos que a diferencia de los centros de mayor envergadura el husillo de la fresa y la torreta portaherramientas se encuentran en la misma unidad (se describirá con mayor profundidad más adelante).



Figura 7.1: Imagen frontal de la fresadora completa

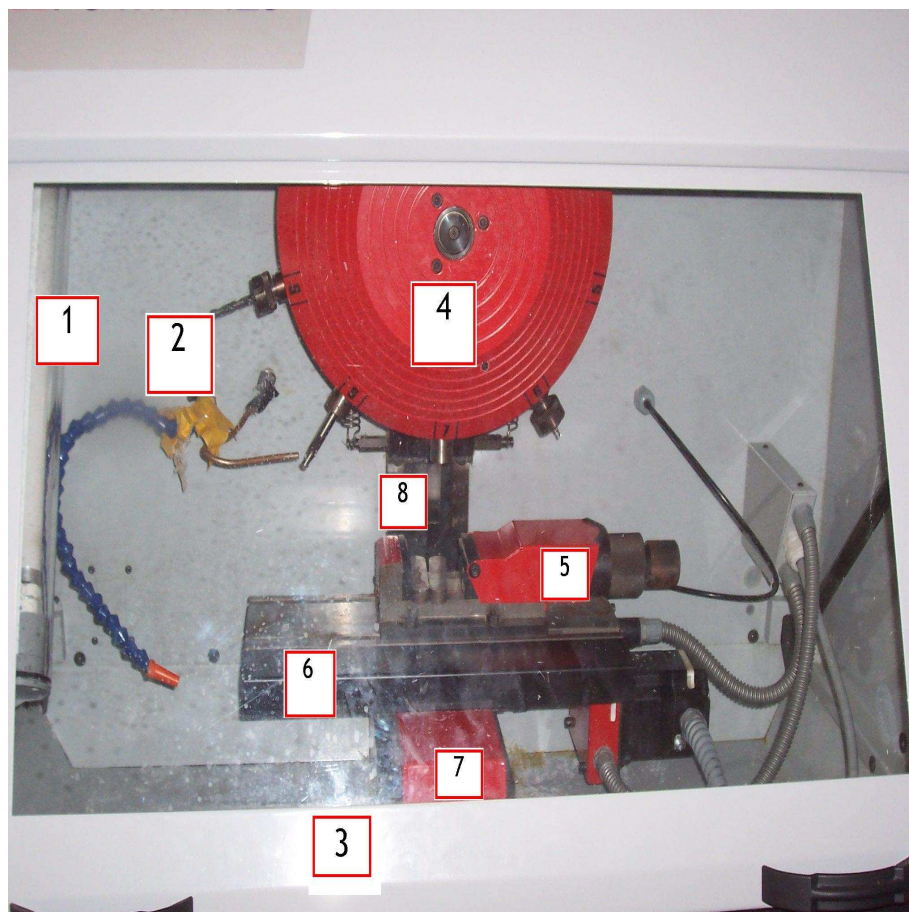


Figura 7.2: Parte de mecanizado

5. Tornillo de sujeción de piezas a mecanizar. Está provisto de un husillo manual para realizar la sujeción de la pieza y un sistema neumático que dar el apriete final para evitar que la misma se desplace durante las operaciones de mecanizado.
6. Mesa de la fresadora y eje X de movimiento. Este carro se encuentra accionado al igual que los otros dos (y, z) mediante un motor paso a paso y husillo de bolas recirculantes, lo que le confiere al mismo una gran precisión de posicionamiento (resolución = $0'0025mm$).
7. Eje Y de movimiento. Presenta las mismas características funcionales que los otros ejes.
8. Eje Z de movimiento. Presenta las mismas características funcionales que los otros ejes.

7.2.1 Carros de desplazamiento

La fresadora dispone de tres ejes de desplazamiento independientes que nos permitirán posicionarnos en cualquier punto del espacio dentro de la zona de trabajo útil de la misma. La zona útil de trabajo vendrá delimitada por el recorrido máximo permitido en cada uno de los citados ejes, siendo estos los que a continuación se detallan:

- Eje X: $185mm$
- Eje Y: $125mm$
- Eje Z: $100mm$

7.2.2 Tambor Portaherramientas

Como su propio nombre indica es el elemento donde se encuentran alojadas las herramientas, este elemento recibe también el nombre de torreta revolver por su forma de operar y dispone de 10 posiciones para la colocación de las mismas. El tambor portaherramientas se encuentra solidario al Eje Z, éste dispone de un movimiento a izquierdas o a derechas independiente del proporcionado por los ejes para llevar a cabo el cambio de herramienta.

Las herramientas empleadas en el CNC son intercambiables dependiendo de la operación de mecanizado que necesitemos realizar. Para la fabricación de nuestras piezas se han empleado las herramientas que a continuación se detallan.

- **Fresa de acabado (I):** Fresa helicoidal de diámetro 40mm empleada para operaciones de acabado de superficies. Puede ser empleada para realizar operaciones de fresado lateral y frontal por la tipología de sus dientes.
- **Fresa de acabado (II):** Fresa helicoidal de diámetro 12mm empleada para operaciones de acabado de superficies y cajas circulares. Puede ser empleada para realizar operaciones de fresado lateral y frontal por la tipología de sus dientes.
- **Fresa de debaste:** Fresa helicoidal de diámetro 40mm empleada para operaciones de desbaste de superficies, dada la tipología de sus dientes.
- **Brocas:** Brocas de diámetros 6 y 3 mm empleadas para realizar los taladros pasantes.

7.2.3 Zona de programación y control

Si anteriormente se ha descrito la zona de mecanizado, ahora debemos comentar la otra parte principal de la que consta la máquina de fresado. La zona de Programación y Control es la zona de la máquina donde se encuentra el cerebro del CNC, en ella se podrán escribir los programas para la ejecución de piezas, se podrán ejecutar comandos de forma individual, realizar ajustes de la máquina (necesarios antes de ejecutar cualquier programa), etc. En la figura 7.3 se muestra dicha zona y en ella se pueden comprobar varias zonas diferenciadas.

1. Monitor 15" para visualizar coordenadas, velocidades, programas, etc. También podremos visualizar las trayectorias de la herramienta cuando trabajamos en modo simulación (muy útil para la verificación de programas).
2. Teclas de funciones. La función asociada a cada una de éstas se corresponde con las mostradas en la pantalla de visualización. Se emplean para recorrer los menús de que dispone el CNC como corrección de herramientas, decalaje del origen, edición de programas, directorio de usuario, simulación, etc.
3. Teclado alfanumérico y funcional genérico propio del control de ejes Sinumerik. A través de este teclado se pueden desarrollar todos los programas del CNC, ejecutar comandos, resetear fallos, etc.
4. Teclado específico del CNC para realizar funciones, entre ellas podemos destacar las de movimientos de ejes en modo manual, apertura y cierre de la mordaza (neumática), apertura y cierre de la puerta de seguridad, cambio manual torreta revolver, etc.



Figura 7.3: Parte de programación y control

5. Selector del modo de operación del CNC, mediante el mismo podremos seleccionar los modos de operación que deseamos emplear; entre los modos más usuales tenemos, Punto referencia, Automático, Jog o Incremental
6. Selector de cambio de la velocidad de avance de los carros. Se selecciona un porcentaje con respecto a la velocidad máxima la cual es fijada en el programa en curso. Este selector es muy empleado durante la ejecución de las piezas por el operario de la máquina, para controlar las velocidades de avance de la herramienta en las distintas trayectorias.
7. Esta es la zona de seguridad de la maquina herramienta, en ella podemos encontrarnos con una seta de emergencia para paro total de la máquina en cualquier situación de movimiento; pulsador que junto con las teclas de apertura y cierre de puerta de seguridad proporcionan un "sistema dos manos" para evitar atropamientos del operario; y finalmente llave de seguridad para posibilitar el funcionamiento del CNC.

7.3 Diseño del acoplamiento para el sensor de fuerza al robot

Los datos de partida de los cuales se disponen para el diseño de las piezas son los planos del efector final del robot y del transductor del sensor (figuras 7.5 y 7.4). En ellos aparecen además todas las medidas de longitudes, diámetros, roscas, tolerancias, etc. Que van a ser necesarias para llevar a cabo el diseño.

El otro punto de partida necesario eran las especificaciones que acordamos teniendo en cuenta la aplicación para la que iban a ser utilizadas las piezas.

1. El acoplamiento debía ser ligero.
2. La distancia entre el transductor y el efector final del manipulador debía ser lo más pequeña posible.
3. El acoplamiento proporcionado debía ser un acoplamiento firme.
4. El montaje y desmontaje debía ser sencillo.

Con estos dos puntos de partida, y tras varios diseños iniciales todos encaminados en la línea del diseño final se llegó a la solución que a continuación se presenta:

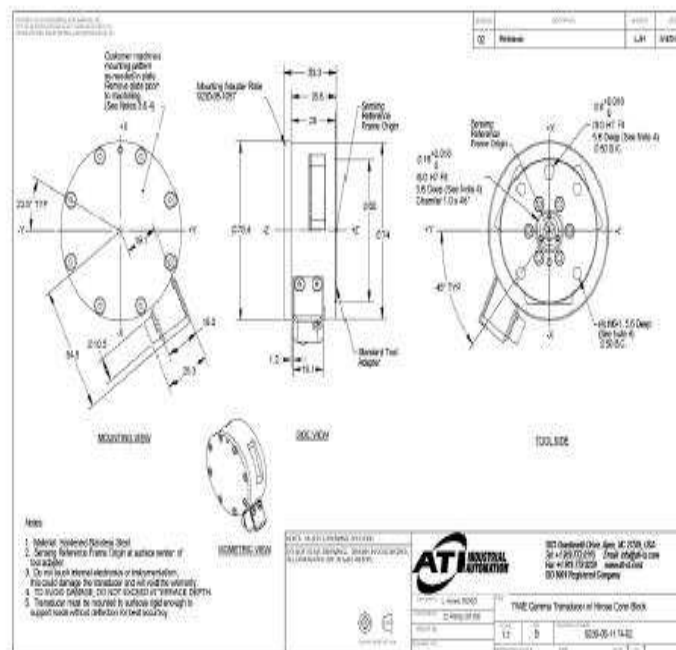


Figura 7.4: Planos del transductor Gamma

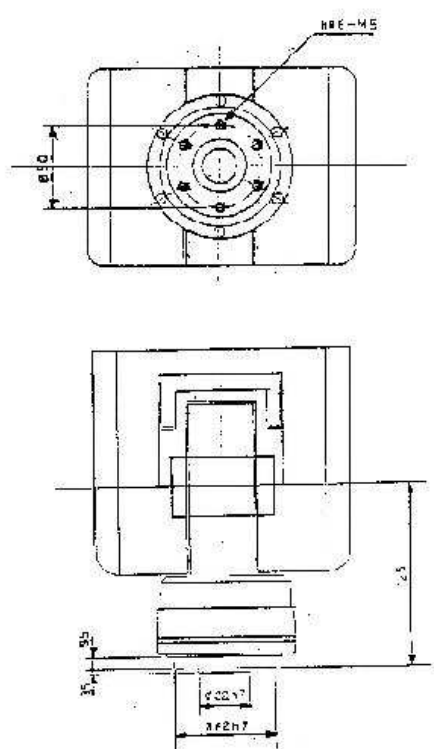


Figura 7.5: Planos del efector final del manipulador

- El acoplamiento estará realizado en dos piezas, una se fijará al efector final del robot mediante los seis taladros roscados de M5 que éste presenta.
- La otra pieza se fijará al sensor de fuerza aprovechando los ocho taladros roscados de M3 de que éste dispone.
- Finalmente ambas piezas se unirán mediante dos tornillos de M10 con sus tuercas correspondientes.
- Las dos piezas comentadas anteriormente permitirán unir el sensor de fuerza al efector final del manipulador. No obstante, se creará una tercera pieza que permitirá acoplar una varilla al transductor del sensor para obtener un punto de contacto con el entorno mucho más concentrado.

7.3.1 Piezas de unión entre el sensor y el manipulador

En la figura 7.6 se muestran los planos de las piezas que se fabricaron finalmente, y en la figura 7.7 se muestra una imagen virtual creada en AUTOCAD del aspecto que presentaría el sistema completo.

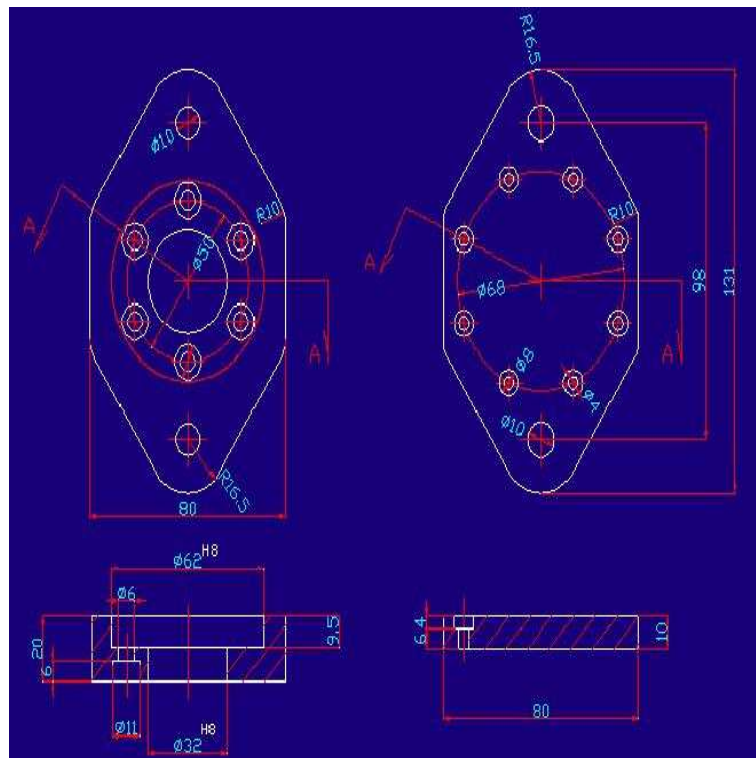


Figura 7.6: Planos de las piezas de acoplamiento

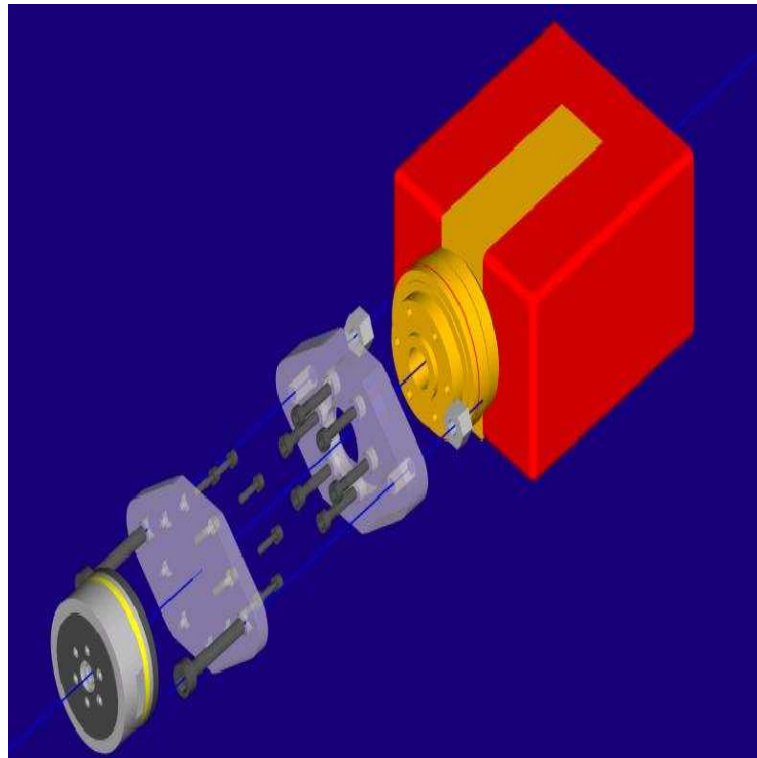


Figura 7.7: Imagen virtual del acoplamiento

Las piezas mostradas en la figuras son las finalmente diseñadas, sin embargo, cuando se procedió a mecanizar las mismas, el material del que se disponíamos era de dimensiones inferiores al ancho requerido por las piezas (anchura de la barra de metacrilato igual a 59 mm.) La solución fue rediseñar sobre la marcha las piezas y el programa manteniendo el diseño original con lo cual las piezas que se mecanizaron en la fresadora fueron las mostradas en la figura 7.8.

Por último se muestran algunas fotos tomadas de las piezas resultantes y del efector final del manipulador una vez montado y acoplado el transductor del sensor de fuerza. Como se puede observar, los resultados obtenidos son muy satisfactorios resultando finalmente un conjunto bastante consistente y compacto. (ver figuras 7.9, 7.10, 7.11 y 7.12).

Se puede observar que aparecen en las fotos anteriores las piezas pero fabricadas en metal. Esto es debido a que uno de los alumnos encargados de la fabricación de las piezas trabajaba en una empresa de fabricación, con la fortuna para nosotros de que , haciendo uso de los mismos planos, nos hizo el favor de fabricar las mismas piezas en un material metálico más resistente e igual de pesado que el metacrilato.

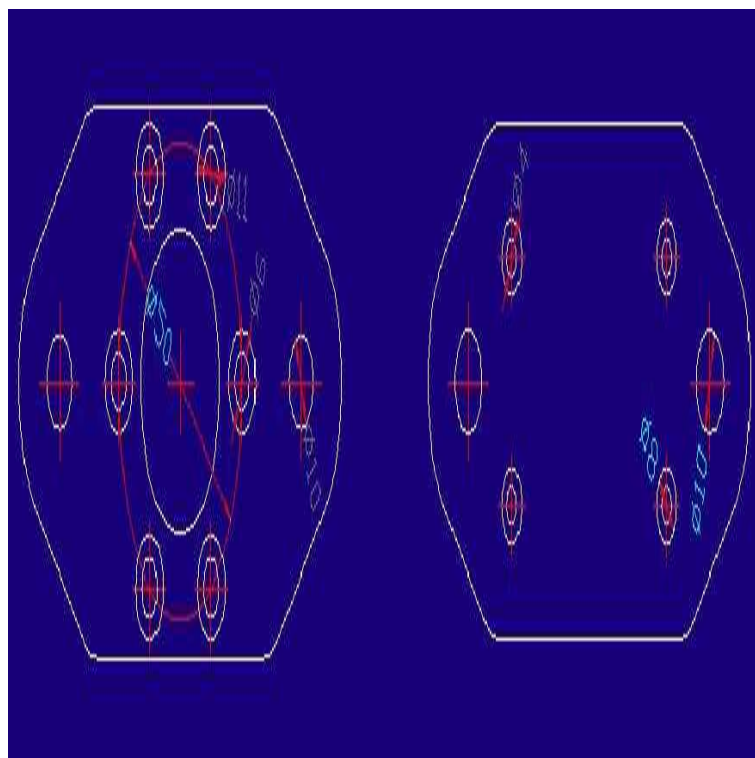


Figura 7.8: Planos de las piezas de acoplamiento modificados

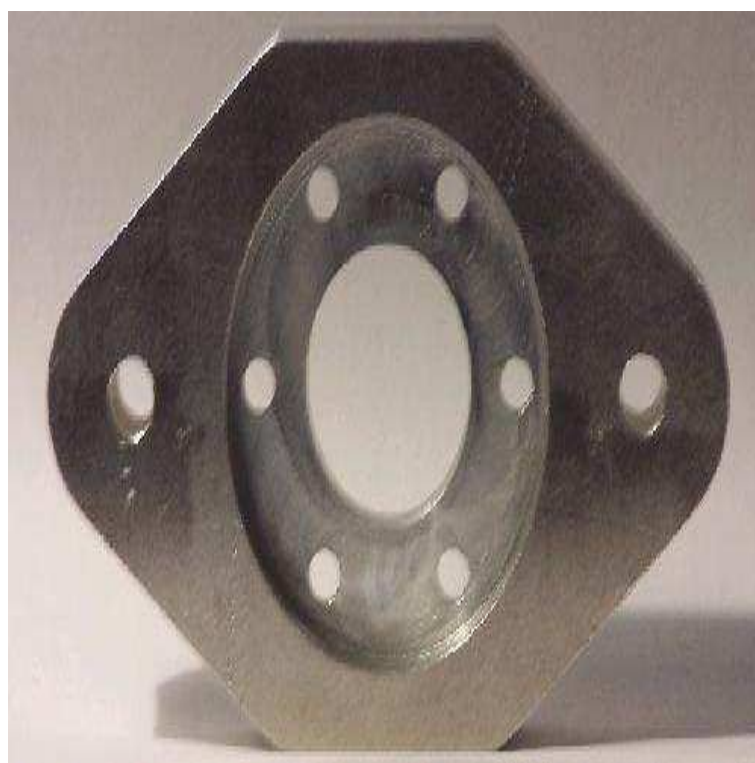


Figura 7.9: Pieza 1

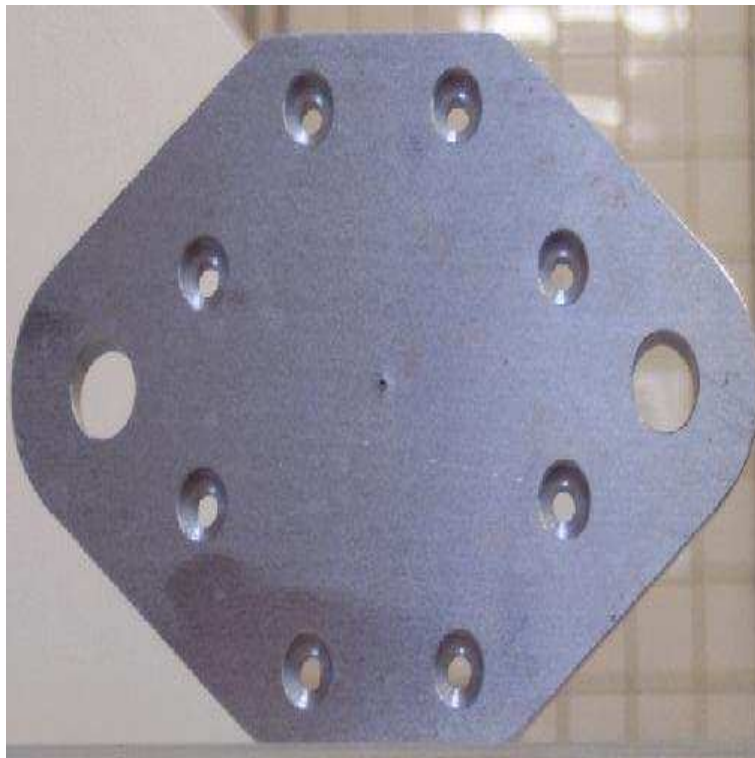


Figura 7.10: pieza2



Figura 7.11: Imagen frontal del conjunto montado

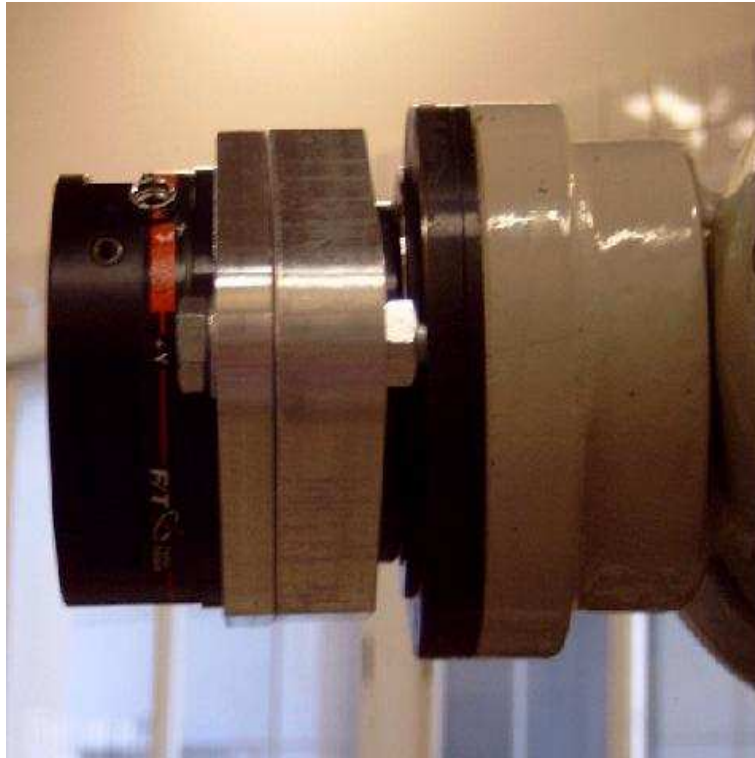


Figura 7.12: Imagen lateral del conjunto montado

7.3.2 Pieza adicional para poder añadir una varilla de contacto al transductor

El objetivo ahora era fabricar una pieza con el fin de acoplar una varilla al medidor de esfuerzos. Este dispositivo iría instalado en la última articulación del robot manipulador RM10. A través de la varilla se transmitirán los esfuerzos al sensor para su medición.

El diseño de la pieza en principio era libre, pero teniendo en cuenta que el diámetro máximo no debe superar los 60mm para no obstaculizar el movimiento de los mecanismos del sensor. Además había que respetar la posición y tamaño de los cuatro tornillos usados para su fijación, así como la cavidad situada en el centro de la pieza para albergar la varilla.

Aunque la pieza de la figura 7.13 es la diseñada inicialmente, la construida finalmente es la que se muestra en la figura 7.14. Como se puede ver, la pieza final ha sufrido ciertas modificaciones debido principalmente al tamaño de la pieza en bruto.

A continuación se muestran una foto de la pieza final una vez construida (figura 7.15) y una foto del sistema completo con las tres piezas diseñadas montadas (figura

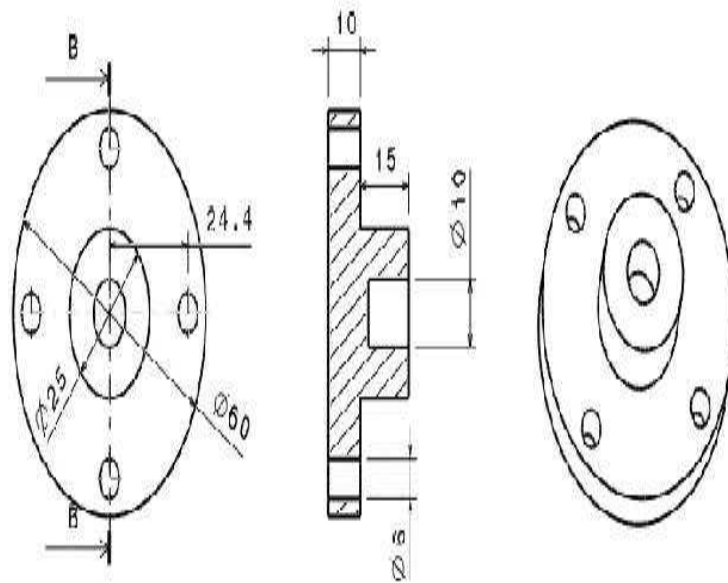


Figura 7.13: Diseño de la pieza inicial

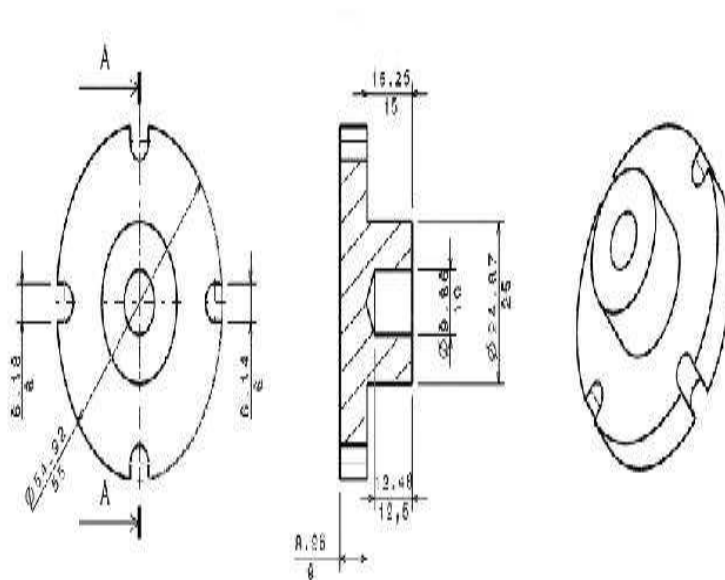


Figura 7.14: Diseño de la pieza final

7.16).



Figura 7.15: Pieza obtenida

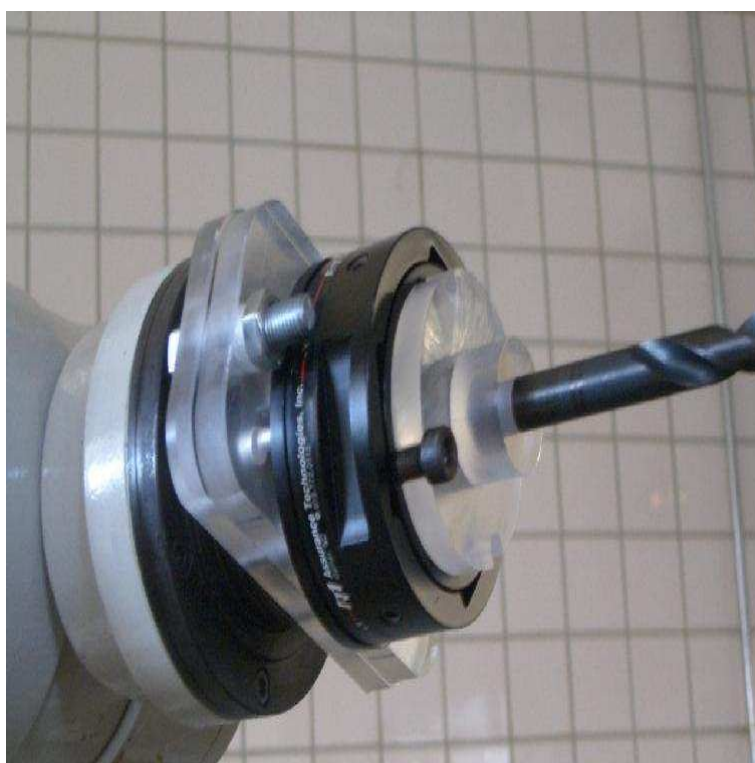


Figura 7.16: Montaje final del sistema completo

Capítulo 8

Pruebas de funcionamiento.

8.1 Introducción

La principal prueba que nos ha permitido comprobar el correcto funcionamiento del protocolo de comunicaciones con el sensor ha sido la posibilidad de comparar los resultados obtenidos con los proporcionados haciendo uso de el software de control del sensor que se proporcionaba con el sensor. Se ha mencionado en más de una ocasión a lo largo del presente documento la existencia de un software de usuario creado por la empresa que fabrica el sensor, que gracias a una conexión mediante el puerto serie con el controlador, permite visualizar los datos de fuerzas y torsiones aplicadas sobre el transductor. Aunque este software era insuficiente para llevara cabo un control en tiempo real, es cierto que las medidas proporcionadas por éste son correctas, y por tanto, para comprobar el correcto funcionamiento de la plataforma creada únicamente fue necesario la comprobación de que los datos coincidían.

Para poder dejar constancia en este documento del correcto funcionamiento del sistema mediante alguna prueba real se han llevado a cabo una serie de pruebas experimentales. Como se puede intuir, las posibilidades que se plantean para estas pruebas se basan en métodos eurísticos de comprobación, que consisten en ejercer fuerzas sobre los distintos ejes del transductor y comprobar visualmente que las medidas de fuerzas y torsiones varían precisamente en los ejes afectados. Para mostrar este tipo de pruebas de funcionamiento se muestran a continuación los resultados obtenidos en algunos experimentos simples que actúan directamente sobre el eje **z**, que resulta ser el más accesible del transductor.

Todas las gráficas que se mostrarán se han obtenido mediante experimentos en tiempo real con *ControlDesk*. Para ello se guardaban las simulaciones en ficheros *.mat* y después se llevaban a cabo las representaciones.

8.2 Prueba de Bias y de medición de F_z

La primera prueba llevada a cabo consistía únicamente en colocar el sensor en una posición de reposo y a partir de dicha posición iniciar la toma de medidas. Pasado cierto tiempo (algo más de 2seg), se accionaría la función de Bias para comprobar que todas las fuerzas y torsiones pasaban a ser cero a partir de dicho instante. Finalmente, pasado otro cierto intervalo de tiempo se colocaría una pesa de $5Kg$ sobre el eje z del sensor, pudiéndose comprobar que la medición de dicho eje cambiaba bruscamente en ese instante mientras que las demás mediciones apenas se veían afectadas.

A continuación se muestran las gráficas correspondientes a las mediciones de fuerzas y torsiones en cada uno de los ejes de coordenadas.

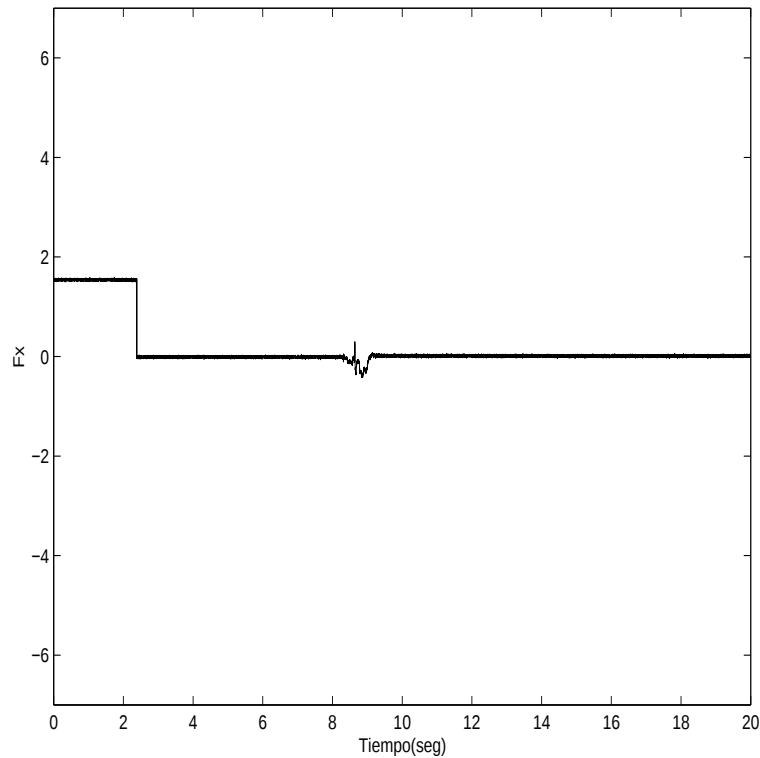


Figura 8.1: Fuerza en el eje x

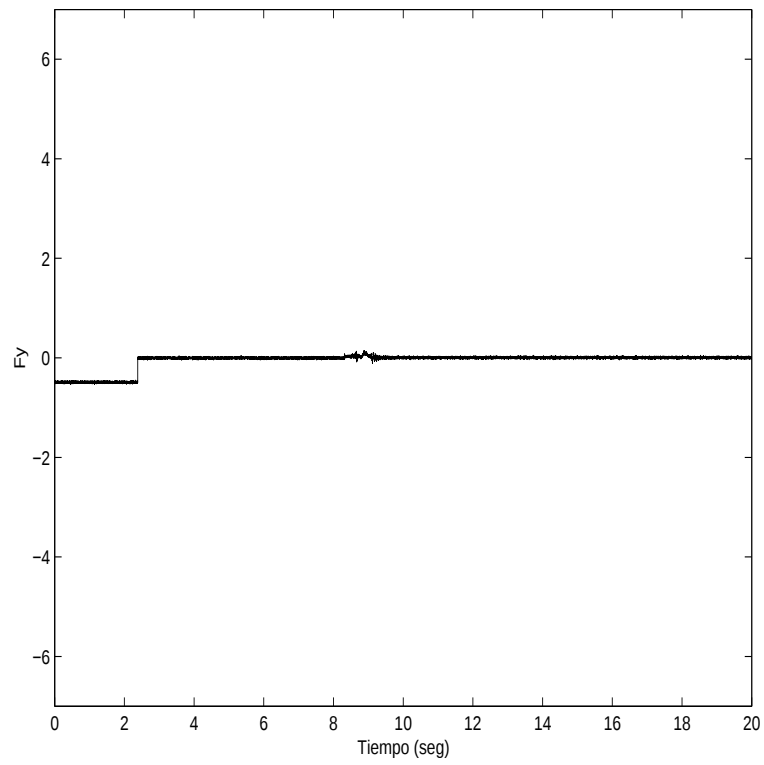


Figura 8.2: Fuerza en el eje y

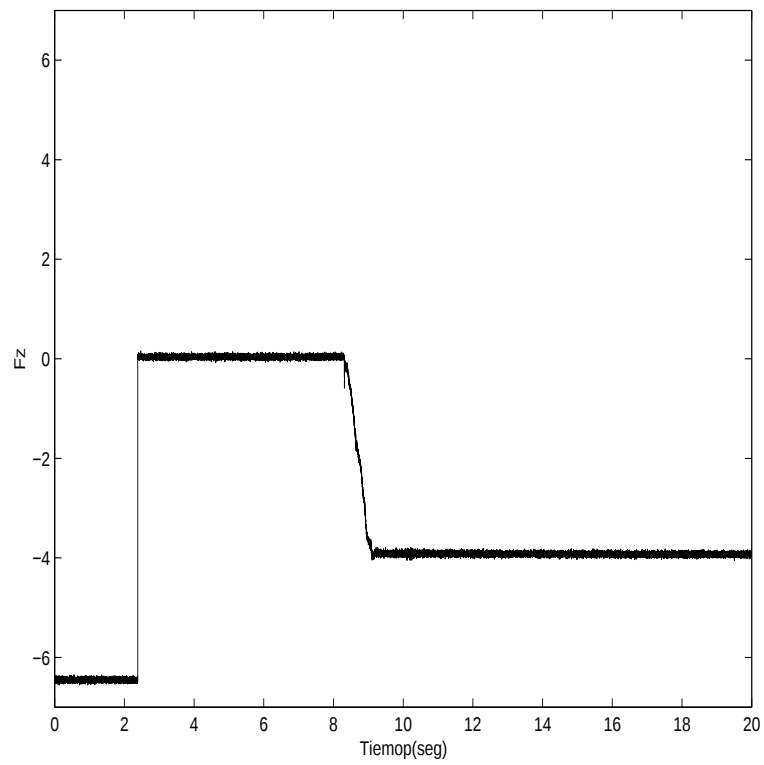


Figura 8.3: Fuerza en el eje z

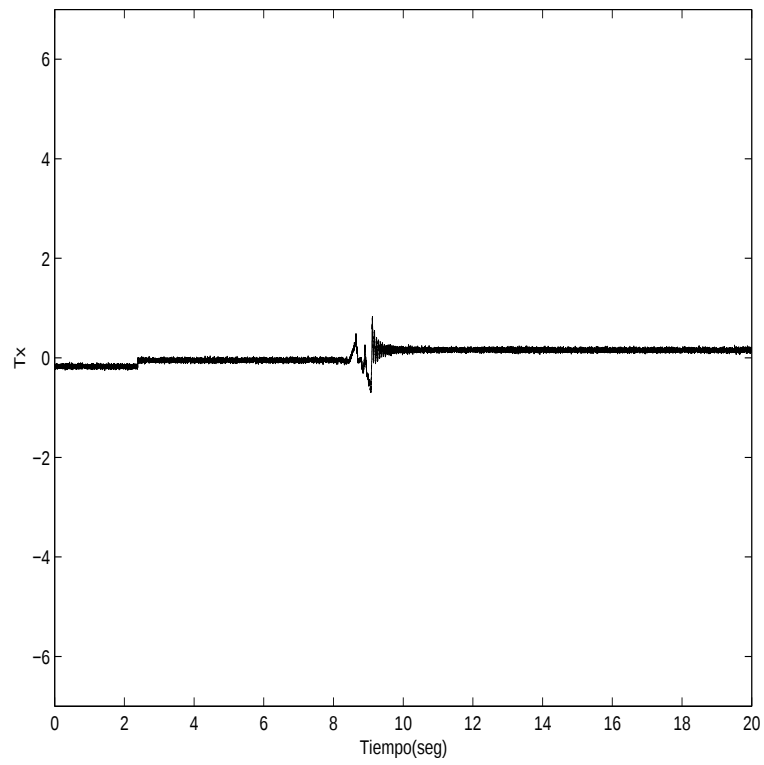


Figura 8.4: Torsión en el eje x

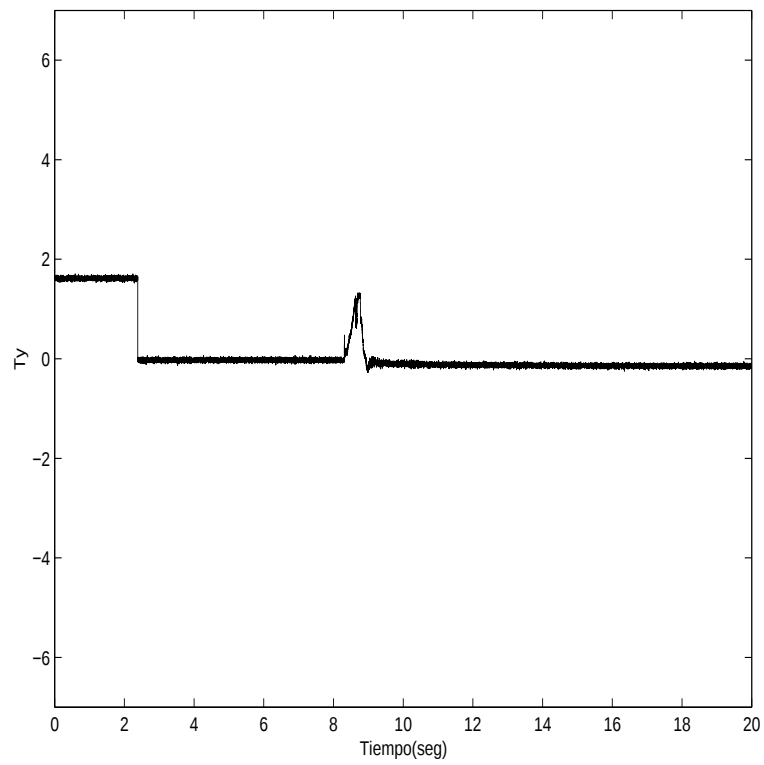


Figura 8.5: Torsión en el eje y

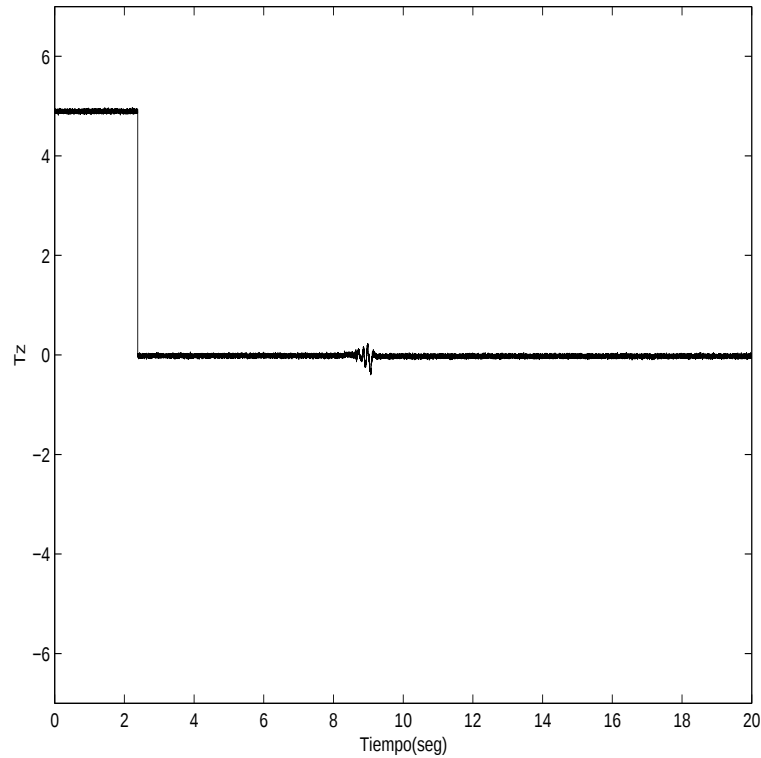


Figura 8.6: Torsión en el eje z

8.3 Comprobación de escala en F_z

La segunda prueba llevada a cabo tenía como finalidad comprobar la correcta medición de fuerzas en el eje $\mathbf{z}$ haciendo uso de dos pesas de $5Kg$. Para ello se partiría de una posición de reposo del sensor en la que la opción de Bias se encontrara inicialmente activa (fuerzas y torsiones nulas), y tras un cierto intervalo de tiempo se actuaría ejerciendo una fuerza sobre el eje $\mathbf{z}$ del transductor colocando una pesa de $5Kg$ sobre su base. Tras cierto intervalo de tiempo se volvería a actuar sobre el mismo eje colocando otra pesa de características similares encima. La finalidad de esta prueba era comprobar en primer lugar que los únicos cambios apreciables en las mediciones se efectuaban en el eje $\mathbf{z}$, y en segundo lugar, comprobar que los incrementos de fuerza que se apreciaban eran similares debido a que las características de las pesas también lo eran.

A continuación se muestra una serie de gráficas con los resultados experimentales obtenidos. Cada gráfica corresponde a fuerzas y torsiones en cada uno de los ejes de coordenadas.

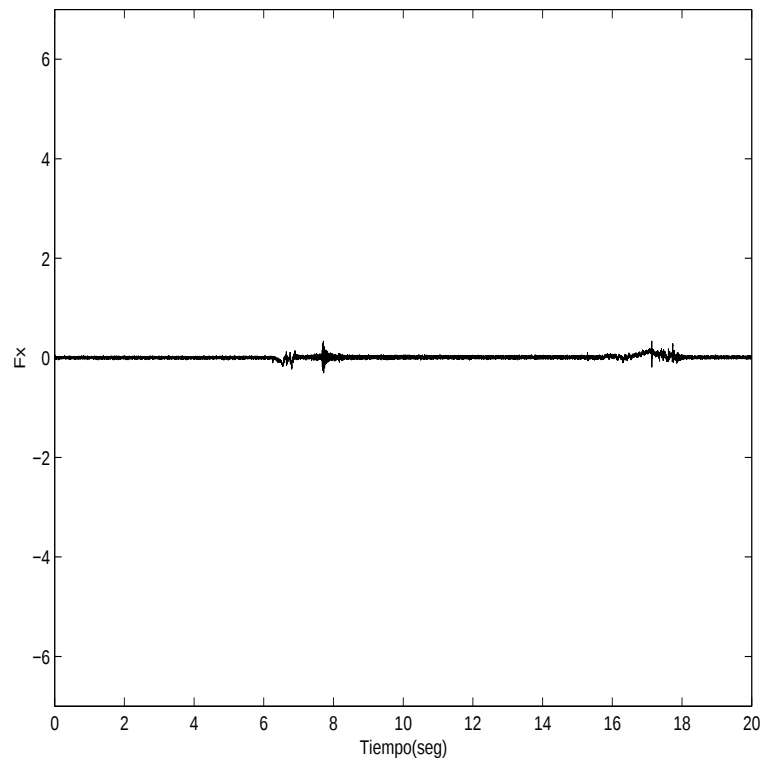


Figura 8.7: Fuerza en el eje x

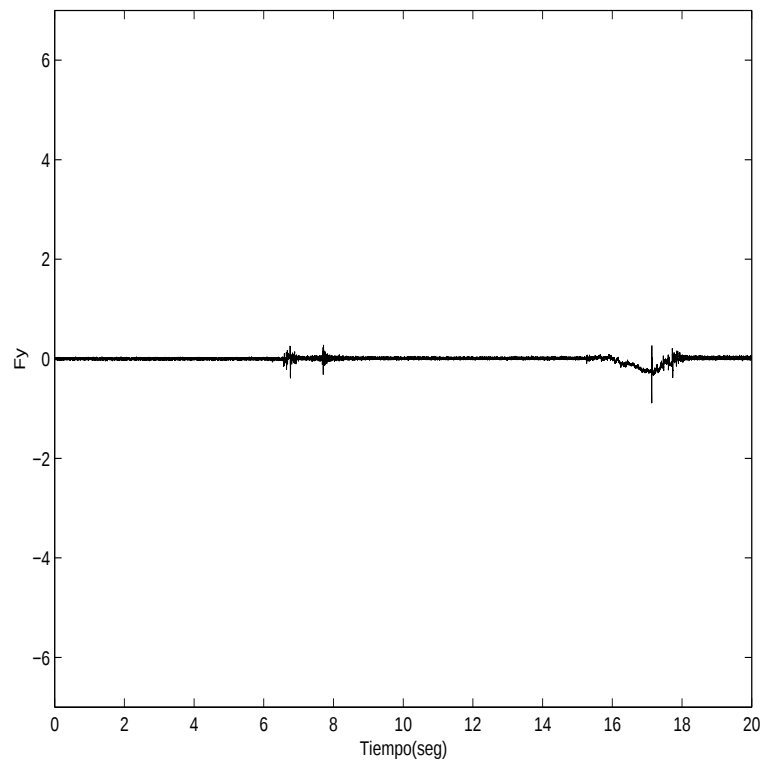


Figura 8.8: Fuerza en el eje y

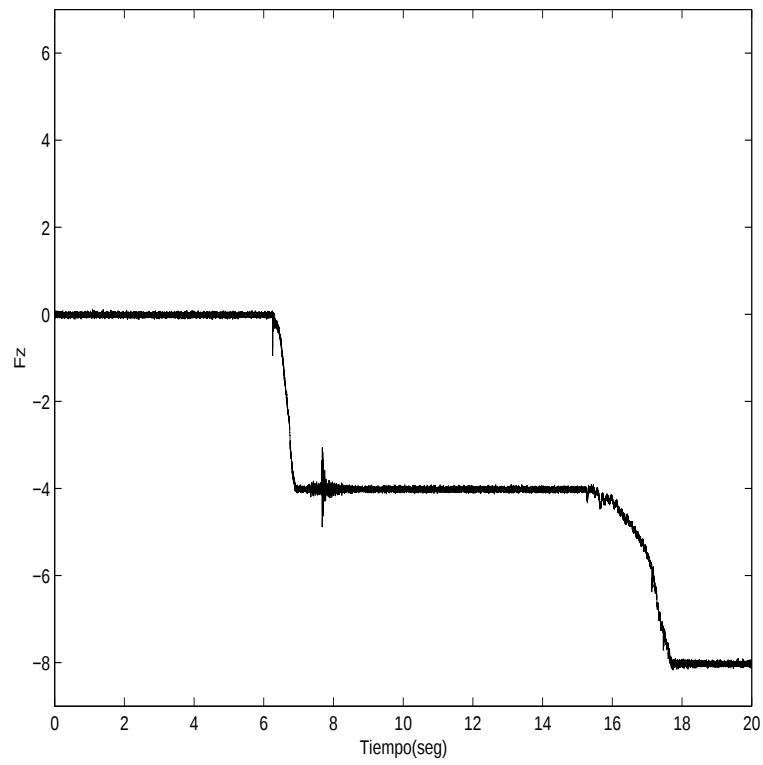


Figura 8.9: Fuerza en el eje z

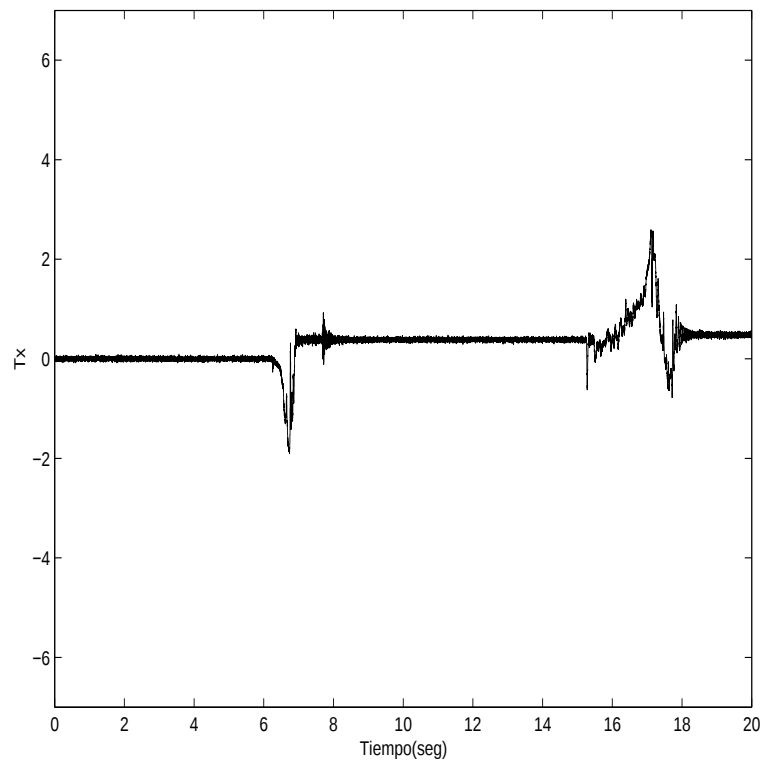


Figura 8.10: Torsión en el eje x

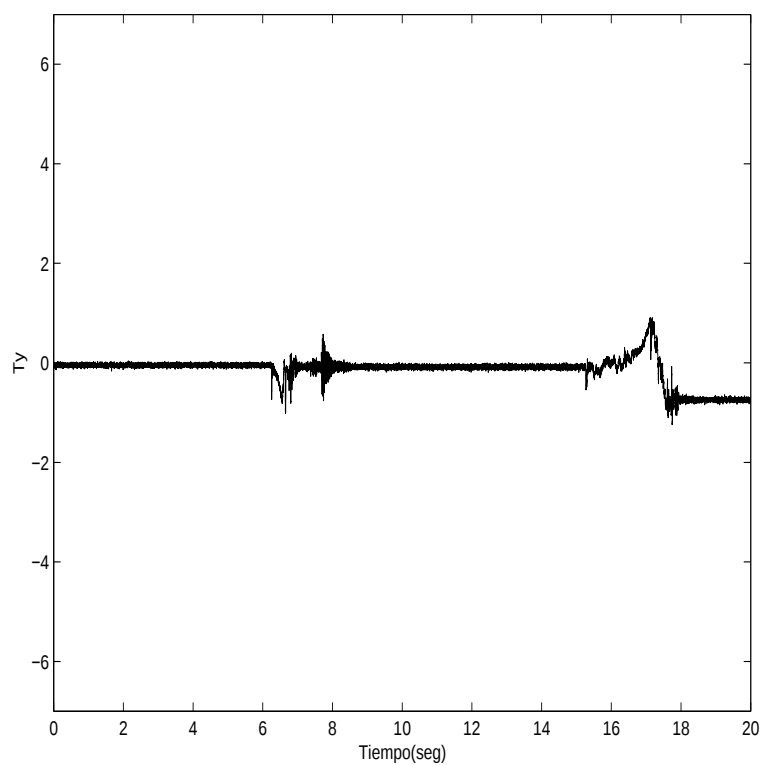


Figura 8.11: Torsión en el eje y

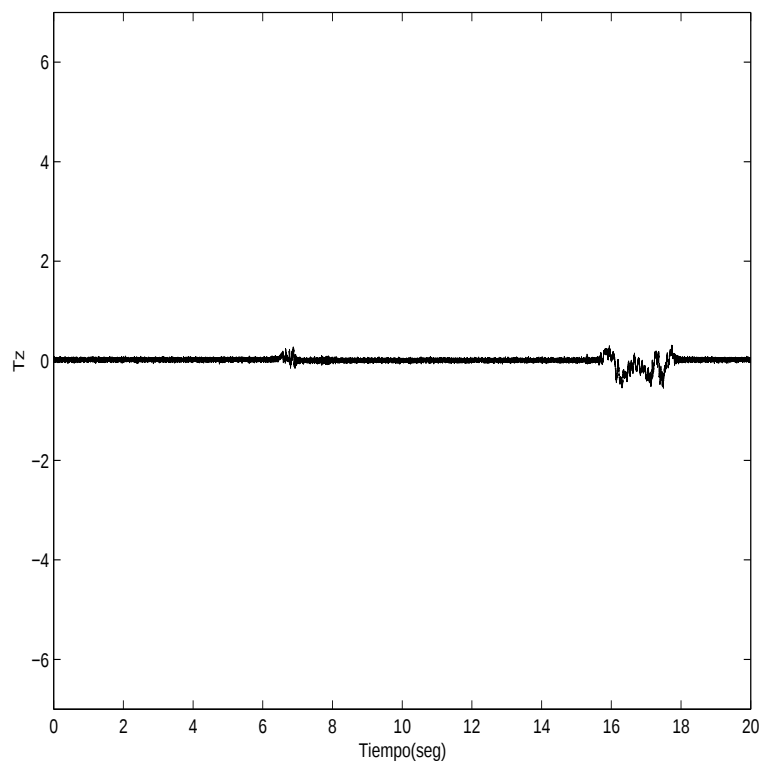


Figura 8.12: Torsión en el eje z

Capítulo 9

Conclusiones y desarrollos futuros.

Como resultado final tras la elaboración del proyecto se ha conseguido integrar la medición de fuerzas y torsiones que se aplican sobre el efector final del manipulador RM-10 en el entorno de control desarrollado para dicho manipulador. Asimismo se ha incluido un conjunto de medidas de seguridad basadas en aislamiento galvánico que permitirán proteger todos los equipos utilizados para la realización del proyecto.

En definitiva, se ha modificado la plataforma de control existente para el robot RM-10, de forma que ahora se podrá disponer, como datos para una posible realimentación, de las fuerzas y torsiones que se aplicarán sobre un transductor acoplado al efector final del manipulador.

Gracias al desarrollo del presente proyecto se abre un amplio campo de posibilidades y extensas líneas de desarrollos futuros. El hecho de permitir el conocimiento en tiempo real de las fuerzas y torsiones aplicadas sobre el sistema es tan sólo el primer paso necesario para poder desarrollar cualquier técnica de control mediante realimentación de esfuerzos. No obstante, se propone estudiar la posibilidad de incluir técnicas de calibración automática del sensor haciendo uso de un acelerómetro ([14], [15] y [16]). Dichas técnicas permitirán estimar las fuerzas inerciales que aparecen debida a la propia inercia del sistema y que son medidas por el sensor. Idealmente, estas fuerzas no deberían tenerse en cuenta para implementar técnicas de control y por tanto, convendría restárselas a las medidas proporcionadas por el sensor. Por este motivo se aconseja que para cualquier desarrollo futuro ésta sea la primera aplicación a desarrollar.

Finalmente cabe destacar el esfuerzo llevado a cabo para conseguir disponer de las medidas de fuerzas y torsiones con un tiempo de muestreo inferior al milisegundo. A pesar de que inicialmente se obtuvieron excelentes resultados mediante una comunicación serie con el controlador del sensor, se estudió todo un conjunto de posibilidades

para intentar implementar un protocolo que proporcionase un tiempo de muestreo más bajo y que aprovecharse hasta el límite las prestaciones del transductor. Como resultado de este estudio, se decidió desarrollar el protocolo de comunicación analógico que es mucho más adecuado para técnicas de control por realimentación de esfuerzos.

Bibliografía

- [1] Rafael Romero Vicente. *Plataforma de Experimentación de Controladores para Robots Industriales*. Escuela Superior de Ingenieros, Universidad de Sevilla, 2001.
- [2] Stefano Chiaverini, Bruno Siliciano, Luigi Villani. A Survey of Robot Interaction Control Schemes with Experimental Comparison. *IEEE/ASME Transactions on Mechatronics*, 4, 1999.
- [3] Richard Volpe and Pradeep Khosla. A Theoretical and Experimental Investigation of Explicit Force Control Strategies for Manipulators. *IEEE/ASME Transactions on Mechatronics*, 38, 1993.
- [4] Neville Hogan. Impedance Control: AN Approach to Manipulation. *Journal of Dynamic Systems Measurements and Control*, 107:1–24, 1985.
- [5] *Mathworks. Matlab Release 11.1 Online Manuals.*, 1998.
- [6] *dSPACE. CLIB, C-Interface Library.*, 1999.
- [7] *dSPACE. ControlDesk, Experiment Guide.*, 1999.
- [8] *dSPACE. ControlDesk Reference.*, 1999.
- [9] *dSPACE. DS1103 PPC Controller Board, RTI Reference.*, 1999.
- [10] *dSPACE. MLIB/MTRACE, MATLAB-dSPACE Interface Libraries.*, 1999.
- [11] *dSPACE. Real-Time Interface (RTI and RTI-MP), Implementation Guide.*, 1999.
- [12] Assurance Technologies, Inc. *Installation and Operations Manual for F/T*, 1991.
- [13] Assurance Technologies, Inc. *Direct Interfacing to Transducer with Multiplexer Board*, 2000.
- [14] J. Gámez García , A. Robertsson , J. Gómez Ortega and R. Johansson. Automatic Calibration Procedure for a Robotic Manipulator Force Observer. *International Conference on Robotics and Automation*, 2005.

- [15] J. Gámez García , A. Robertsson , J. Gómez Ortega and R. Johansson. Force and Acceleration Sensor Fusion for Compliant Robot Motion Control. *International Conference on Robotics and Automation*, 2005.
- [16] J. Gámez García , A. Robertsson , J. Gómez Ortega and R. Johansson. Sensor Fusion of Force and Acceleration for Robot Force Control. *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.