

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



INGENIERÍA INDUSTRIAL: ELECTRICIDAD
PROYECTO FIN DE CARRERA:

**DISEÑO DE LA ARQUITECTURA HARDWARE DE UN
AEROGENERADOR DE VELOCIDAD VARIABLE DFIG**

Autor: RICARDO ALBARRACÍN SÁNCHEZ

Directores: HORTENSIA AMARÍS DUARTE

CARLOS ÁLVAREZ ORTEGA

Leganés, Octubre 2008



ÍNDICE

1.	INTRODUCCIÓN	3
1.1.	Situación actual	3
1.2.	Antecedentes.....	6
1.3.	Objetivo del Proyecto	11
2.	DESCRIPCIÓN DEL PROBLEMA A RESOLVER	12
2.1.	Revisión de las arquitecturas Hardware para el control de aerogeneradores ...	14
2.1.1.	Propuesta de resolución del problema	14
2.1.2.	Descripción del sistema propuesto	15
2.1.3.	Sistema completo	17
2.1.4.	“Bloque Convertidor”	17
2.1.4.1.	“Sistema de Electrónica de Potencia” del “CONDOR Converter”	17
2.1.4.2.	“Sistema de Electrónica de Control ” del “CONDOR Converter”	17
2.1.5.	“Bloque Turbina”	18
2.1.5.1.	“Sistema de Electrónica de Potencia de la Turbina”	18
2.1.5.2.	“Sistema de Electrónica de Control de la Turbina”	18
2.1.6.	Aspecto final del sistema completo diseñado.....	19
3.	DESCRIPCIÓN DEL PROTOTIPO	20
3.1.	BANCADA DE MOTORES	23
3.1.1.	MOTOR DC.....	23
3.1.2.	MOTOR AC.....	24
4.	xPC TARGET.	25
5.	ESPECIFICACIONES DEL SISTEMA DE CONTROL	30
5.1.	SEÑALES DE ENTRADA	30
5.2.	SEÑALES DE SALIDA	30
6.	DESCRIPCIÓN DE LA ARQUITECTURA DE COMUNICACIONES	32
6.1.	RESULTADOS.....	33
6.1.1.	HARDWARE DE CONEXIÓN ENTRE PC <i>HOST</i> -PC-104 CON CABLE RS232	33
6.1.1.1.	Configurando el PC Host.....	36
6.1.1.2.	Comenzando la comunicación. Crear un modelo en Simulink para ejecutar con xPC Target.....	37
6.2.	HARDWARE DE CONEXIÓN ENTRE PC <i>HOST</i> -PC <i>TARGET</i>	41
6.2.1.	CONEXIÓN <i>HOST-TARGET</i> . Mediante cable RJ45 cruzado	41



6.2.1.1. Configurando el PC Host.....	42
6.2.2. CONEXIÓN HOST-TARGET. Mediante cable RJ45 y Router	45
6.3.1. CONEXIÓN PC HOST- VARIOS PC TARGETs.....	47
7. ADQUISICIÓN DE SEÑALES.....	50
7.1. ENVÍO DE DATOS MEDIANTE CABLE RS 232.....	50
7.2. ENVÍO DE DATOS A TRAVÉS DE TARJETAS DE ADQUISICIÓN DE DATOS. DESDE LA TARJETA NI SCB-68 A NI SCB-100	51
7.2.1. Envío de un número decimal.....	51
7.2.2. Envío de un número decimal desde la tarjeta SCB-68 a la tarjeta SCB-100 de NI.....	53
7.2.3. Envío de una señal sinusoidal.....	56
7.2.3.1. Dificultades durante la implementación	58
7.2.3.2. Obtención de las gráficas en el osciloscopio.....	60
7.2.3.3. Esquemas de conexión en las tarjetas SCB de NI. Cableado, cable de 8 hilos.....	62
7.2.3.4. Retardos en el envío. xPC Target, SCB-68 y SCB-100 en el mismo PC Target.....	66
7.2.3.5. Retardos en el envío. xPC Target, generador de funciones SCB-100....	68
7.2.3.6. ENVÍO DE DATOS POR EL PROTOCOLO UDP A TRAVÉS DE CABLE DE RED RJ45.....	71
8. PROGRAMACIÓN DE LA ARQUITECTURA DE CONTROL. ENVÍO DE LA PWM DE CONTROL A TRAVÉS DE LA NI	74
9. SELECCIÓN DE LA ARQUITECTURA HARDWARE DEL CONTROL DEL MOTOR DC...	76
9.1. Ejemplo de conexión de la máquina DC a través de la xPC Target	76
10. CONCLUSIONES.....	80
11. BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN	83



1. INTRODUCCIÓN



1. INTRODUCCIÓN

1.1. Situación actual

La energía eléctrica ha sido el impulsor del desarrollo industrial mundial desde el siglo XIX hasta nuestros días. Esta energía ha sido principalmente producida por la combustión de combustibles fósiles. Sin embargo, este tipo de generación es contaminante, cara y limitada. Por todo esto es necesario encontrar y desarrollar fuentes alternativas como las energías renovables [1].

España está teniendo un amplio crecimiento de la intensidad energética en los últimos 30 años. La creciente y excesiva dependencia energética exterior (cercano al 80% en los últimos años) y la necesidad de preservar el medio ambiente, obligan al fomento de acciones eficaces para un uso eficiente de la energía y la utilización de fuentes limpias. Por lo que es de vital importancia el crecimiento de las fuentes renovables, junto a una importante mejora de la eficiencia energética que respondan a estrategias económicas, sociales y medioambientales viables [2].

España tiene un gran potencial para producir energía eléctrica a partir de Fuentes de Energía Renovables (RES, *Renewable Energy Sources*) como eólica, fotovoltaica, hidroeléctrica y solar térmica. El “Plan de Energías Renovables 2005-2010” [2] fija como objetivo que las RES, en 2010, cubran como mínimo el 12% de la demanda total de la energía primaria con una estimación de potencia instalada de 20155 MW para la energía eólica.

La energía eólica es un tipo de energía renovable que ha sido reconocida medioambientalmente positiva y económicamente competitiva [3], especialmente en lugares con buen promedio de viento (> 8.5 m/s velocidad de viento anual).

En el marco Español, la potencia eólica instalada al final del 2007 era de 15.145 MW, produciéndose en dicho año un crecimiento del 30 % con respecto a la potencia instalada en el 2006. En el 2007 se instalaron 3.514,8 MW, lo que supuso superar el récord alcanzado en 2004 cuando se instalaron 2.361 MW. Además el número de parques ascendió a 672 frente a los 538 a principios de año [4].

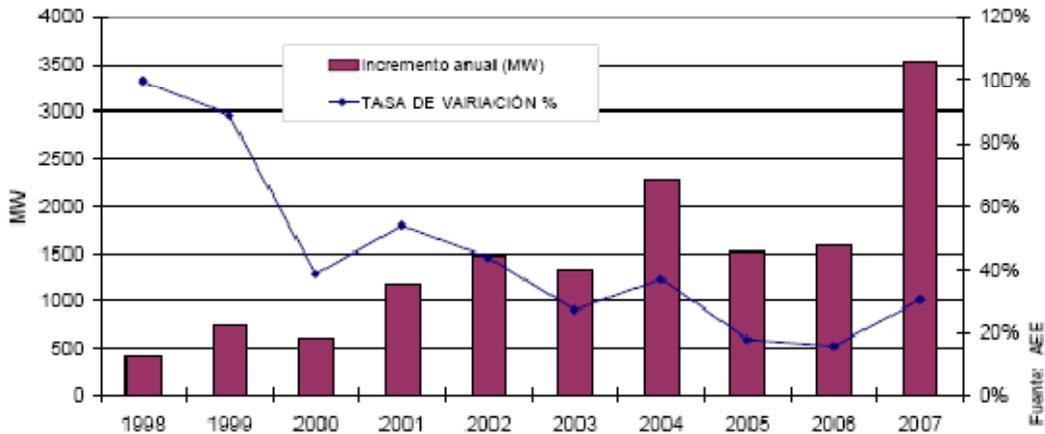


Fig. 1. Incremento anual de la potencia eólica instalada y tasa de variación

Este fuerte crecimiento hace alcanzar los objetivos anuales marcados en [2] y cuyo objetivo final es el alcanzar los 21.155 MW en el año 2010. Vemos los objetivos marcados en [2]:

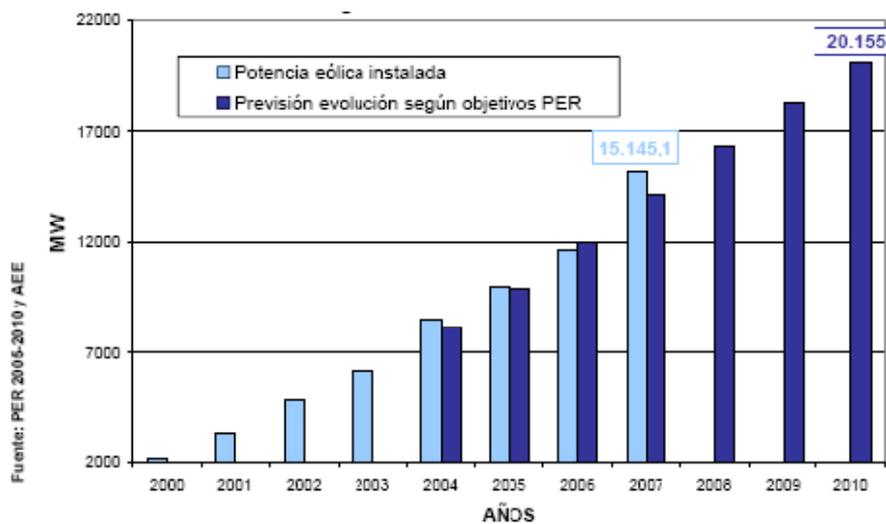


Fig. 2. Evolución anual de la potencia eólica instalada y previsión según el Plan de Energías Renovables 2005-2010

En cuanto a comunidades autónomas, es Castilla la Mancha la comunidad con más potencia instalada con un total de 3.131,36 MW instalados y con un crecimiento de 37,25 % [4].

En cuanto a términos porcentuales, es Andalucía la comunidad que más crecimiento en potencia instalada ha experimentado con un 140,65 %, llegando a un total de 1.459,71 MW instalados [4].

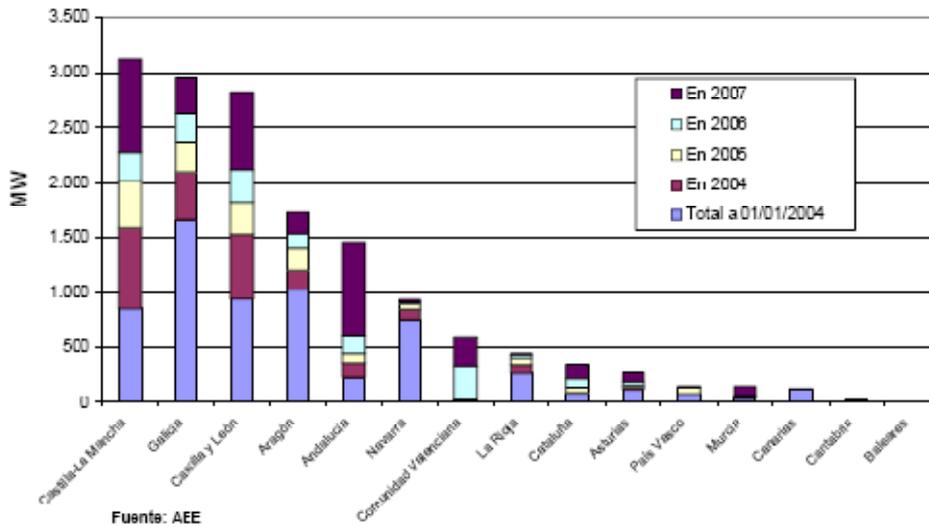


Fig. 3. Potencia eólica instalada por comunidades

En cuanto a la generación eólica en 2007 se alcanzaron los 26.407 GWh, lo que supone un 10 % de la demanda eléctrica. Esta cifra convierte a la energía eólica en la cuarta tecnología por su contribución al sistema por detrás de las térmicas de carbón, las térmicas de gas de ciclo combinado y de la energía nuclear [5].

En cuanto a las instalaciones de aerogeneradores, el 85% de nueva instalación de los aerogeneradores utilizan un Generador Asíncrono Doblemente Alimentado (DFIG). Por ello el presente proyecto se centra en el estudio de este tipo de aerogeneradores.

1.2. Antecedentes

La energía eólica debe su nombre al dios griego del viento Eolo. Sus aspas evocan casi de inmediato a los molinos de viento de Don Quijote de la Mancha. Pero su origen es más antiguo. David Rittenhouse, en su libro "La energía eólica", cuenta que el primer dibujo alusivo a molinos de viento datan de hace 3,000 años antes de Cristo. Se trata de una cerámica china. "Hay referencias a un molino de viento en Babilonia en el siglo III a.C. Al parecer se usaron corrientemente en Persia en el siglo VII de nuestra era y se difundieron por Europa Occidental e Inglaterra hacia el siglo XII y por China en el siglo XIII", plantea.

Pero las primeras turbinas eólicas generadoras de electricidad son atribuidas a Poul la Cour, en 1891.

Hoy en día se estima que la capacidad mundial de generación de energía eólica se ha multiplicado 10 veces en la última década. Robert Thresher, director del estadounidense Centro Nacional de Tecnología Eólica del Departamento de Recursos Energéticos de Estados Unidos, calcula que de 3,5 gigavatios (un gigavatio equivale a 1.000 millones de vatios) en 1994 se ha pasado a 50 gigavatios hacia fines de 2004 [6].

Posteriormente, en el siglo XIV se desarrolló el molino de torre, en donde sólo gira la parte superior del molino, mientras que la parte inferior es una torre de ladrillo o piedra solidaria al suelo.

Con la revolución industrial del siglo XIX y tras la aparición de los motores térmicos, las máquinas eólicas quedaron apartadas a las regiones rurales. De esta época es la turbina eólica multipala americana, que consta de un rotor multipala de unos tres metros de diámetro, de rotor horizontal y orientación con veleta. Además fue este ingenio el que marcó la sustitución de la madera por el hierro en la construcción de sus álabes a partir de 1891 [7].

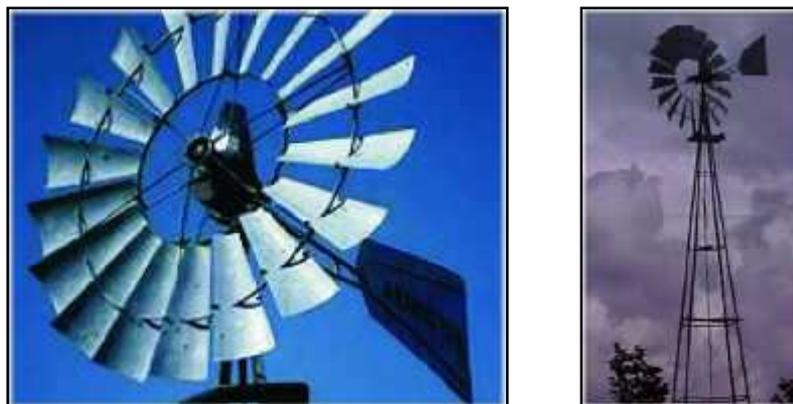


Fig. 4. Turbinas eólicas multipala americanas

En 1892, en Dinamarca el profesor Latour diseñó el primer aerogenerador eléctrico marcando el comienzo de la tecnología eólica moderna.

Como consecuencia de la aplicación de la ingeniería aeronáutica y los avances en Mecánica de Fluidos a comienzos del siglo XX, aparecieron aeroturbinas (máquinas rotatorias en las que la energía del viento es transformada en otra forma de energía) de alto rendimiento y alta velocidad. En esa misma época Betz demostró que la máxima fracción de energía extraíble de una corriente uniforme abierta es del 60% y Glauert demostró que se puede obtener mayor rendimiento cuanto mayor es el coeficiente de velocidad λ , cociente entre la velocidad de punta debida al giro y la velocidad incidente del viento “tip speed ratio”.

En 1927 Darrieus desarrolla una turbina de eje vertical. Consiste en aspas verticales con perfil aerodinámico. El inconveniente es la limitación de altura y con ello de la potencia nominal que tiene.

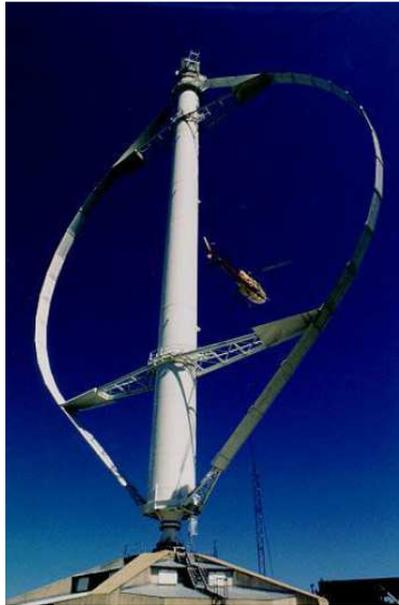


Fig. 5. Turbina eólica tipo Darrieus de eje vertical

Las turbinas eólicas de eje vertical son aquellas cuyo rotor está situado verticalmente. En el eje de la turbina.

La principal ventaja de las turbinas de eje vertical es que el generador y la caja multiplicadora están en el suelo y son muy accesibles, y que no necesitan mecanismos de orientación. Entre las desventajas está la ligera inferior eficiencia y el hecho de que no auto arranquen, así como la necesidad de desmantelamiento total para reemplazar el eje.

Las de eje horizontal necesitan alinearse con la dirección del viento, de forma que el viento sople paralelo al eje de rotación.

Después de la segunda guerra mundial y como consecuencia del bajo precio del petróleo se ralentizó el desarrollo de la energía eólica. Destacar la política de Dinamarca que, después de evaluar los recursos del país, instala en 1957 un generador de 240 KW sirviendo de base para el desarrollo de las modernas turbinas eólicas [referencia]

En la década de los 80, el aerogenerador (sistema que convierte la energía cinética del viento en energía eléctrica) típico era de 200 kW y disponía de hélices de paso fijo, controlando la velocidad de giro de la turbina mediante la pérdida aerodinámica.

Posteriormente se desarrollaron las turbinas de paso variable, que permite variar el ángulo de ataque de la pala respecto al viento, mejorando y disminuyendo la sustentación.

Durante un tiempo convivieron en el mercado ambas tecnologías, imponiéndose finalmente la segunda, ya que la primera tiene el techo de potencia en 1 MW.

En la actualidad los aerogeneradores disponen de una potencia comprendida entre 1 y 2 MW y su coste de instalación estimable es de 950 \$/kW a 1300 \$/kW según el tipo y el tamaño de la máquina frente a los 3000 \$/kW a principio de la década de los 80.

En cuanto a la eficiencia, los aerogeneradores actuales son tan eficientes que entre el cubo de la hélice y la salida eléctrica sólo pierden el 7 % de la energía que captan del viento. La eficiencia de las palas se sitúa en torno al 50 %, muy próximo al límite de Betz.



Fig. 6. Turbina eólica de eje horizontal de 5 MW de Repower Systems [9].

Las turbinas eólicas de eje horizontal son aquellas cuyo rotor es sensiblemente paralelo a la dirección del viento.

En este tipo de turbinas el viento, al hacer girar las palas del rotor, genera una energía cinética que se transmite, a través del eje principal, al alternador cobijado en la góndola. Se genera una corriente eléctrica que es transmitida mediante cables conductores a un centro de control donde se almacena en acumuladores o se distribuye a los centros de consumo o se evacua hacia la red de transporte de energía eléctrica.

Esta tecnología tiende al gigantismo. En las granjas “off shore” de Europa se ven molinos de 100 o 120 metros de altura, con hélices que barren lentamente círculos de 80 y hasta 100 metros de diámetro, y generan hasta 5 megavatios por unidad, aunque lo habitual son los 2,5 o 3 megavatios. Sin embargo, este desarrollo viene limitado por los materiales de construcción de los molinos. A mayor tamaño, más ligeros y robustos deben ser los materiales y en consecuencia, más caros [8].

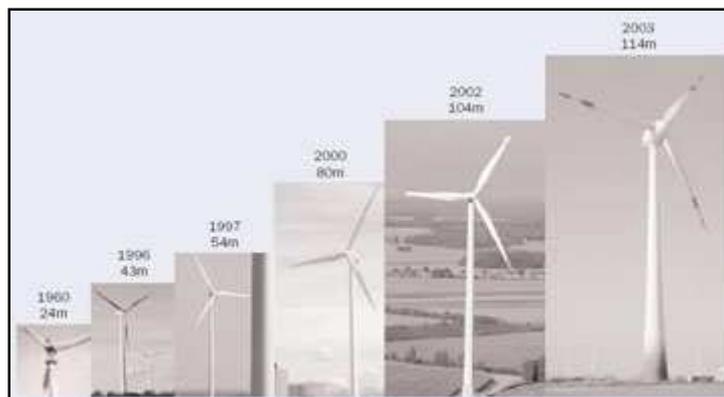


Fig. 7. Evolución del tamaño de los aerogeneradores

El futuro nos depara parques eólicos marinos a escasos kilómetros de las costas, donde se podrán instalar grandes turbinas eólicas (5MW) más ruidosas con un bajo impacto ambiental [8].



Fig. 8. Parque eólico marino. Aerogenerador V120

El V120 es un avanzado aerogenerador de alta mar. Diseñado especialmente para trabajar bajo las extremadamente duras condiciones marítimas y para cumplir con los requisitos de grandes proyectos en alta mar. El diseño flexible del V120 implica la posibilidad de aplicación de una gran variedad de métodos de instalación, de forma que el aerogenerador puede instalarse sobre diversos tipos de cimientos y trabajar bajo diferentes condiciones de red. Gracias a su potencia nominal de 4,5 MW, el número de aerogeneradores necesarios para obtener un rendimiento determinado se ha visto reducido de nuevo, así como el diseño de un “mantenimiento único anual” asegura que los costes por tiempo de parada, personal y mantenimiento se reducen al mínimo. Este es un factor muy importante cuando se habla de instalaciones en alta mar de difícil acceso.

El uso extensivo de tecnologías punteras, tales como el diseño de palas ligeras basado en carbono y madera, así como las nuevas y más fuertes aleaciones de acero, hacen que este tipo de aerogeneradores tengan un bajo peso que reduce el consumo de material, recorta los costes de producción y transporte, y permite construir instalaciones en alta mar en una sola alzada. Reduciendo también los elevados costes de las cimentaciones [10].



1.3. Objetivo del Proyecto

El presente proyecto tiene por objetivo el diseño y control de un prototipo de aerogenerador de velocidad variable doblemente alimentado en el laboratorio de la Universidad Carlos III utilizando una arquitectura hardware de bajo coste.

El hardware implementado debe facilitar el control en tiempo real del prototipo del aerogenerador. Por lo que los tiempos de ejecución y muestreo son críticos.

El prototipo de aerogenerador utilizado ha sido una bancada de motores que consta de un motor de corriente continua unido por un eje a un motor de corriente alterna.

Para el control del aerogenerador en tiempo real se utilizará la aplicación *xPC Target* de *The Mathworks* [11], [12], [13], [14], la cual permite trabajar en tiempo real.

Se utilizarán tarjetas de adquisición de datos de *National Instruments*, SCB-68 y SCB-100 [15] que permiten adquirir entradas y salidas: digitales y analógicas del sistema real y controlar los inversores del aerogenerador DFIG.

Durante el presente proyecto se tendrá en cuenta la posibilidad de la utilización de una tarjeta *DSP* del fabricante *Texas Instruments Spectrum Digital* de modelo comercial *eZDSP™ F2812* [16]. Esta tarjeta permite obtener cálculos en tiempo real comparables a los ofrecidos con la aplicación *xPC Target*.

El empleo de la aplicación *xPC Target* nos permite la reutilización de CPUs, ordenadores antiguos en desuso, para el cálculo de procesos en tiempo real. Permitiendo tener un coste relativamente bajo mediante una implementación común de hardware innovador para la conexión y comunicación entre un sistema real que se pueda controlar en tiempo real.



2. DESCRIPCIÓN DEL PROBLEMA A RESOLVER



2. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Para la realización del presente proyecto fin de carrera ha sido necesario la utilización del conocimiento de diversas asignaturas del plan de estudios de la titulación de Ingeniería Industrial especialidad Electricidad, a saber: Accionamientos eléctricos, Máquinas eléctricas, Análisis dinámico de sistemas eléctricos, Señales y sistemas, Ingeniería energética, Dirección y sistemas de información, Ingeniería de control y Electrónica industrial. Sin embargo, ha sido necesario ampliar el conocimiento profundizando en temas no cubiertos, en algunos casos, en ninguna de las asignaturas de la titulación, concretamente:

- Tecnologías de comunicaciones (hardware/software).
- Medida y adquisición de señales eléctricas.
- Programación software para el control en tiempo real.
- Programación y control de tarjetas de comunicaciones.
- Control discreto de sistemas reales.

Para el aprendizaje de los conocimientos arriba expuestos fue de vital importancia la consulta de trabajos previos sobre estudios de aplicaciones en tiempo real [17], [18], [19]. Así como, la formación de la idea principal de aplicar estos estudios en tiempo real al prototipo de un aerogenerador.

Para aprender a utilizar la aplicación xPC Target se basó el estudio, en un primer momento, en la utilización de un manual desarrollado por la Universidad de Chicago [20] ampliado posteriormente por el trabajo de investigación llevado a cabo en el laboratorio. Esta aplicación se utilizaría para realizar la arquitectura de comunicación del prototipo de un aerogenerador de velocidad variable.

En la actualidad muchas arquitecturas empleadas para el control de convertidores electrónicos en aerogeneradores se basan en realizar el diseño apropiado del sistema electrónico de control y potencia de una turbina eólica de velocidad y paso variables.

Las arquitecturas de comunicaciones y control convencionales (*D-space* [21], *DSP* [16], etc.) son costosas pero sencillas en su programación y control del sistema real o asequibles económicamente pero complicadas a la hora de escribir los programas. Además son rígidas y poco actualizables.

Estas arquitecturas o sistemas disponen de un procesador, donde se vuelca el programa ejecutable que controla el aerogenerador, y de tarjetas I/O (entrada/salida) para el control en tiempo real y un software de programación. En muchas ocasiones el programa ya tiene implementado el control PWM de los convertidores electrónicos.



En el presente estudio se tuvo que implementar el control PWM [7].

	<i>D-space</i> 1104 [21]	PC de bajas prestaciones con tarjeta IO de NI 6602
Características procesador	MPC8240 250 MHz	Pentium III 1000 MHz
Señales entrada/salida	16 analógicas 1 digital	32 digitales
Señales PWM	4 salidas PWM	8 salidas PWM
Coste	4300€	950€

Tabla 1. Precios. *D-space* 1104 [21] frente a la propuesta hardware de este proyecto

Un inconveniente que ofrecen algunas soluciones comerciales de tipo *Plug-and-Play* (conectar y funcionar) es que son poco flexibles, demasiado cerradas y no permiten que el usuario pueda implementar sus propias estrategias de control o tenga un control interno del proceso.

Otras opciones, no explotadas comercialmente hasta la fecha para el control de aerogeneradores, son las que se basan en el control de comunicaciones entre un *PC Host* (servidor), y un *PC Target* (cliente). En este esquema, es suficiente utilizar como procesador un ordenador básico de bajas prestaciones.

xPC Target es una solución para realizar e implementar de manera rápida un prototipo, desarrollarlo y ejecutar aplicaciones en tiempo real utilizando ordenadores convencionales. Utiliza el software *Matlab Simulink* [11], [12], [13], [14] y la toolbox *xPC-Target* [13]. Se basa en la comunicación *Target-Host* (cliente-servidor) [20] a través de una conexión *Ethernet*.

En el *PC Host* (servidor) se diseñan los programas mediante *Matlab Simulink*, utilizando la programación estándar. No se requieren conocimientos específicos de lenguajes de programación. Posteriormente se compilan automáticamente utilizando un compilador de lenguaje C/C++. El programa ejecutable se vuelca en el *xPC Target* (cliente) mediante un cable serie RS-232 o un cable de red RJ-45 mediante el protocolo UDP o TCP/IP (TCP/IP será el protocolo utilizado por ser el más eficaz frente al UDP. Y la conexión RJ-45 frente a la serie).

En el *PC Target* (cliente) corren los programas ejecutables que controlan en tiempo real el prototipo. Éste puede ser activado para trabajar en conexión con el *PC Host* utilizando el *External mode* (modo externo) dentro de *Simulink* en el *PC Host* o activado para trabajar de forma autónoma mediante un comando en la ventana de comandos de *Matlab* del *PC Host*.



2.1. Revisión de las arquitecturas Hardware para el control de aerogeneradores

En el proyecto [21] se encuentra inmerso en el campo de la investigación sobre la técnica de los convertidores CC/CA de tres niveles aplicada en el sector de la aerogeneración, es decir, en la obtención de energía eléctrica a partir de energía eólica.

El principal objetivo es realizar un diseño apropiado del sistema electrónico de control y potencia de una turbina eólica de velocidad y paso variables de 250kVA. Para ello, se diseñan e implementan dos tarjetas electrónicas:

- Una tarjeta basada en una FPGA, cuyo objetivo es el de controlar, junto con un procesador digital de señal (DSP), el convertidor NPC de tres niveles de un aerogenerador, y a la vez, realizar la comunicación con la electrónica de control de la turbina del aerogenerador y la interacción con un PC.
- Una segunda tarjeta, que junto con otro DSP, estará destinada a realizar todo el control de la turbina del aerogenerador.

Para simular todo el sistema de aerogeneración con resultados experimentales, se montará una bancada de motores, que constará de un motor DC y un motor AC. El motor AC es el generador, mientras que el motor DC es el encargado de arrastrar al aerogenerador. Ambos motores están acoplados a un eje común, de forma que el motor DC se comporta como un simulador de viento.

El motor DC estará excitado por un ‘driver’ comercial, el cual se puede programar de forma remota. Toda esta tarea se ha realizado sobre un convertidor “back-to-back” NPC de tres niveles comercial, existente en el laboratorio del Departamento de Electrónica de la Universidad de Alcalá.

2.1.1. Propuesta de resolución del problema

Este proyecto está enfocado a realizar el sistema electrónico y potencia de una turbina eólica de velocidad y paso variables de 250kVA. El objetivo de mayor importancia es encontrar una posible solución para desarrollar de manera eficiente la implementación experimental de un conjunto emulador de un sistema de aerogeneración y etapas de conversión de energía para su conexión a red, con la flexibilidad suficiente para que permita modificar una serie de parámetros de funcionamiento dando lugar a un amplio abanico de pruebas experimentales.

- El sistema que se ha propuesto para tal fin, consta fundamentalmente de cuatro elementos, que son:
 - un convertidor “back_to_back” NPC de tres niveles usado en turbinas eólicas



- la tarjeta de control destinada a controlar dicho convertidor
 - la tarjeta de control de la turbina del aerogenerador y
 - la propia turbina, que se simulará mediante un motor AC y que, deberá comportarse como una turbina eólica de velocidad y paso variables de 250kVA.
- Los procesadores utilizados a la hora de realizar el control del sistema son:
 - un DSP y una FPGA para el control del convertidor y
 - otro DSP para el control de la turbina.

Entre los diferentes procesadores digitales existentes para la realización de diseños de alto nivel, como el que aquí se persigue, bajo coste y controladores en tiempo real que trabajen de forma independiente, destacan los DSPs (*“Digital Signal Processors”*) y las FPGAs (*“Field Programmable Gate Arrays”*). Las principales ventajas que presentan los DSPs es que son plataformas de diseño de alto nivel y sobre todo se caracterizan por su implementación software, la cual permite la realización de diseños flexibles y muy fáciles de usar y actualizar, puesto que existen puentes con lenguaje C, C++ y Matlab. En cuanto a las FPGAs cabe destacar que presentan un mayor rendimiento que los habituales microprocesadores o incluso que los DSPs. No obstante, un diseño basado simplemente en FPGAs carece normalmente de flexibilidad, en cuanto a la implementación software se refiere.

Los controladores en tiempo real basados en FPGAs y DSPs presentan las ventajas de ambos procesadores. Por ello el “Sistema de electrónica de Control” del “CONDOR Converter”, utilizado en este proyecto, está basado en dos procesadores, la FPGA “SPARTAN 3E” de Xilinx (<http://www.xilinx.com>) y el DSP “TMS320C6713” de Texas Instruments (<http://www.ti.com>), que se encuentra en la tarjeta comercial “TMS320C6713DSK” de Spectrum Digital (<http://www.spectrumdigital.com>).

Por otro lado, el control de la turbina se realizará por medio de otro DSP, el “TMS320F2812” de Texas Instruments, que se encuentra en la tarjeta comercial “eZdsp™F2812” también de Spectrum Digital.

2.1.2. Descripción del sistema propuesto

El objetivo que se busca con la creación del sistema completo, ha sido el de desarrollar un prototipo experimental de un grupo aerogenerador, incluyendo todas sus partes, desde la generación hasta su conexión a la red eléctrica, intentando emular con el mayor realismo posible todas las posibilidades y reacciones de funcionamiento.

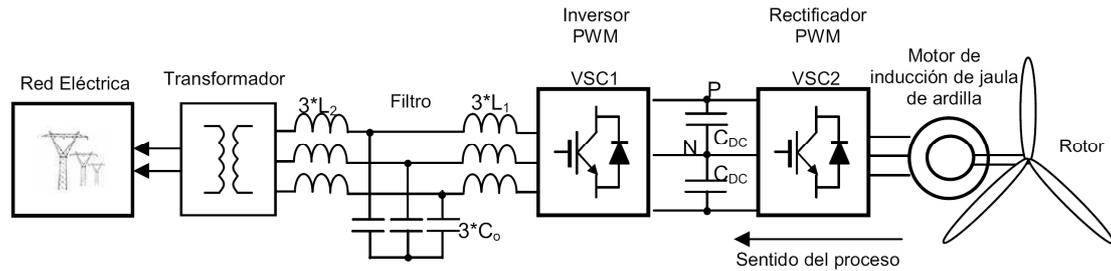


Fig. 9. Esquema de un generador eólico conectado a la red eléctrica [21]

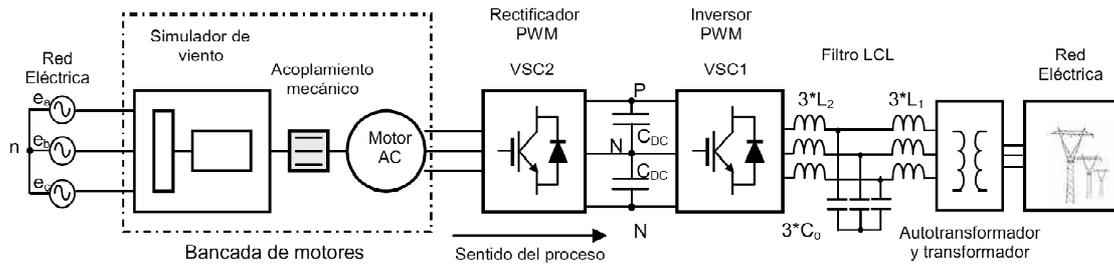


Fig. 10. Esquema de simulador de viento conectado a la red eléctrica [21]

En primer lugar se encuentra la parte generadora. En el sistema real del generador eólico conectado a red, esa parte generadora viene determinada por tres componentes bien diferenciadas, el viento que impulsa las palas, las palas y el rotor del aerogenerador. A la hora de hacer la simulación, habrá que tener en cuenta el funcionamiento real del sistema, es decir, la variación constante que se produce en la dirección y velocidad del viento, ya que la dirección afecta directamente a la velocidad de giro de las palas, y éstas deben adaptar su disposición física constantemente. Para llevar a cabo esto, se utiliza un equipo que simula la presencia de viento a partir de un motor DC y un convertidor DC basado en un “Puente en H” con tiristores.

Este equipo permite llevar a cabo la programación desde un PC de una curva de funcionamiento mediante la cual es posible simular razonablemente una variación de viento. Para la transmisión del movimiento creado a la segunda etapa del sistema, se utiliza un acoplamiento mecánico, de forma que se transmitirá un giro a velocidad o par constante según se haya indicado en la programación.

En segundo lugar, se tiene un segundo motor, este de AC, que será el encargado de simular el funcionamiento de una turbina. Se puede exponer como hecho puntual, que al existir siempre un motor al que se arrastra, es decir el de AC, se va a añadir una importante inercia al sistema, detalle que se puede interpretar como el arrastre que producen los trenes de engranajes y las palas existentes en los aerogeneradores.

Posteriormente se encuentra un convertidor “back-to-back” NPC de tres niveles, el cual se detalla a continuación. Antes de entregar la energía a la red debe estar presente la correspondiente etapa de filtrado, que normalmente es un filtro LCL y una etapa de adaptación y aislamiento, compuesta por un autotransformador y un transformador. Con el transformador se consigue un aislamiento galvánico entre el



convertidor y la red eléctrica, mientras que el autotransformador permite simular reducciones de la red eléctrica y también permite probar los algoritmos de control para diferentes impedancias de la red eléctrica, bajo desequilibrios en la red eléctrica.

2.1.3. Sistema completo

El sistema completo puede dividirse en dos subsistemas bien diferenciados; por un lado está el bloque conformado por el convertidor, denominado “Bloque Convertidor” y por otro, está el subsistema de la turbina del aerogenerador, el cual, se denomina “Bloque Turbina”.

2.1.4. “Bloque Convertidor”

El convertidor, que se emplea como plataforma para realizar ensayos prácticos, se denomina “CONDOR Converter” y se divide en el “Sistema de Electrónica de Potencia” y el “Sistema de Electrónica de Control”.

2.1.4.1. “Sistema de Electrónica de Potencia” del “CONDOR Converter”

El “Sistema de Electrónica de Potencia” es un convertidor “back-to-back” NPC de tres niveles. La potencia nominal del convertidor es de 100kVA y se trata de una estructura completamente bidireccional, o lo que es lo mismo, “back-to-back”, es decir, puede entregar energía a la red y por otro lado tomar energía de la red para excitar a la máquina AC.

2.1.4.2. “Sistema de Electrónica de Control” del “CONDOR Converter”

El “Sistema de Electrónica de Control” del “CONDOR Converter”, está formado por la tarjeta comercial “TMS320C6713DSK” del DSP “TMS320C6713” y la tarjeta “FPGA & USB INTERFACE” basada en la FPGA “SPARTAN 3E”.

La función del “Sistema de Electrónica de Control” del “CONDOR Converter” es la de realizar todo el control correspondiente al convertidor, realizar la lectura de señales procedentes de la turbina y realizar la comunicación entre la tarjeta de control de la turbina y la del propio convertidor.

La tarjeta comercial “TMS320C6713DSK” y la parte de la tarjeta “FPGA & USB INTERFACE” perteneciente a la FPGA, forman lo que se llama el “Módulo Coprocesador”. El resto de elementos del “Sistema de Electrónica de Control” que no forman parte del “Módulo Coprocesador” forman el llamado “Módulo Interfaz”.

El “Módulo Coprocesador” se encarga fundamentalmente de excitar a los ‘drivers’ de los IGBTs, analizar las señales de error procedentes de los mismos, analizar las señales procedentes de los ADCs, realizar la lectura de las señales procedentes del encoder de la turbina y realizar la excitación de los relés.

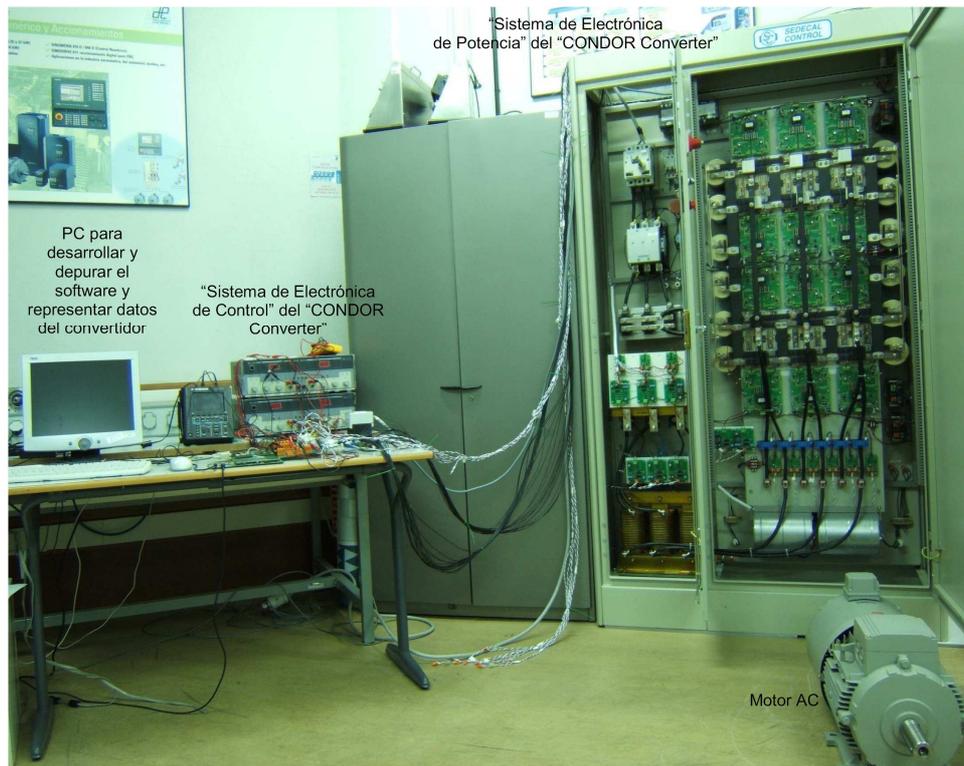


Fig. 11. Fotografía del “Sistema Electrónico de Potencia” y “Sistema Electrónico de Control” del “CONDOR Converter”. [21]

2.1.5. “Bloque Turbina”

En cuanto al subsistema correspondiente a la turbina eólica, también se puede decir, que está formado por una parte de potencia, denominada “Sistema de Electrónica de Potencia de la Turbina” y por una parte de control, denominada “Sistema de Electrónica de Control de la Turbina”.

2.1.5.1. “Sistema de Electrónica de Potencia de la Turbina”

El “Sistema de Electrónica de Potencia de la Turbina”, está formado por un equipo que se encarga de simular variaciones de viento y por un motor AC que simula el funcionamiento completo de una turbina eólica. El simulador de viento está formado por un motor DC y el ‘driver’ correspondiente’, mientras que el motor DC y el motor AC forman la llamada bancada de motores, los cuales están unidos mediante un acoplamiento mecánico.

2.1.5.2. “Sistema de Electrónica de Control de la Turbina”

Por su parte, el “Sistema de Electrónica de Control de la Turbina” estará formado por el “Módulo Turbina”, que a su vez se encuentra dividido en las tarjetas “TURBINE INTERFACE” y en la tarjeta comercial “eZdsp™F2812” del DSP “TMS320F2812”. La función del “Módulo Turbina” será la de realizar todo el control correspondiente a la turbina del aerogenerador, es decir, realizar el control y posición de la turbina en función de las señales procedentes de la veleta, encargarse del cambio

de palas, leer las señales procedentes del encóder del anemómetro y realizar la operación que tenga que llevarse a cabo en función de dicha información, etc.

2.1.6. Aspecto final del sistema completo diseñado

En la siguiente figura se presenta todo el sistema completo desarrollado, sustituyendo el motor AC por una turbina eólica real, por lo que se puede observar la aplicación directa del sistema en el sector de la aerogeneración, que es el principal objetivo de este proyecto. Cada aerogenerador lleva asociado su propio PC, de tal modo, que todos los PCs estarán comunicados entre sí vía Ethernet y todo el sistema quedaría supervisado por otro PC ubicado en la central principal de control.

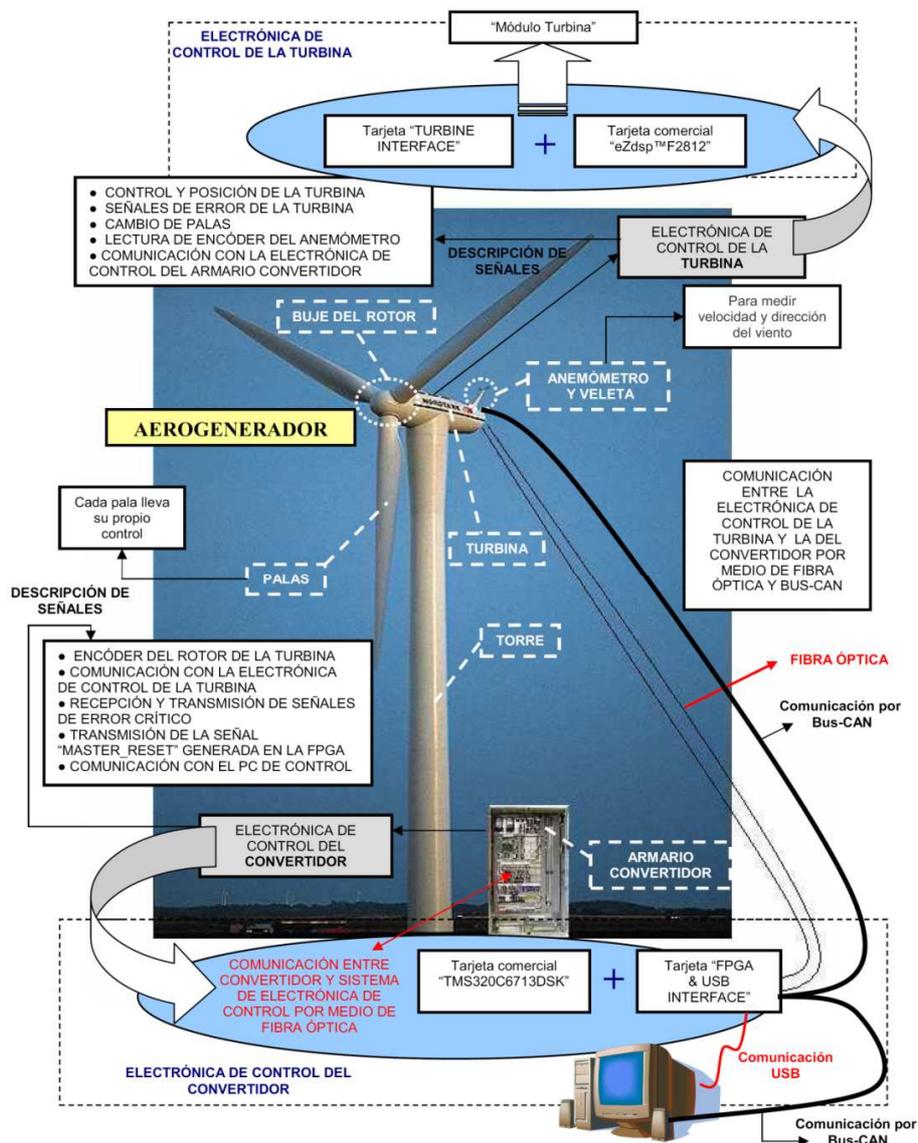


Fig. 12. Sistema electrónico de control y potencia de una turbina eólica de velocidad y paso variable de 250kVA [21]



3. DESCRIPCIÓN DEL PROTOTIPO



3. DESCRIPCIÓN DEL PROTOTIPO

Este proyecto fin de carrera se ha realizado en el laboratorio de investigación del departamento de Ingeniería Eléctrica de la Universidad Carlos III de Madrid que dispone del material necesario para implementar el hardware de comunicación y construir el prototipo del aerogenerador:

- Máquina eléctrica de AC de rotor bobinado de 1.2 kW del fabricante LETAG. (funciona como generador eléctrico doblemente alimentado DFIG).
- Máquina eléctrica de corriente continua de 2 kW del fabricante LETAG (funciona como turbina, arrastra a la máquina AC).
- Dos convertidores trifásicos para el control de la máquina DFIG.
- Inversor SEMISTACK-IGBT (SKM GB 123D SKD 51) de SEMIKRON para el control de la máquina de corriente continua.
- Tarjeta adquisición de datos de *National Instruments*.
- Tarjeta de salida PWM de *National Instruments* para los convertidores electrónicos.
- Un ordenador que funciona como ordenador servidor.
- Instrumental de medida: polímetros, osciloscopio, etc.

Adicionalmente, la Universidad Carlos III de Madrid dispone de los programas, software *Matlab/Simulink* y *PSIM* así como de ordenadores que funcionarán como ordenadores esclavos.

Se ha realizado un trabajo en equipo entre el autor del presente proyecto, el compañero que desarrolló el estudio del prototipo de la turbina [7], los tutores y el personal técnico de apoyo a los laboratorios docentes y de investigación.

Del proyecto [21] se extrajeron las ideas de utilizar una bancada de motores como equivalente a un aerogenerador, cuyas ventajas de ocupación de espacio y maniobrabilidad frente a trabajar con un aerogenerador real son reseñables para su estudio en laboratorio; La posibilidad de realizar el control del mismo mediante algún tipo de tarjeta. En nuestro caso, las tarjetas de adquisición de datos de *National Instruments*; así como la posibilidad de abaratar costes mediante un diseño de control innovador y rentable.

Con el objetivo de poder investigar el comportamiento estático y dinámico de los aerogeneradores, así como diseñar sistemas de control y disponer repetitividad en los ensayos, y la evidente necesidad de aprovechar el espacio disponible se hace

imprescindible la implementación de un emulador de turbina eólica. Este emulador servirá para reproducir con exactitud la conducta mecánica del rotor de una turbina eólica (Fig. 13).

Para llegar a este objetivo se realizará la emulación de la turbina eólica [7]. En este caso se trata de la utilización de un ordenador para reproducir la dinámica de la turbina eólica que incluya el par medio y los pares oscilantes a los que se encuentra sometido. Estos son, principalmente y para el aerogenerador que no disponen unión rígida de la palas al buje, el efecto de sombra de torre, el efecto de la cortadura del viento y el peso de las palas. Este ordenador se encargará además de controlar una máquina eléctrica (en nuestro caso un motor DC) de forma electrónica a través de un convertidor para que desarrolle un par equivalente al proporcionado por la turbina eólica. Un motor DC de excitación independiente resulta ser ideal para este tipo de aplicaciones desde el punto de vista de su fácil control, ya que su par resulta ser proporcional a la corriente de inducido, y desde el punto de vista de su fácil implementación. El motor DC se encarga de arrastrar un generador asíncrono doblemente alimentado, formando en conjunto un aerogenerador DFIG.

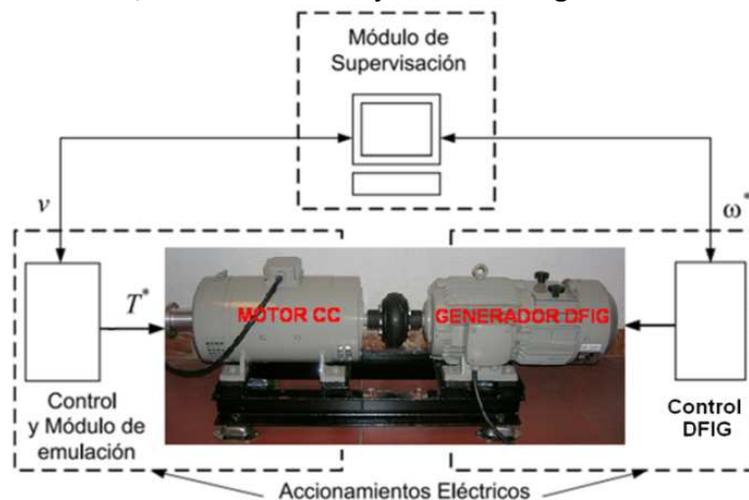


Fig. 13. Emulador de una turbina eólica

A continuación vemos las principales ventajas de esta opción:

- Una máquina eléctrica es mucho más barata y más sencilla de regular que un prototipo a escala de la turbina eólica.
- Mediante una máquina eléctrica se dispone la posibilidad de simular diferentes modelos de turbinas eólicas, sin más que cambiar parámetros en el programa de control. De esta forma, podríamos simular turbinas eólicas de paso fijo, paso variable, monopalas, bipalas, etc., mientras que si utilizamos prototipos de turbinas eólicas, tendríamos que disponer de un modelo de cada tipo.
- Otra ventaja es prescindir de un túnel de viento para reproducir condiciones atmosféricas, que resultaría además de caro muy complejo.

Son estas ventajas las que hacen que se emule el comportamiento de la turbina eólica mediante un prototipo físico basado en un motor eléctrico.

En el laboratorio se ha utilizado una bancada de motores. Un motor de continua, (empleado como prototipo de las palas del aerogenerador) y un motor de alterna, unidos por su eje.

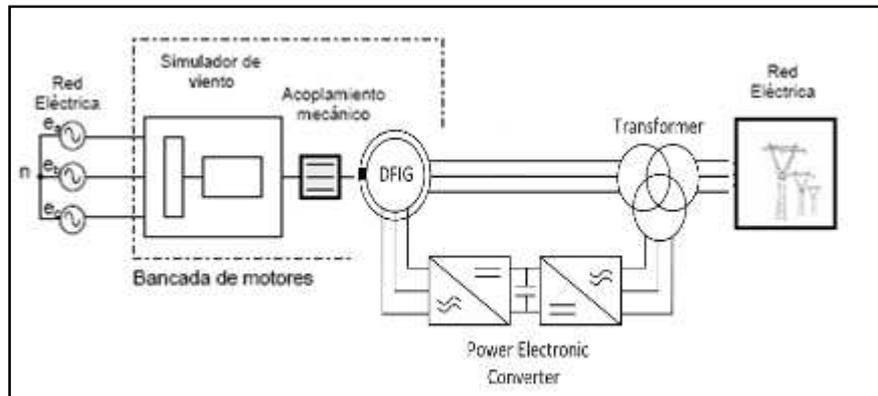


Fig. 14. Esquema de simulador de viento conectado a la red eléctrica.

La bancada de motores nos permitirá disponer en el laboratorio el equivalente a un molino eólico, sin la necesidad de tener un molino real. Permittiéndonos hacer ensayos en el laboratorio, en lugar de en un parque eólico.

Esto es posible, gracias a poder ejecutar un modelo mediante *xPC Target* el cual hará comportarse al motor de corriente continua como actuarían las palas de un aerogenerador real.

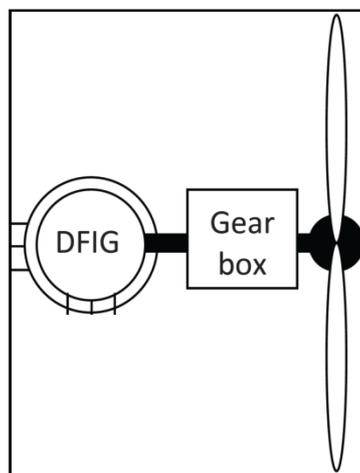


Fig. 15. Generador doblemente alimentado, caja de engranajes y palas

Este modelo real de molino eólico se ha sustituido en el laboratorio por una bancada de motores, como en [21]. Sin necesidad de disponer de palas.

3.1. BANCADA DE MOTORES

La bancada de motores consta de un motor DC y de un motor AC acoplados mecánicamente.



Fig. 16. Motor DC, acoplamiento mecánico, motor AC

3.1.1. MOTOR DC

El motor DC es el que se encarga de transmitir al motor AC, a través de su eje acoplado mecánicamente, los movimientos que simulan el comportamiento del viento, y que son generados a partir del programa compilado en el PC *Target*. Las características del motor DC son las que se muestran a continuación:

- Tensión de inducido: 220V
- Corriente de inducido: 11A
- Tensión del inductor: 200V
- Corriente del inductor: 0.9A
- Velocidad nominal: 1000rpm
- Potencia nominal: 2.2KW



Fig. 17. Placa de características del motor DC

3.1.2. MOTOR AC

El motor AC (asíncrono y trifásico) junto con el acoplamiento mecánico del eje es la parte del prototipo que equivale a comunicar el “Sistema de Electrónica de Potencia de la Turbina real” con el “Sistema de Electrónica de Potencia”, es decir, los movimientos que simulan el comportamiento del viento y que hacen mover al propio motor AC, son convertidos a corriente alterna de manera que se pueda conectar a red.

- Tensión nominal: 230/400V
- Corriente nominal: 8.3/4.8A
- Frecuencia: 50 Hz
- Velocidad nominal: 930rpm
- Potencia nominal: 1.5KW
- $\text{Cos}\phi$: 0.60



Fig. 18. Placa de características del motor AC



4. xPC TARGET



4. xPC TARGET.

Para el control del aerogenerador en tiempo real se utilizará la aplicación *xPC Target* de *Mathworks*, que dispone *Simulink* de *Matlab*, la cual permite trabajar en tiempo real. Esta aplicación facilita realizar modelos en *Simulink* que, mediante hardware añadido al ordenador (tarjeta de adquisición de datos *National Instruments*) [15], nos permite operar con entradas y salidas en sistemas reales, por ejemplo el motor de corriente continua.

Se utilizarán tarjetas de adquisición de datos de *National Instruments*, modelos *SCB-68* y *SCB-100*. Dichas tarjetas son el hardware que permite tener, en el ordenador en el que se ejecuta el modelo, entradas y salidas: digitales, analógicas y PWM con las que poder actuar con el sistema en tiempo real.

El empleo de la aplicación *xPC Target* [13] nos permite reutilizar ordenadores obsoletos para aplicaciones actuales. Servirán para el cálculo de procesos en tiempo real. Es posible así conseguir un coste relativamente bajo. Únicamente es necesaria la implementación de hardware específico de entradas/salidas para la conexión y comunicación entre el PC y el sistema a controlar en tiempo real.

xPC Target [13], [14] es una aplicación *Host/Target* (servidor/cliente) de *Matlab* que permite conexas modelos creados en *Simulink* (aplicación de *Matlab*) con sistemas físicos reales, y simularlos en tiempo real.

De esta manera podremos realizar un prototipo que nos permitirá ejecutarlos en un ordenador destino, *PC Target*, visualizar los resultados obtenidos en tiempo real permitiéndonos tomar decisiones como cambiar en función de los datos obtenidos los parámetros de diseño.

La aplicación *xPC Target* nos permite:

- Programar modelos complejos de manera sencilla mediante la herramienta *Simulink*, sin necesidad de conocimientos de lenguajes de programación. No siendo necesario programar en lenguaje de bajo nivel.
- *xPC Target* funciona como un sistema operativo en tiempo real en el *PC Target*, lo que consigue que el modelo actúe como un proceso real.
- Controlar un sistema de manera remota desde otro ordenador.

El control remoto de estos sistemas es muy útil por varias razones: (1) permite realizar modificaciones en los parámetros del sistema sin manipular los equipos directamente, con lo que se aumenta la seguridad; y (2) permite obtener interesantes e importantes resultados a nivel de investigación, pues se pueden monitorizar en un PC las salidas del convertidor y las diferentes actuaciones sobre los equipos empleados en las pruebas.



En el laboratorio se ha trabajado con un motor de corriente continua que actúa equivalentemente a las palas de un aerogenerador, el cual arrastra a un motor de alterna que se comporta como generador. Esto es equivalente a la generación de un aerogenerador el cual entrega energía a la red eléctrica.

Se ha implementado un programa con *Matlab* de control del motor de corriente continua para que se comporte como las palas de un aerogenerador. En dicho modelo se tendrá en cuenta las características que presenta el giro de las palas del aerogenerador como son las fluctuaciones de las palas, del par, cuando pasan por la torre del aerogenerador así como el control frente a fallos.

Este modelo implementado en la aplicación de *The Mathworks, Simulink* de *Matlab* [13] en un ordenador origen, *PC Host*, se instalará en un ordenador destino, *PC Target*, que independientemente del *PC Host*, realizará los cálculos para controlar el motor de corriente continua.

Esta aplicación permite, generando un modelo (ya sea una señal sinusoidal, como veremos en lo que sigue) como el del motor de corriente continua en un ordenador origen (*PC Host*) enviar dicho modelo a un ordenador destino (*PC Target*) o varios, (o varios ordenadores destino con diversos modelos) para que dicho ordenador destino trabaje independientemente de nuestro ordenador origen (*PC Host*), de manera que a éste último se le libera de carga de trabajo.

La aplicación *xPC Target* genera un código de programación en lenguaje C, a partir de los modelos que se implementan en *Simulink*, que se ejecuta en el *PC Target*, sin necesidad de que en él haya instalado ningún sistema operativo (Linux, Windows). Por lo que puede emplearse un ordenador de pocas prestaciones a nivel actual, pero suficiente, a nivel de cálculo, utilizando todas sus capacidades de computación.

Para facilitar la visión de la topología hardware que vamos a implementar, se puede hacer notar que necesitaremos un único *PC Host* que cargará las aplicaciones en el *PC Target* o *PCs Targets* que se necesiten.

Como estos *PC Targets* se quedarán fijos para la aplicación en concreto, en este caso el control del motor de corriente continua, podemos apreciar que el *PC Host* al no tener que ser un ordenador sobremesa estático en el lugar de trabajo puede ser un ordenador portátil (aunque en nuestros ensayos es un PC de sobremesa) que cargue la aplicación *xPC Target* en el *PC Target* y que posteriormente no sea necesaria su estancia fija en el recinto de instalación de la topología de comunicación en donde se ejecuta el modelo.

En la aplicación en tiempo real *xPC Target* se distinguen dos componentes:

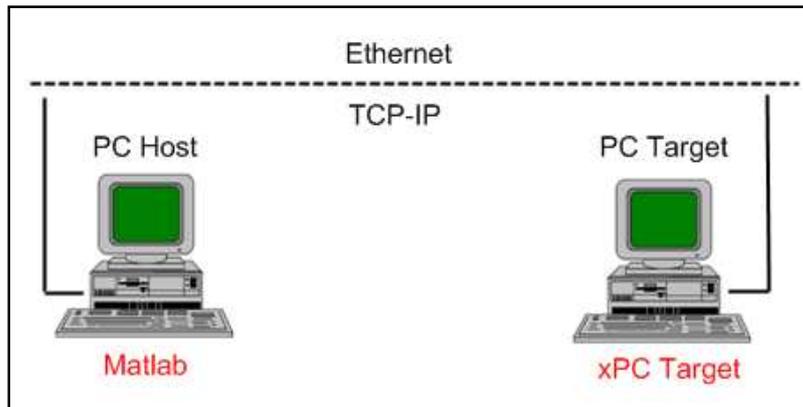


Fig. 19. Esquema de conexión *Host - Target*

- **PC Host.**

Este es el ordenador origen. En él se tiene instalado el sistema operativo Windows XP, el programa *Matlab* (también debe tener instalado Visual C para generar el código de programación requerido por la aplicación *xPC Target*) y una tarjeta de red necesaria para la conexión mediante el protocolo TCP-IP con el *PC Target*.

Es nuestro ordenador de trabajo. Generaremos en él los modelos necesarios (que se cargarán en el *PC Target*) para enviar datos, y como finalidad, para controlar el motor de corriente continua.

- **PC Target.**

Este es el ordenador destino. Puede haber más de uno. No tiene instalado sistema operativo. En él se ejecuta la aplicación *xPC Target* diseñada para el *PC Target*. Dicha aplicación se ejecuta habitualmente desde un disquete, que previamente se generó en el *PC Host*. Dicha aplicación especifica la dirección IP del *PC Target* en cuestión y protocolo de comunicación TCP-IP mediante conexión con cable RJ-45.

Es el ordenador que ejecuta el cálculo en tiempo real. Dicho cálculo consta de las siguientes fases:

- **Inicialización:** En un primer momento se ejecuta en él la aplicación *xPC Target* mediante un disquete.
- **Volcado del programa desde el PC Host:** En un segundo lugar, mediante *Simulink* de *Matlab* del *PC Host* y a través de la conexión TCP-IP descargaremos al *PC Target* los modelos a utilizar.
- **Ejecución y funcionamiento en tiempo real:** Posteriormente podremos arrancar, parar y adquirir datos del *PC Target*, remotamente desde el *PC Host*, mediante los comandos necesarios desde *Matlab*.

Para que el sistema disponga de canales de entrada y salida para la aplicación en tiempo real se pueden instalar tarjetas adicionales como, por ejemplo: tarjetas de adquisición de datos (Tarjetas *National Instruments*) [15], o tarjetas de salidas PWM para el control de los convertidores trifásicos del prototipo del aerogenerador.



Fig. 20. Tarjetas de adquisición de datos NI SCB-68, SCB-100

Las entradas y salidas pueden ser digitales, analógicas y PWM. Además de estos dos componentes principales existe otro componente imprescindible: *la comunicación entre ambos*.

La aplicación *xPC Target* de *Matlab* [13] permite comunicación serie, por cable RS-232 o por cable de red RJ45 mediante tarjetas Ethernet instaladas en cada PC las cuales utilizan el protocolo TCP-IP. Dicho programa tiene restricciones de tarjetas Ethernet empleadas. Se han utilizado tarjetas de red compatibles con la aplicación, R8319.

Debido a que existen únicamente estas dos maneras de interconexión entre PC *Host* y PC *Target*, como el envío de datos es más rápido por cable de red, se ha utilizado dicho tipo de comunicación entre ambos

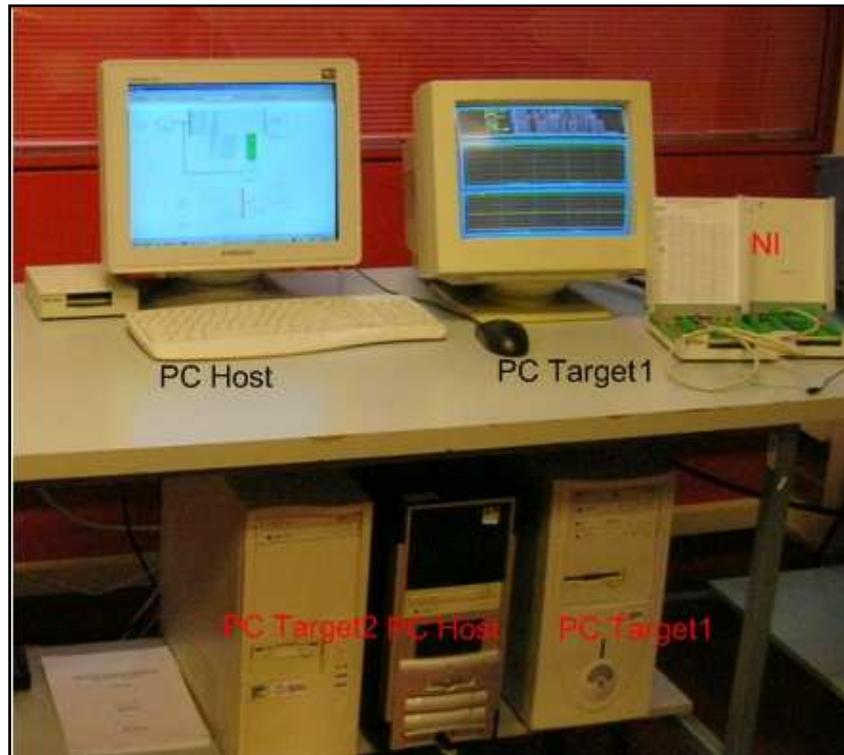


Fig. 21. PC *Host*, PC *Targets* y tarjetas National Instruments



5. ESPECIFICACIONES DEL SISTEMA DE CONTROL



5. ESPECIFICACIONES DEL SISTEMA DE CONTROL

El sistema de control permite gobernar el motor DC acorde a las condiciones de viento existentes. El ámbito de este proyecto es facilitar el hardware de comunicación, que no ponga limitación a la comunicación, adquisición y envío de dichas señales de control que se utilizará en [7]. Por lo que este sistema de control se realizará en trabajos futuros.

5.1. SEÑALES DE ENTRADA

A la hora de implementar el hardware de comunicación deberemos permitir la adquisición de las siguientes señales:

- Para controlar la máquina de corriente continua (equivalente a las palas del molino)
 - Corriente y tensión del inducido
 - Corriente y tensión del inductor
 - Velocidad de giro de la bancada
- Para el control de la máquina de AC de rotor bobinado:
 - 3 Corrientes y 3 tensiones del estator
 - 3 Corrientes y tensiones del rotor
 - Tensión de la etapa de continua de los convertidores trifásicos
 - Velocidad de giro de la bancada (la misma que en caso de la máquina DC)

5.2. SEÑALES DE SALIDA

Las medidas en el motor saldrán hacia el PC *Target*, realimentando las medidas del modelo, teniendo una respuesta en tiempo real para el sistema.

- Para controlar el motor AC
 - 6 Salidas PWM de los dos convertidores trifásicos de la máquina AC
 - pulsos al inversor de la máquina DC.

La velocidad de ejecución de la aplicación implementada, el hardware construido con el PC *Target* más la tarjeta de adquisición de datos de *National*

Instruments, no pone límite a la respuesta de estas señales. Por lo que es más que válida para el sistema aerogenerador desarrollado.

Se enviarán como salidas del hardware, además, los pulsos hacia el convertidor, que disparará el control de los mismos, permitiéndonos gobernar el giro del motor. Aquí se presenta la incógnita de que nuestro hardware sea capaz de responder a tiempo. Comprobando posteriormente que es más que capaz. El retardo del sistema es de $100\mu\text{s}$ y nuestros convertidores necesitan conmutar a 1ms .

A continuación se muestra un esquema del convertidor:

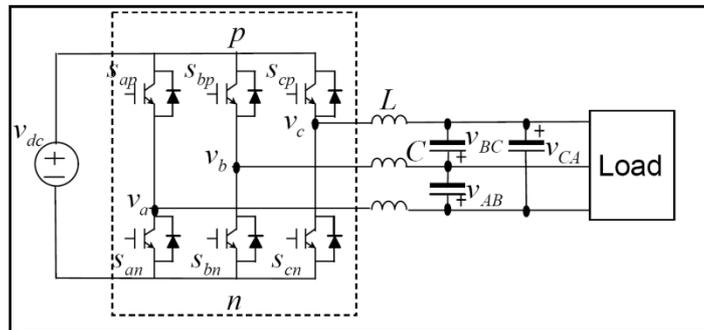


Fig. 22. Convertidor DC/AC

En la siguiente imagen se muestra el convertidor utilizado en el laboratorio:



Fig. 23. Convertidor utilizado en el laboratorio



6. DESCRIPCIÓN DE LA ARQUITECTURA DE COMUNICACIONES

6. DESCRIPCIÓN DE LA ARQUITECTURA DE COMUNICACIONES

El objetivo del hardware a implementar es la comunicación entre un PC *Host* que implementa un programa que se descargará en forma de “*driver*” en un PC *Target*, además de gobernarlo remotamente.

También ha de ser capaz de adquirir señales de la manera más rápida y eficiente a través de la tarjeta de adquisición de datos de *National Instruments* [15], con el objetivo de disparar el convertidor que gobierna al motor DC. Por lo que el esquema simplificado de lo que buscamos y finalmente conseguiremos es el siguiente:

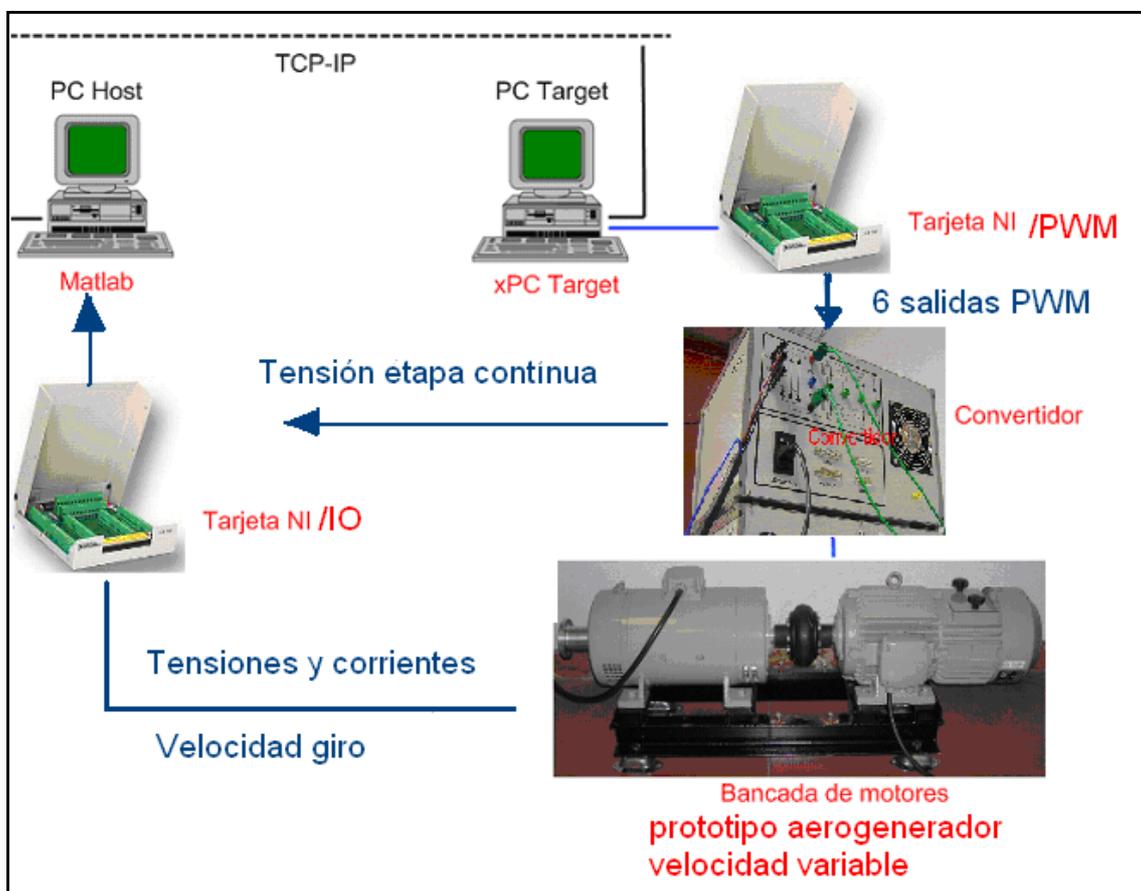


Fig. 24. Arquitectura hardware de las comunicaciones del prototipo del aerogenerador de velocidad variable.

Para conseguir el esquema arriba mostrado se realizaron los siguientes estudios para comprobar las capacidades de la aplicación *xPC Target* para nuestro problema.

6.1. RESULTADOS

6.1.1. HARDWARE DE CONEXIÓN ENTRE PC *HOST*-PC-104 CON CABLE RS232

Se pretende que un PC *Target* (ordenador destino), que en este apartado este ordenador destino es el PC-104, realice los cálculos de un modelo de *Simulink* enviado de manera remota por un PC *Host* (ordenador anfitrión).

De esta manera liberaremos de carga de cálculo a nuestro ordenador principal, PC *Host*, pudiendo controlar un ordenador destino, PC *Target* o varios, que no tiene/en por qué estar en la misma sala.

La aplicación *xPC Target* de *Matlab* nos permite realizar este control, pudiendo realizar cálculos en tiempo real y permitiéndonos configurar sus características. Entre otras, estableciendo la conexión mediante un cable serie RS-232 o mediante una configuración TCP/IP.

Esta última, a su vez, nos permite realizar la conexión con un cable RJ45 (cruzado) desde la tarjeta Ethernet de nuestro PC *Host* a la tarjeta Ethernet del PC *Target*; o realizar la conexión a través de una LAN (*Red de Área Local*) conectándonos a un Router con un cable RJ45 (recto) desde la tarjeta Ethernet del PC *Host* al Router y con otro cable/s RJ45 (recto) desde la tarjeta Ethernet del PC/s *Target* al/ a los puerto/s libre/s del Router.



Fig. 25. Cable RS-232



Fig. 26. Cable RJ45

A continuación se muestra la apariencia física del ordenador PC-104 utilizado.



Fig. 27. PC-104. De izquierda a derecha y de arriba hacia abajo:
(Fuente alimentación, unidad de disquete, unidad de CD-ROM, teclado y placa base)



Fig. 28. PC-104. De izquierda a derecha y de arriba hacia abajo:
(Fuente alimentación, unidad de disquete y placa base)



Fig. 29. PC-104. De izquierda a derecha y de arriba hacia abajo:
(Fuente alimentación, unidad de disquete, teclado y placa base)
Conexión de teclado y Monitor (clavija azul) en la parte de abajo a la derecha



Fig. 30. Monitor del PC-104.

El material que se necesita es:

- Un PC Host con sistema operativo Windows en donde pueda ejecutarse Matlab (Windows XP, en este estudio) - con una unidad de disquete, Matlab y un compilador Visual C (Microsoft Visual Studio) instalado en él.
- El PC-104 con un monitor VGA y unidad de disquete. Conexión mediante cable RS-232 (cuando se conecte debe ir a los dos COM1, al COM1 del PC Host y al COM1 del PC-104)

6.1.1.1. Configurando el PC Host

Al PC *Host* hay que decirle con qué PC *Target* es con el que se comunica. También hay que determinar la configuración que se almacena en el disco de arranque que carga la aplicación *xPC Target* en el PC *Target*.

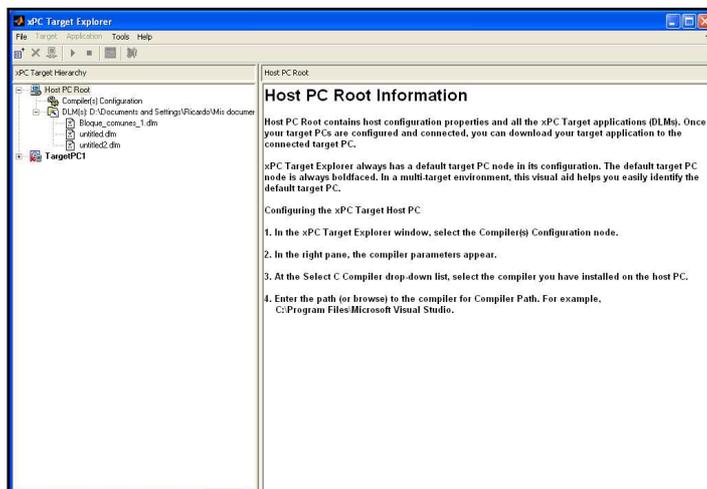


Fig. 31. Imagen del xPC Target Explorer

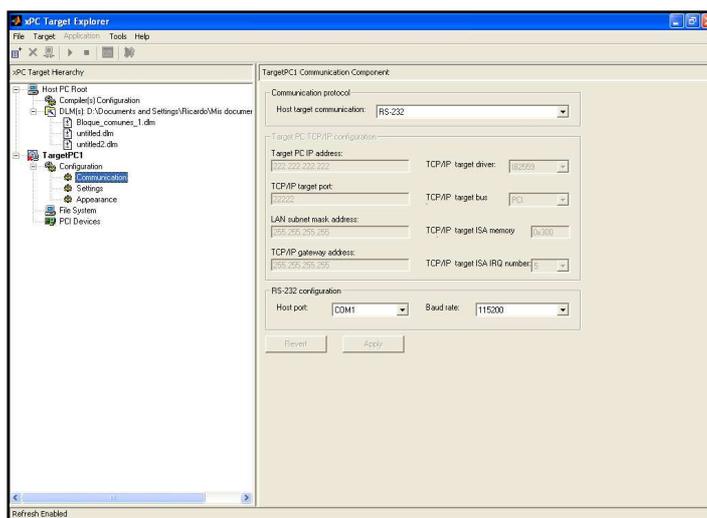


Fig. 32. xPC Target Explorer. Communication. Conexión RS 232

6.1.1.2. Comenzando la comunicación. Crear un modelo en Simulink para ejecutar con xPC Target

Una vez se haya creado un modelo *Simulink*, se necesita hacer algunos ajustes para que pueda ejecutarse el modelo en *xPC Target*. Esta sección describe la creación de un modelo sencillo.

En un primer lugar se crea un archivo en *Simulink*. En este caso se crea un modelo que genera una señal sinusoidal. Dicho modelo se compila y se envía al PC *Target* y su ejecución se muestra en el monitor del mismo.

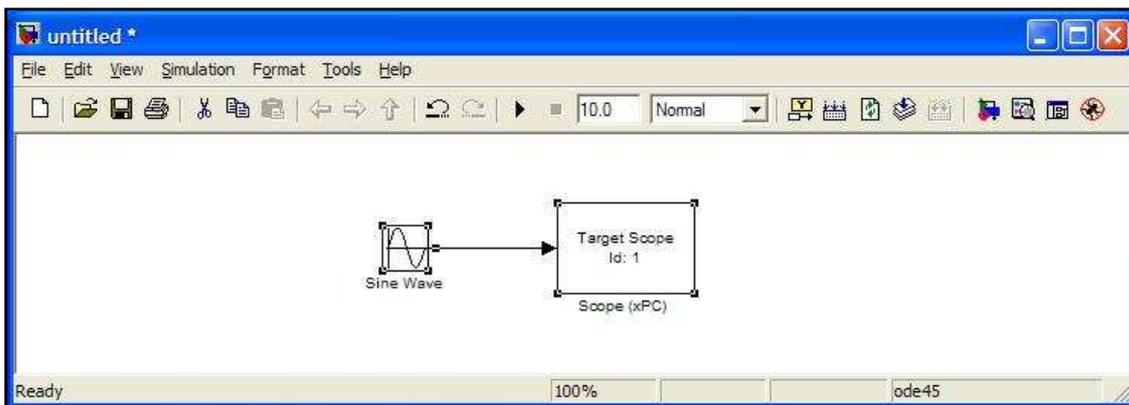


Fig. 33. Modelo Simulink. Señal sinusoidal mostrada en el PC *Target*

Se compila en el PC *Host* el modelo, que se enviará a través del cable serie por el COM1 al PC *Target* (PC-104).

Una vez compilado, en el monitor del PC *Target* aparecerá la aplicación xPC *Target* dispuesta a ejecutar el modelo. En la siguiente figura se muestra el “Scope” que se generó en el modelo de *Simulink* y que se compila en el PC *Target*.



Fig. 34. Monitor PC *Target*. Comando *xpcargetspy*

Para ejecutar el modelo de *Simulink* hay que conectarse al PC *Target*.

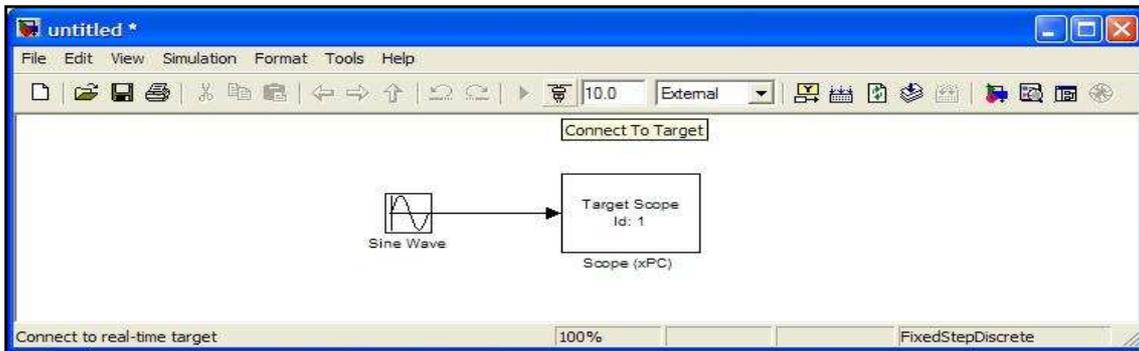


Fig. 35. Conectando con el PC Target

Al ejecutar el modelo de la onda sinusoidal se obtuvo en el monitor del PC Target la siguiente imagen.

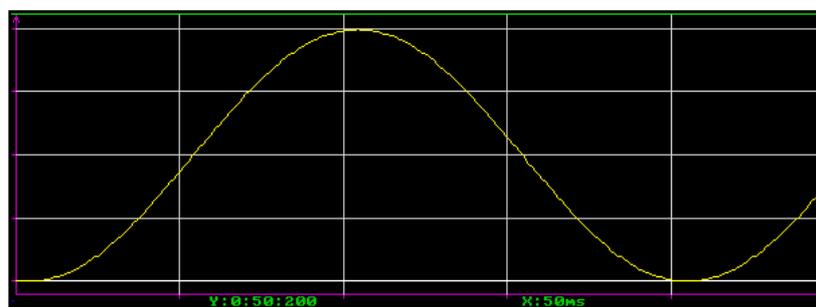


Fig. 36. Monitor PC Target. Señal sinusoidal

Además, al cambiar la escala de tiempos se obtuvo la siguiente imagen. Se incluye además el tiempo de ejecución ($1.617e-5s$) y el tiempo de muestreo ($0.001s$).

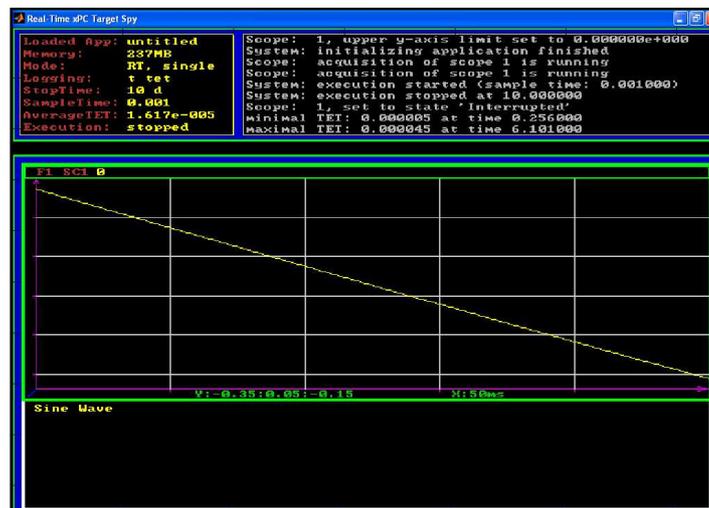


Fig. 37. Monitor PC Target. Ampliación señal sinusoidal

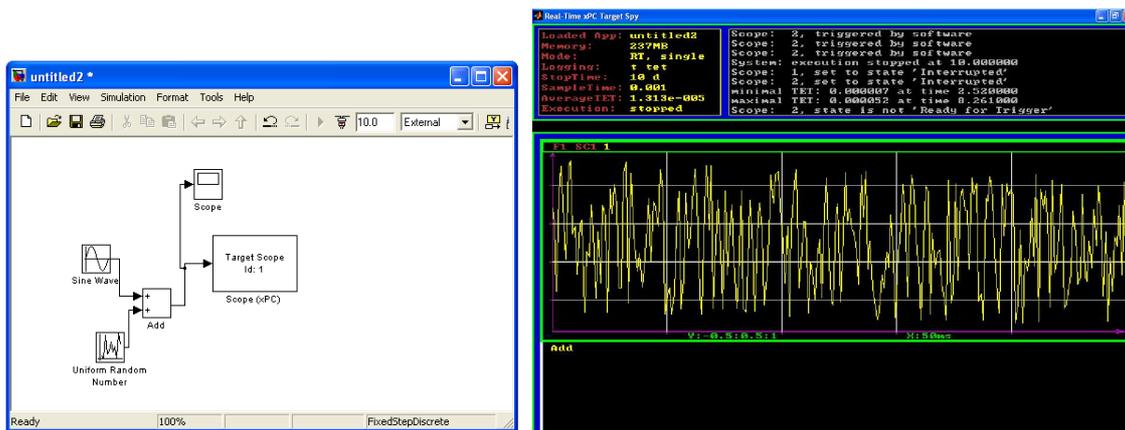


Fig. 38. Ensayo de señal sinusoidal más una aleatoria

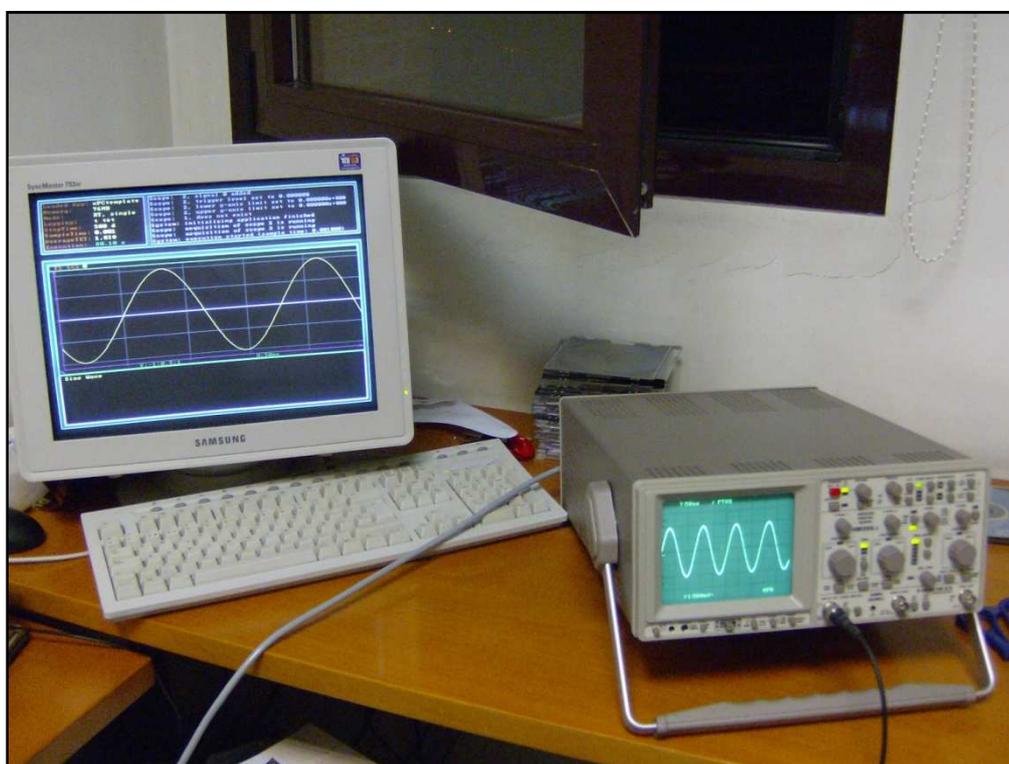


Fig. 39. Señal del PC Target obtenida en el osciloscopio



Fig. 40. Señal del *PC Target* obtenida en el osciloscopio. Ensayo sinusoidal más ruido

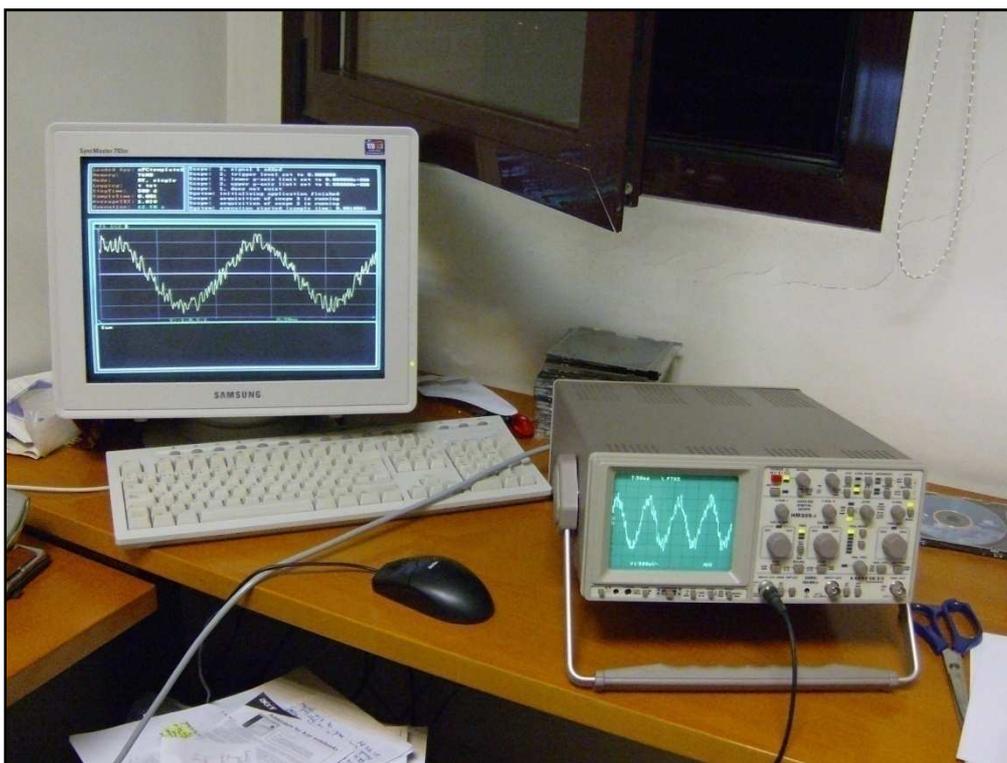


Fig. 41. Señal del *PC Target* obtenida en el osciloscopio. Segundo ensayo sinusoidal más ruido

6.2. HARDWARE DE CONEXIÓN ENTRE PC *HOST*-PC *TARGET*

6.2.1. CONEXIÓN *HOST-TARGET*. Mediante cable RJ45 cruzado

En esta sección se parte del estudio presentado en el apartado 6.1.1, la conexión de un PC *Host* con un PC *Target* (PC-104). En el paso anterior el estudio se realizó con un PC de sobremesa y el PC-104.

En este apartado, como variante del anterior 6.1.1, se utilizaría el protocolo TCP/IP mediante la comunicación con cable de red RJ45.

Se realizará la conexión con un cable RJ45 (cruzado) desde la tarjeta Ethernet del PC *Host* a la tarjeta Ethernet del PC *Target*.



Fig. 42. Cable RJ45



Fig. 43. Tarjeta Ethernet RTL8139D [22]

Se establecerá la conexión del PC *Target* con el PC *Host* mediante un cable RJ45 cruzado. Esto es, un cable RJ45 en el que la posición de varios de los cables que lo componen se cambian de posición como se puede ver en la Fig. 45.



Fig. 44. Cable RJ45

Cable cruzado (4 pares, tarjetas 10/100/1000baseT) T568B

Pin N°	Extremo 1	Extremo 2	Función
1	Orange	Green	Transceive data +
2	Orange	Green	Transceive data -
3	Green	Orange	Receive data +
4	Blue	Blue	Bi-directional Data +
5	Blue	Blue	Bi-directional Data -
6	Green	Orange	Receive data -
7	Orange	Blue	Bi-directional Data +
8	Orange	Blue	Bi-directional Data -

Cable cruzado (2 pares, tarjetas 10/100baseT)

Pin N°	Extremo 1	Extremo 2	Función
1	Orange	Green	Transceive data +
2	Orange	Green	Transceive data -
3	Green	Orange	Receive data +
4	Blue	Blue	Bi-directional Data +
5	Blue	Blue	Bi-directional Data -
6	Green	Orange	Receive data -
7	Orange	Blue	Bi-directional Data +
8	Orange	Blue	Bi-directional Data -

Fig. 45. Código de colores de cables RJ45 cruzados

La conexión se realiza conectando un extremo del cable RJ45 a la tarjeta Ethernet del PC *Host* y el otro extremo a la tarjeta PCI de Ethernet (Fig. 43) conectada en el PC *Target*.

A continuación se muestra una imagen del PC *Target* utilizado. Un Pentium III a 600Mhz con 64Mb de RAM.



Fig. 46. PC *Target*

Para este ensayo se utilizará lo siguiente:

- Un PC *Host* con Windows (Windows XP) - con una unidad de disquete, Matlab y un compilador Visual C (Microsoft Visual Studio) instalado en él. Debe tener un puerto libre de conexión Ethernet.
- Un PC, en este caso un Pentium III a 600MHz y 64Mb con un monitor VGA, unidad de disquete ratón y teclado. Debe tener una tarjeta de red Ethernet compatible (Ver página 2-33 [12]).

Conexión mediante cable RJ45 cruzado (Fig. 43 y Fig. 45)

6.2.1.1. Configurando el PC *Host*

Para este apartado hay que determinar los valores del protocolo TCP/IP para el PC *Host*.

Se ha instalado una tarjeta compatible con la aplicación *xPC Target*.

Puesto que es una tarjeta de conexión PCI, en **TCP/IP Target bus** se debe elegir PCI.

En **Target PC IP address** se elige la dirección del PC *Target*.

En **TCP/IP Target port** aparece por defecto 22222, es válido cualquier valor comprendido entre 20000 y 65536 (valores no reservados para telnet, ftp, etc. y por tanto que no interfieren, en la comunicación, con estas aplicaciones).

LAN subnet mask address es la dirección de máscara de subred de la red LAN.

Si se prefiere, se puede obtener la dirección de máscara de subred LAN del cuadro de diálogo de Conexiones de red en el PC *Host*. Dependiendo de la plataforma Microsoft Windows, se puede acceder a este cuadro de diálogo de varias maneras. Por ejemplo, en un sistema Windows XP Professional, se puede utilizar esta secuencia:

Inicio>>configuración>> panel de control >> Conexiones de red

Si el ordenador se conecta con un cable cruzado, puede aparecer un cuadro de diálogo como el de la Fig.47. Se debe escribir la máscara de subred y la dirección TCP/IP de puerta del enlace adecuados.

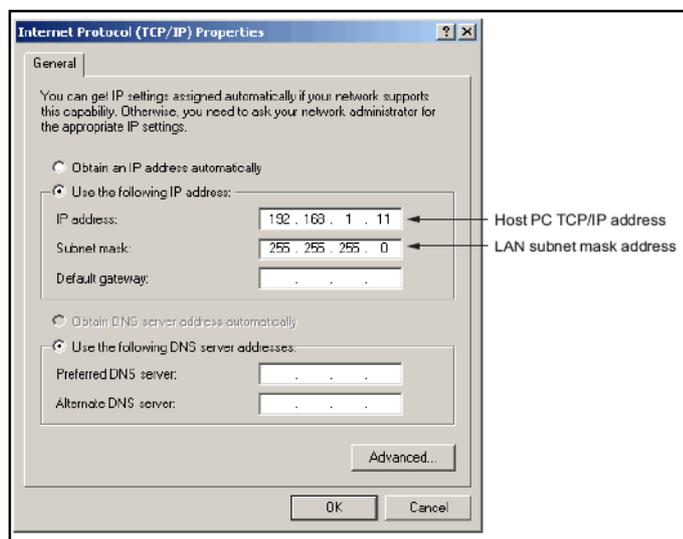


Fig. 47. Dirección IP del PC *Host* [11]

Opcionalmente, se pueden introducir las siguientes propiedades, en función de las circunstancias específicas:

TCP/IP gateway address. Esta propiedad tiene por defecto 255.255.255.255. Esto significa que no se utiliza una puerta de enlace para conectarte a tu PC *Target*. Si te conectas al ordenador con un cable cruzado, se debe dejar esta propiedad como 255.255.255.255.

Si se comunica con el PC *Target* dentro de una red local (LAN), es posible que no se necesite definir una puerta de entrada y cambiar esta configuración.

Si se comunica desde un PC *Host* situado en una LAN diferente del PC *Target*, se necesita definir una puerta de enlace y escribir la dirección IP. Sería necesario definir esto si se trabaja a través de Internet. Se debería preguntar al administrador del sistema para obtener la dirección IP de la puerta de enlace.

Una vez se haya creado un modelo *Simulink*, se necesitan hacer algunos ajustes para que pueda ejecutarse el modelo en *xPC Target*.

Para este apartado se seguirán los mismos pasos que se utilizaron para la comunicación con el PC-104. Siendo en este caso la comunicación mediante un cable de red RJ45 a las tarjetas de red del PC *Host* y PC *Target*, en lugar de utilizar un cable serie RS-232 a los puertos COM de los ordenadores PC *Host* y PC *Target*. Se utilizará el modelo *untitled.mdl* de *Simulink* explicado en el apartado 6.1.1.

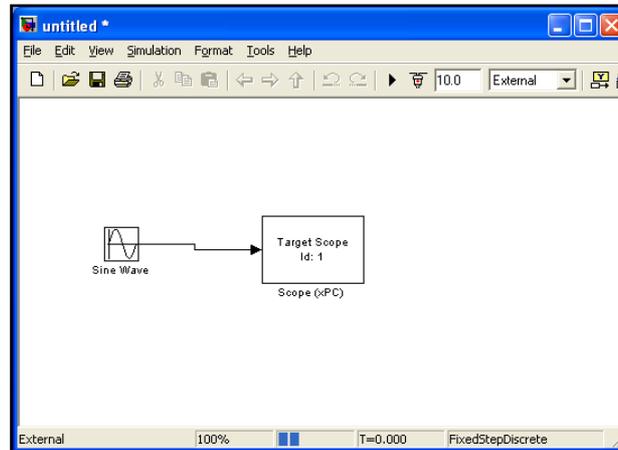


Fig. 48. *untitled.mdl*. Ejecución del modelo

6.2.2. CONEXIÓN HOST-TARGET. Mediante cable RJ45 y Router

Se realizará la conexión del PC *Target* con el PC *Host* mediante cables RJ45 rectos y un Router.

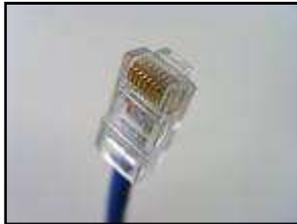


Fig. 49. Cable RJ45

Cable recto (normal/paralelo) T568B

Pin N°	Extremo 1	Extremo 2	Color	Función
1			Blanco - Naranja	Transceive data +
2			Naranja	Transceive data -
3			Blanco - Verde	Receive data +20
4			Azul	Bi-directional Data +
5			Blanco - Azul	Bi-directional Data -
6			Verde	Receive data -
7			Blanco - Marrón	Bi-directional Data +
8			Marrón	Bi-directional Data -

Fig. 50. Código de colores del cable RJ45 recto

La conexión se realiza conectando un extremo de un cable RJ45 a la tarjeta Ethernet del PC *Host* y el otro extremo a un puerto libre del Router. Se necesita otro cable RJ45 para conectar un extremo a la tarjeta PCI de Ethernet (Fig. 43) conectada en el PC *Target* (tarjeta compatible con *xPC Target*) [12] y el otro extremo a un puerto libre del Router.

Se utilizará el mismo PC *Target* que se ha utilizado en la anterior topología 6.2.1.

El PC *Target* utilizado es el mismo que se muestra en la Fig.46. Un Pentium III a 600Mhz de 64Mb de RAM.

Para este apartado se utilizaron:

- Un PC *Host* con Windows (Windows XP)- con una unidad de disquete, *Matlab* y un compilador Visual C (*Microsoft Visual Studio*) instalado en él. Debe tener un puerto libre de conexión Ethernet.
- Un PC, en este caso un Pentium III a 600MHz y 64Mb (Fig. 47) con un monitor VGA, con unidad de disquete, ratón y teclado.
- Un Router. En este caso se ha utilizado un 3com ADSL 11g Wireless. Con cuatro puertos libres RJ45 y alimentación (DC12V 1.25A máx)



Fig. 51. 3com ADSL 11g Wireless con dos cables RJ45 rectos y alimentación

- Un cable recto RJ45 para conectar el PC *Host* al Router y otro cable RJ45 recto para conectar el PC *Target* al Router.

6.3.1. CONEXIÓN PC HOST- VARIOS PC TARGETS

En este apartado se explica cómo se accedió a varios PC *Target* desde un PC *Host*. La aplicación de este apartado es válida para cualquiera de los casos 6.1.1, 6.2.1, 6.2.2 anteriores, partiendo de haber establecido la comunicación ya realizada por cualquiera de los métodos anteriormente explicados. Existiendo la limitación técnica de poderse comunicar con tantos PCs como puertos COM libres haya en el caso de la comunicación serie, y con tantos PCs como puertos libres tenga el Router en el caso de la comunicación mediante protocolo TCP/IP.

En un primer momento se tuvo que añadir en el explorador de la aplicación *xPC Target* un segundo PC *Target*.

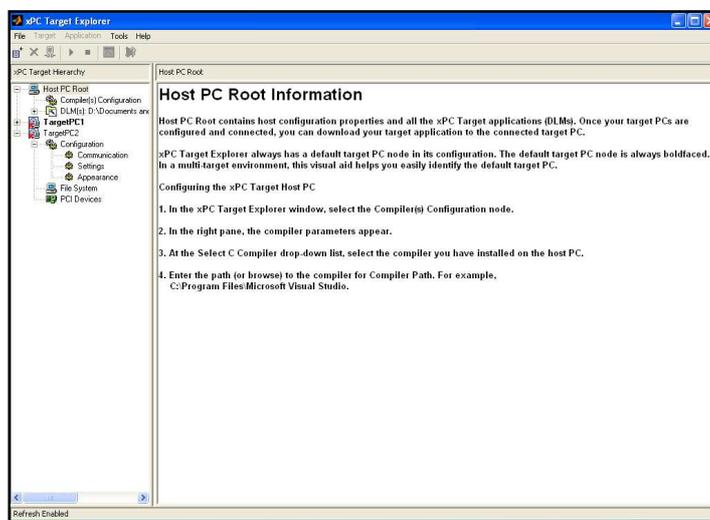


Fig. 52. Adición de un nuevo PC *Target*. *TargetPC2*

Para realizar la comunicación con el PC *Target2* primero se crea un nuevo disquete de arranque con la aplicación *xPC Target*. Lo único que no debe ser igual es *Target PC IP address*, que no debe coincidir con los demás PC *Target*.

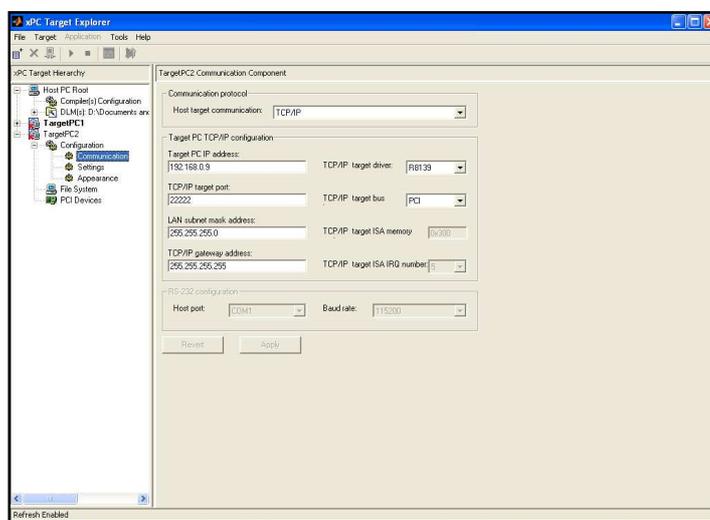


Fig. 53. Comunicación. TCP/IP del *TargetPC2*

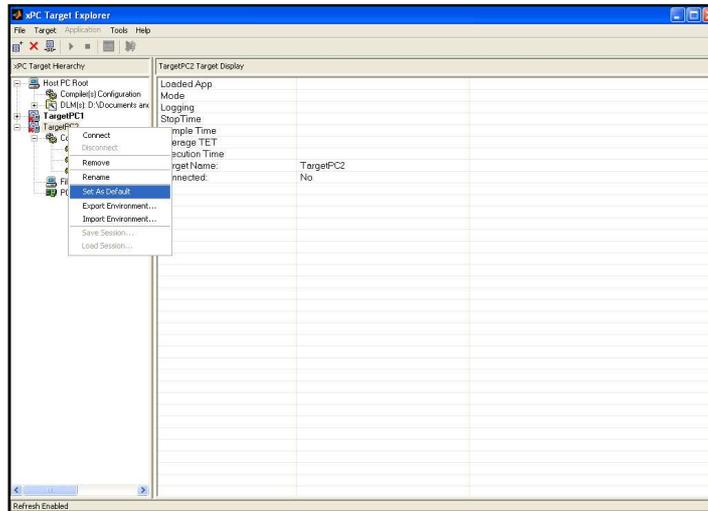


Fig. 54. Estableciendo el *TargetPC2* por defecto

Para establecer la comunicación con varios PC *Target* hay que realizar la comunicación de manera secuencial, de uno en uno, puesto que sólo se puede establecer la conexión simultánea entre un PC *Host* y un único PC *Target*, debido a que la comunicación es para cálculo en tiempo real y la aplicación no permite la comunicación de un PC *Host* con varios PC *Target* simultáneamente. [13]

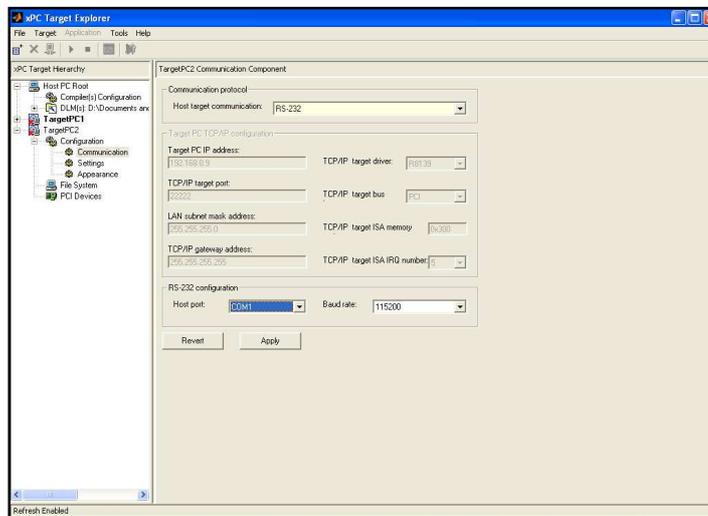


Fig. 55. Configuración. Puerto RS 232 del PC *Host*

A continuación se muestra una imagen de la comunicación en funcionamiento del PC *Host* con dos PC *Targets*, mediante Router y cables de red RJ45.



Fig. 56. Imagen tomada en el laboratorio de Proyectos. Comunicación *PC Host* con dos *PC Targets* mediante Router



7. ADQUISICIÓN DE SEÑALES

7. ADQUISICIÓN DE SEÑALES

7.1. ENVÍO DE DATOS MEDIANTE CABLE RS 232

Como primera aproximación, en el laboratorio se enviaron señales a través del puerto COM del *TargetPC1* al *TargetPC2*. Utilizando una “demo” de la *Toolbox* del *xPC Target* de *Matlab*. [13]

Según los procedimientos arriba explicados. De manera secuencial primero en el *PC Target1* se compila el siguiente modelo de *Simulink*:

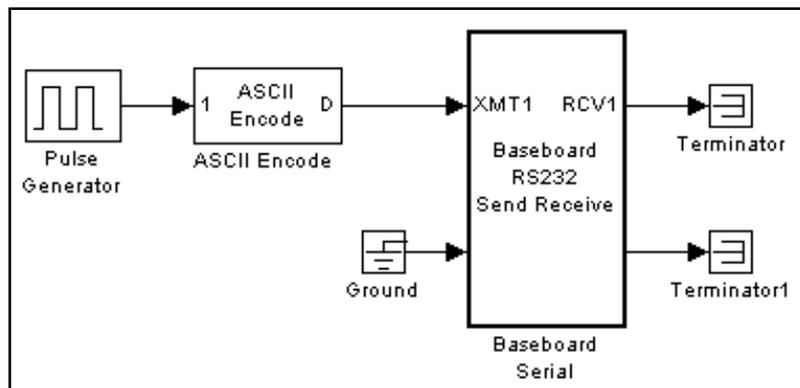


Fig. 57. Modelo en el *PC Target1* [13]

Se genera un pulso que se convierte a código ASCII y se envía por el puerto COM del *PC Target1* a través del cable serie RS 232.

Se configura el *PC Target2* por defecto, sin interrumpir el *PC Target1*, y se le compila el siguiente modelo de *Simulink*:

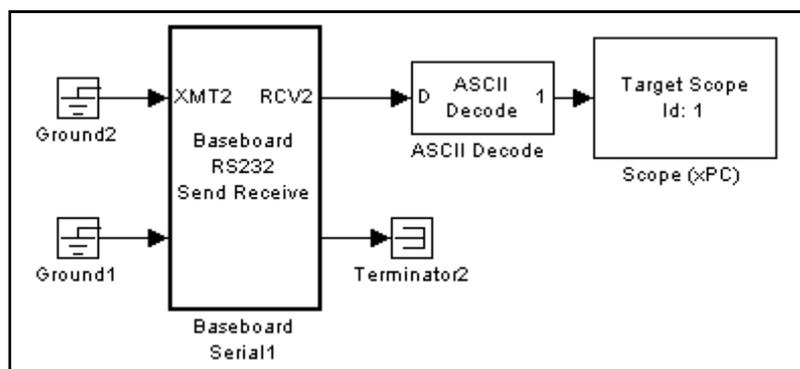


Fig. 58. Modelo en el *PC Target2* [13]

El *PC Target2* se encarga de recibir la señal que viene del *PC Target1* a través del cable serie RS-232 por el puerto COM. Decodifica el código ASCII y muestra en el osciloscopio de su pantalla los pulsos enviados por el *PC Target1*.

7.2. ENVÍO DE DATOS A TRAVÉS DE TARJETAS DE ADQUISICIÓN DE DATOS. DESDE LA TARJETA NI SCB-68 A NI SCB-100

Para estos apartados se enviará una señal que genera el PC *Target* (la cual se pasa a binario) a través de una tarjeta SCB-68 conectada en dicho PC *Target*. Dicha señal es recogida por otra tarjeta (SCB-100) conectada a dicho ordenador. Esta tarjeta reenvía la información hacia el ordenador (en donde se reconstruye a analógico) y éste la vuelve a enviar reconstruida a través de la tarjeta SCB-100 hacia el exterior (analógico).

7.2.1. Envío de un número decimal.

Primero se enviará un número entero (se ha elegido el 37), se pasará por una función que lo convierte en binario (el bloque Embedded *MATLAB* Function) y lo envía cada bit (cada número binario) por cada uno de los ocho cables que se pueden obtener como salidas digitales de la tarjeta NI. [15]

Para aclarar el esquema, se ha representado el nombre de la salida en la tarjeta NI (véase por ejemplo PFI_0 que representa el dígito más representativo, de valor 2^7 en decimal) acompañado del color del cable que en el esquema se ha utilizado.

Para este ejemplo se deben obtener los siguientes números binarios:

Número a convertir	Decimal	Binario
	2^7 128	0
	2^6 64	0
	2^5 32	1
	2^4 16	0
	2^3 8	0
	2^2 4	1
	2^1 2	0
	2^0 1	1
37		$32+4+1$

Fig. 59. Representación en binario del número 37

En la Fig. 60 que se muestra en la página siguiente se muestra el modelo de *Simulink* utilizado para la conversión en binario del número 37.

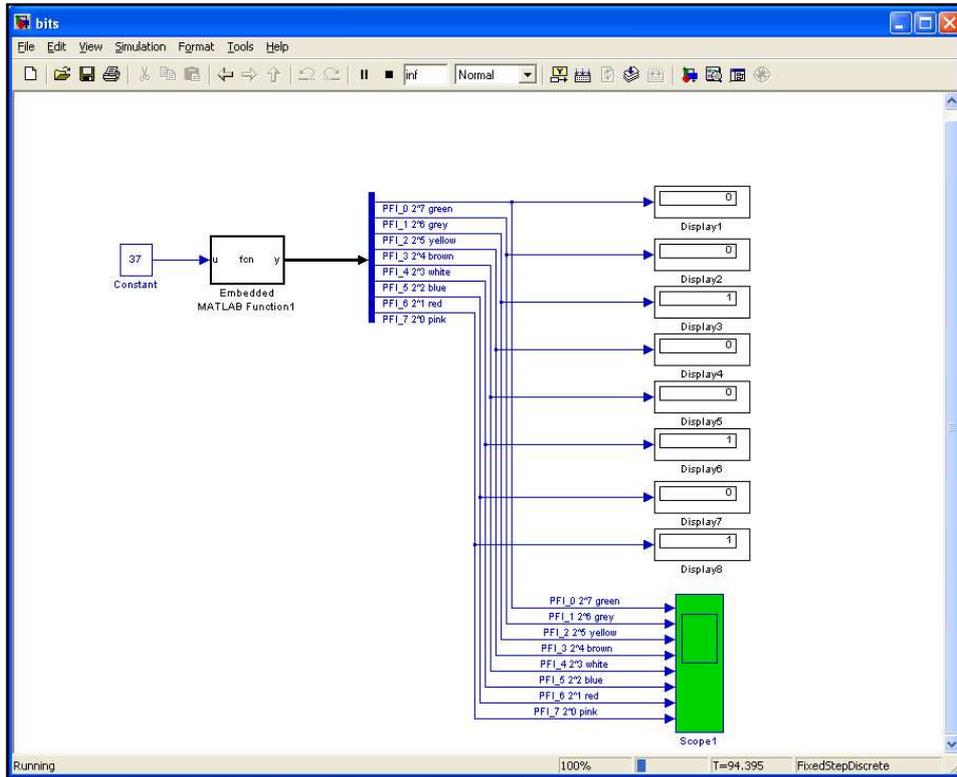


Fig. 60. Modelo para la conversión de un número decimal en binario.

La función decimal a binario se ha definido en el bloque:

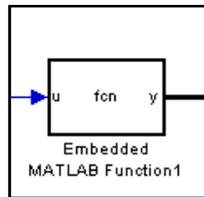


Fig. 61. Función decimal a binario

El resultado que se obtiene en el *Scope* es el siguiente:

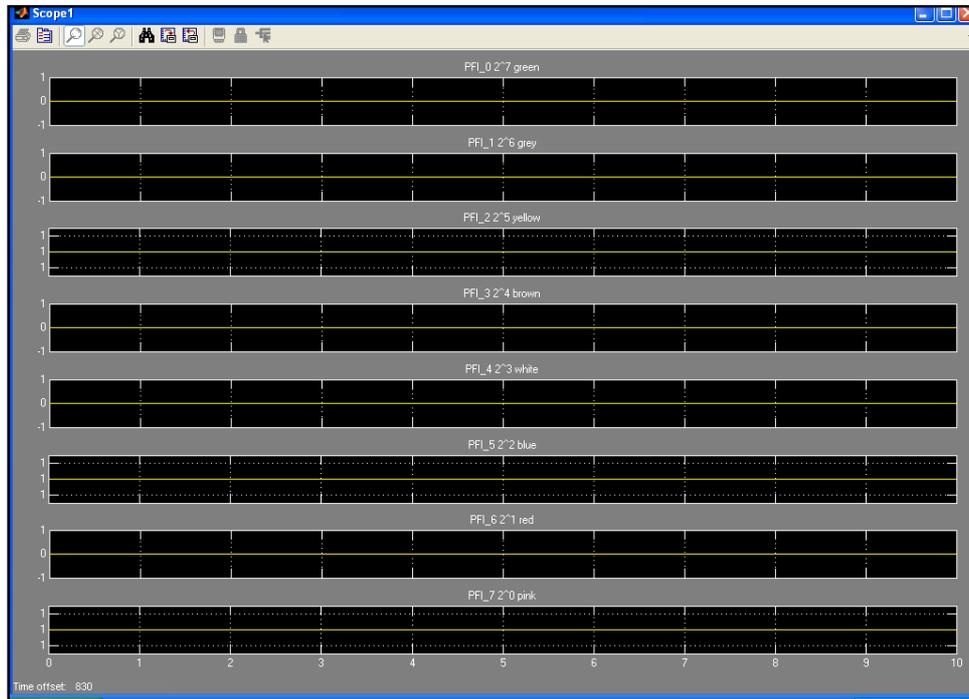


Fig. 62. *Scope1*. Representación en binario del número 37

Como se puede apreciar el número 37 en binario, es el que se busca, es el que se quiere obtener. En el laboratorio se hicieron más pruebas con diversos números y todas las representaciones eran correctas.

NOTA: Hay que tener en cuenta que con 1 byte (8 bits) como máximo se pueden representar el número 256, que es $2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0$. Por lo que la amplitud máxima de la onda que se utiliza será de 256 unidades.

7.2.2. Envío de un número decimal desde la tarjeta SCB-68 a la tarjeta SCB-100 de NI

Ahora se realizará lo anterior pero enviando el número 37 a través de las salidas digitales de la tarjeta SCB-68 de NI a las entradas digitales de la tarjeta SCB-100 de NI (Fig. 75 a 80). Y se verá el número 37 en el osciloscopio (Fig. 62 y 66).

Previamente se convertirá el valor decimal en binario, y de binario a decimal.

El modelo que se ha construido en *Simulink* para realizar este apartado ha sido el que se muestra en la Fig. 63.

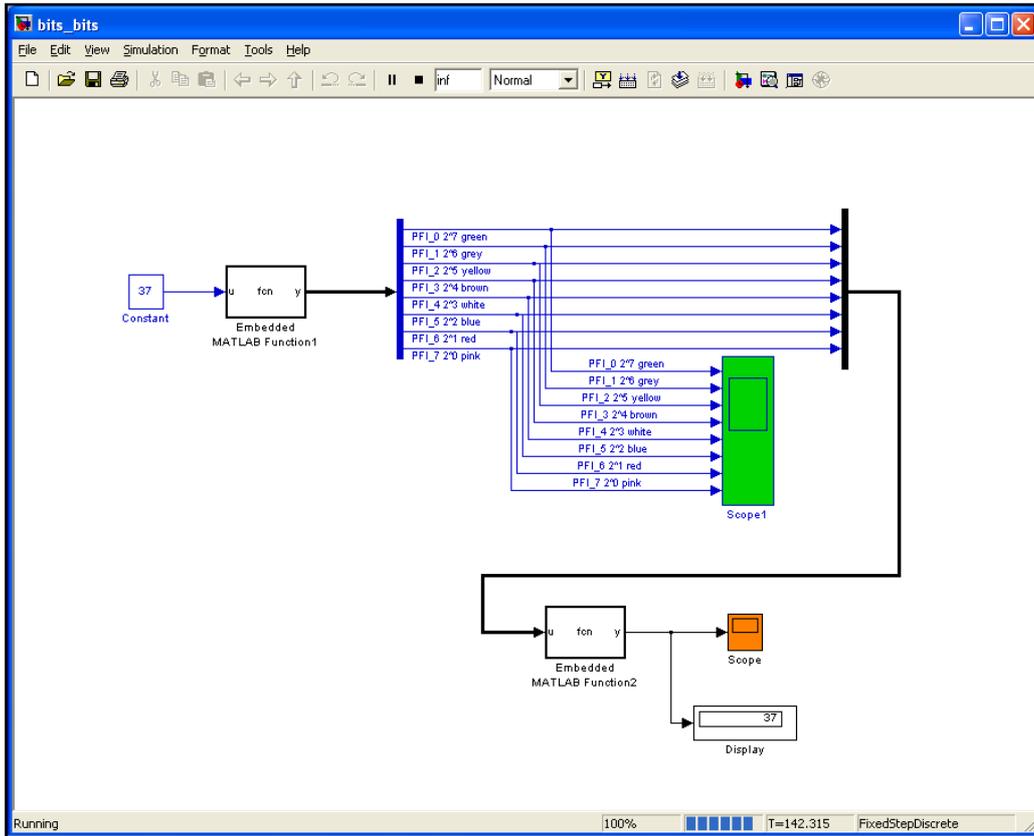


Fig. 63. Conversión de decimal a binario y de binario a decimal

Se obtiene el número 37 en el osciloscopio. También se puede ver en el Scope de salida y en el display (Fig. 63).

Seguidamente, se implementó el modelo de la Fig. 64 las tarjetas SCB-68 (envía los datos digitales) y SCB-100 (recibe los datos digitales y envía los datos analógicos) de NI.

NOTA: La tarjeta *SCB-68* corresponde a la tarjeta *PCI-6602*. La tarjeta *SCB-100* corresponde a la tarjeta *PCI-6025E*. Véase el modelo. [15]

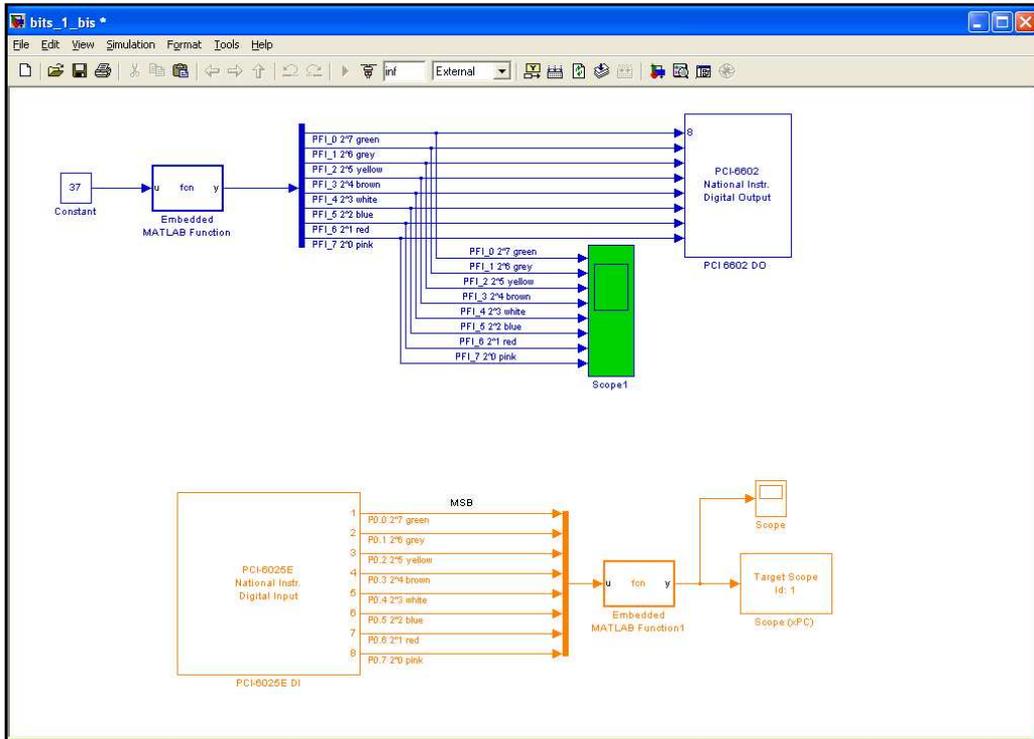


Fig. 64. Envío del número 37 desde la SCB-68 a la SCB-100

Se ha añadido un *Target Scope*, para que se muestre el número 37 también en el monitor del PC *Target* (Fig. 64).

7.2.3. Envío de una señal sinusoidal

A continuación se representará una onda sinusoidal (definida en discreto, mediante números), la cual se convertirá a binario, a través de una función como la representada en el caso anterior 7.2.2. Se enviará por las salidas digitales de la tarjeta SCB-68 de NI (Fig. 75 a 80). Y a continuación se recopilará por las entradas digitales de la tarjeta SCB-100 de NI y se le aplicará una función que represente los datos binarios a su salida. Se tomará la salida analógica de la tarjeta SCB-100 de NI para ver en el osciloscopio la salida que se obtiene, una sinusoidal exacta (Fig. 67) a la de entrada, si todo se hace correctamente.

Lo que se ha variado es que se genera una señal sinusoidal en lugar de un número. El modelo a implementar es el siguiente:

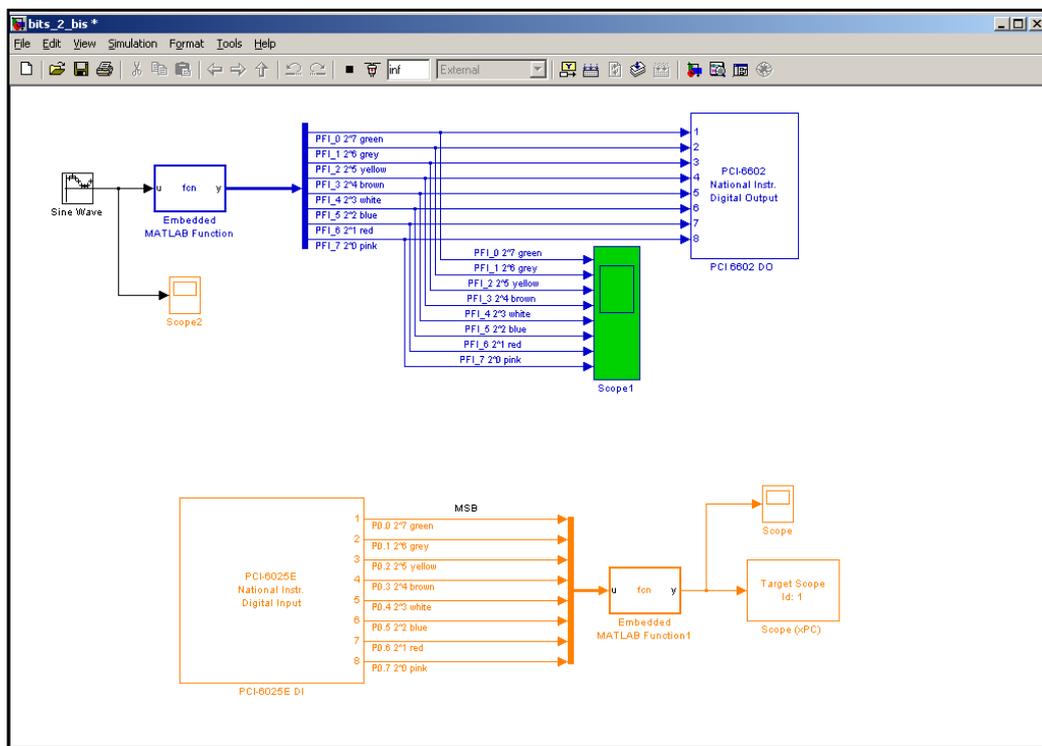


Fig. 65. Envío de onda sinusoidal desde la SCB-68 a la SCB-100

A la salida de la función primera enviado hacia las salidas digitales de la SCB-68, se obtiene:

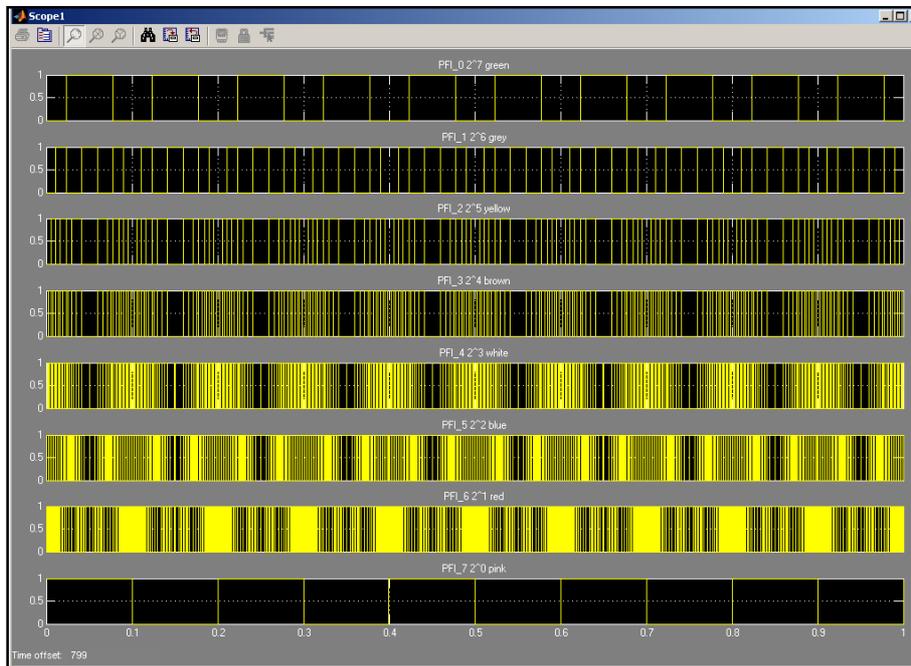


Fig. 66. Componentes de la onda sinusoidal en binario

Teniendo 1 byte (8 canales de 1 bit) se puede tener una definición de 256 muestras (2^8). Luego la máxima amplitud será de 256.

Obteniendo la onda sinusoidal en el PC Target:

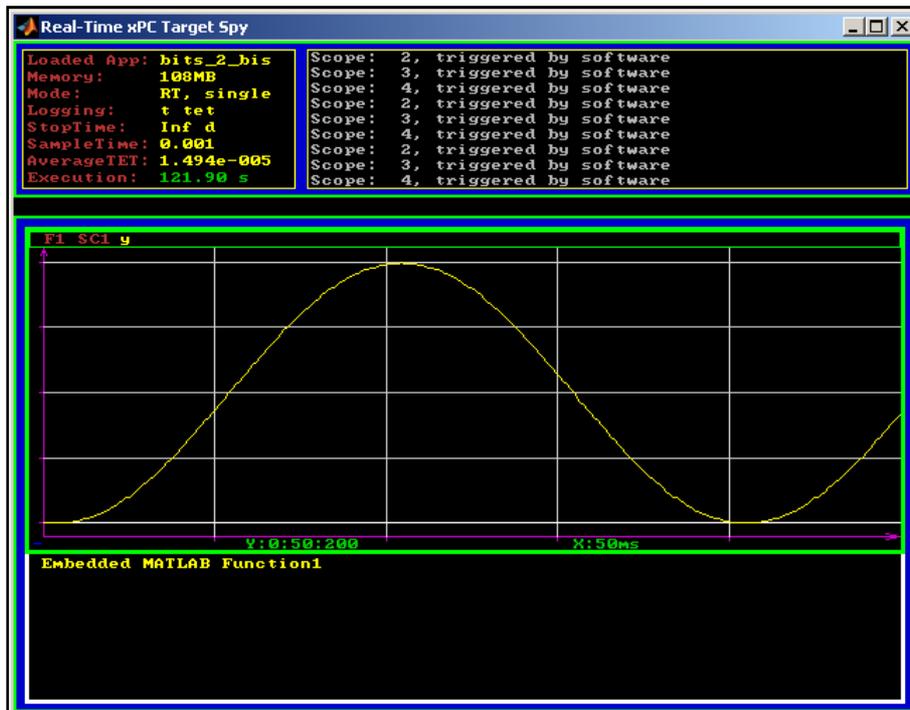


Fig. 67. Monitor del PC Target. Señal obtenida a la salida del modelo.

7.2.3.1. Dificultades durante la implementación

En un primer intento, la onda obtenida en el PC *Target* fue:

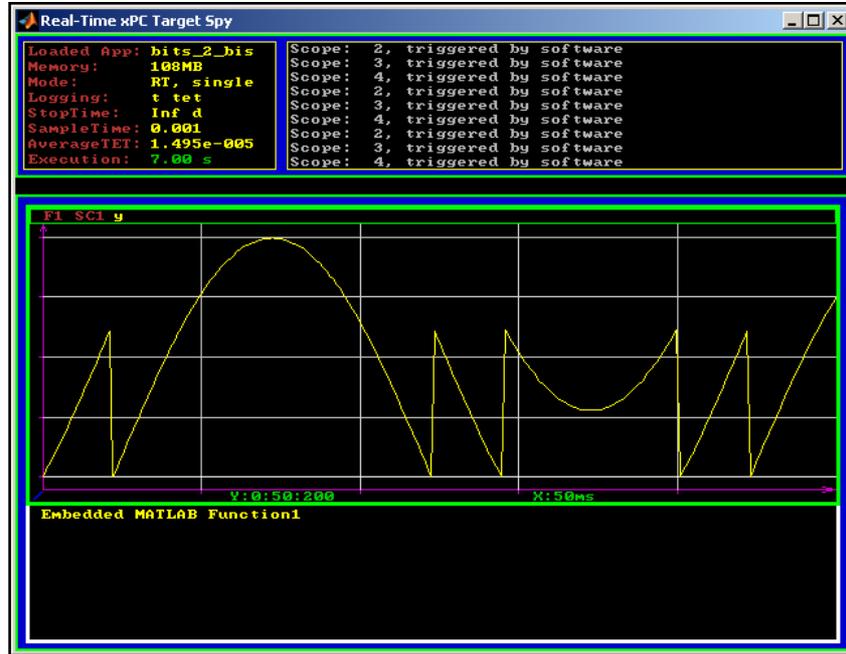


Fig. 68. Distorsión a la salida de la onda de entrada

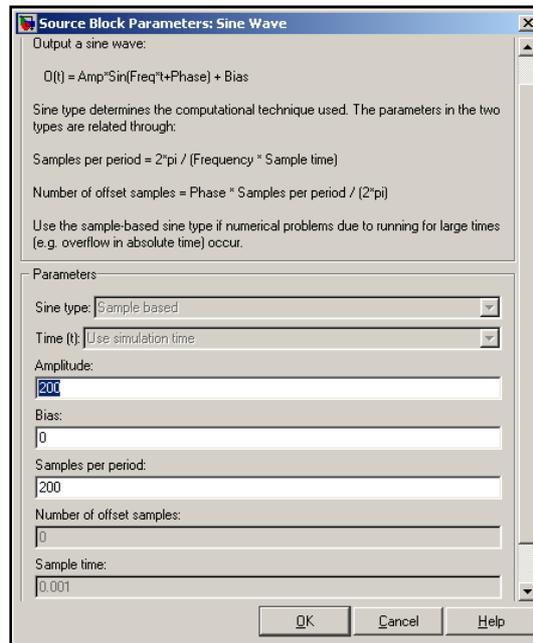


Fig. 69. Valor de Bias que da una señal desvirtuada

Esto se corrigió de dos maneras:

1. Subiendo la referencia del eje vertical.

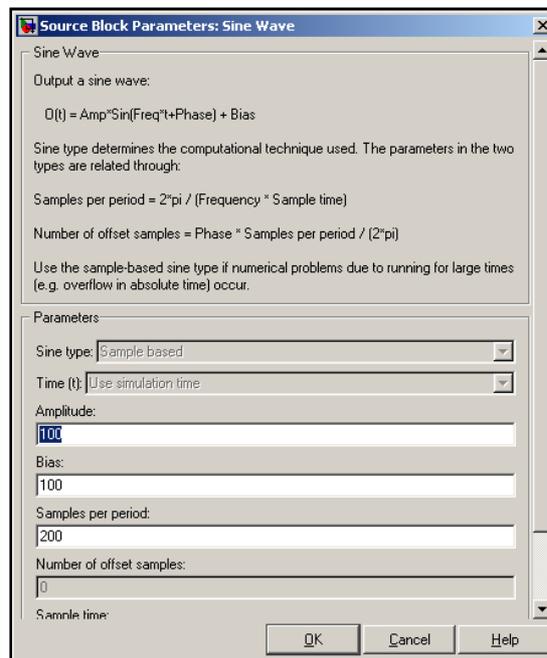


Fig. 70. Valores que dan una señal sinusoidal representada por encima del eje horizontal

2. Se puede utilizar 1 bit como signo para evitar este problema. Pero se pierde en precisión, porque con esta solución sólo se podría utilizar 7 bits (2^7) para representar la onda y 1 bit para el signo.

La implementación en este caso, nos obliga a cambiar las funciones anteriormente escritas, para pasar de decimal a binario y de binario a decimal.

Ahora se tiene una precisión de 128 valores pero se gana en que la onda se puede representar con valores negativos, se encuentra atravesando el eje horizontal (pasa por cero).

7.2.3.2. Obtención de las gráficas en el osciloscopio

Se implementa un bloque para obtener las salidas analógicas de la NI SCB-100. Se conectan a tierra las tierras de los dos canales del osciloscopio, a AO GND de la tarjeta SCB-100 y se obtienen las salidas AO 0 y AO 1 (Fig. 79 y 80).

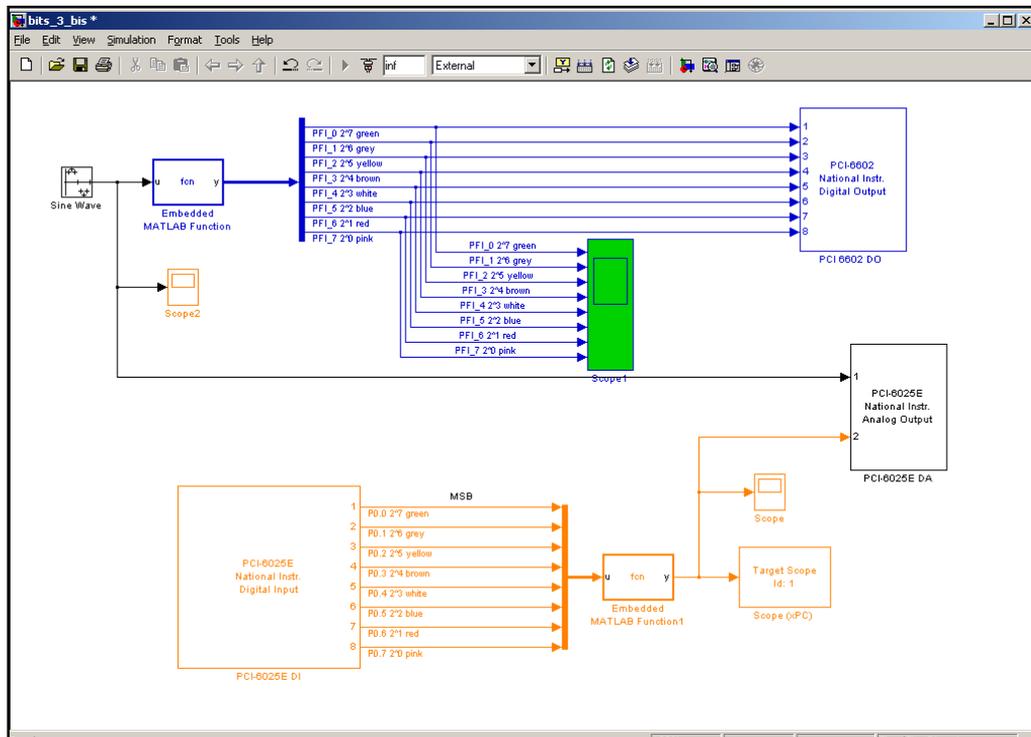


Fig. 71. Envío de onda sinusoidal desde la SCB-68 a la SCB-100. Con salida analógica de la SCB-100

En este ensayo se obtenía también la señal sinusoidal desvirtuada en el osciloscopio, que es la misma que la medida en el PC *Target*.

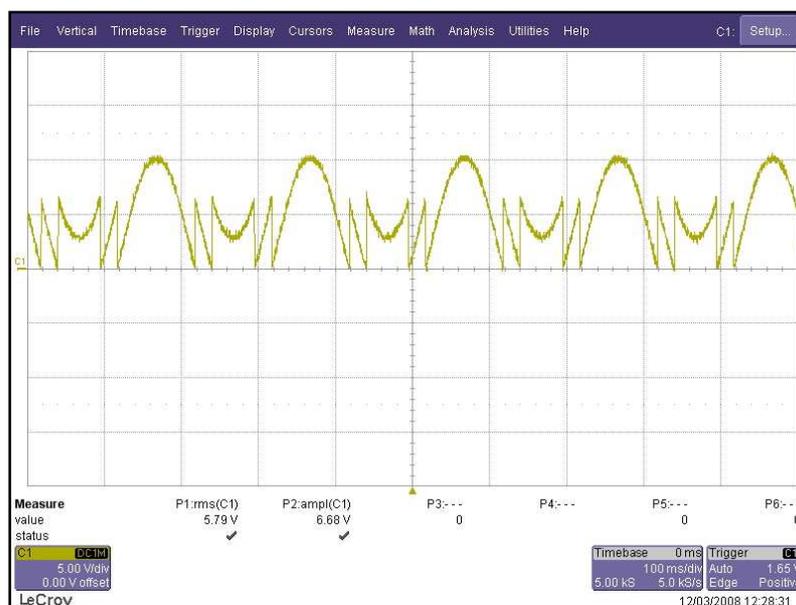


Fig. 72. Osciloscopio. Salida analógica de la SCB-100. Señal desvirtuada

Corrigiendo lo anterior, poniendo una ganancia que evitó esta distorsión se obtuvo la señal de la manera correcta.

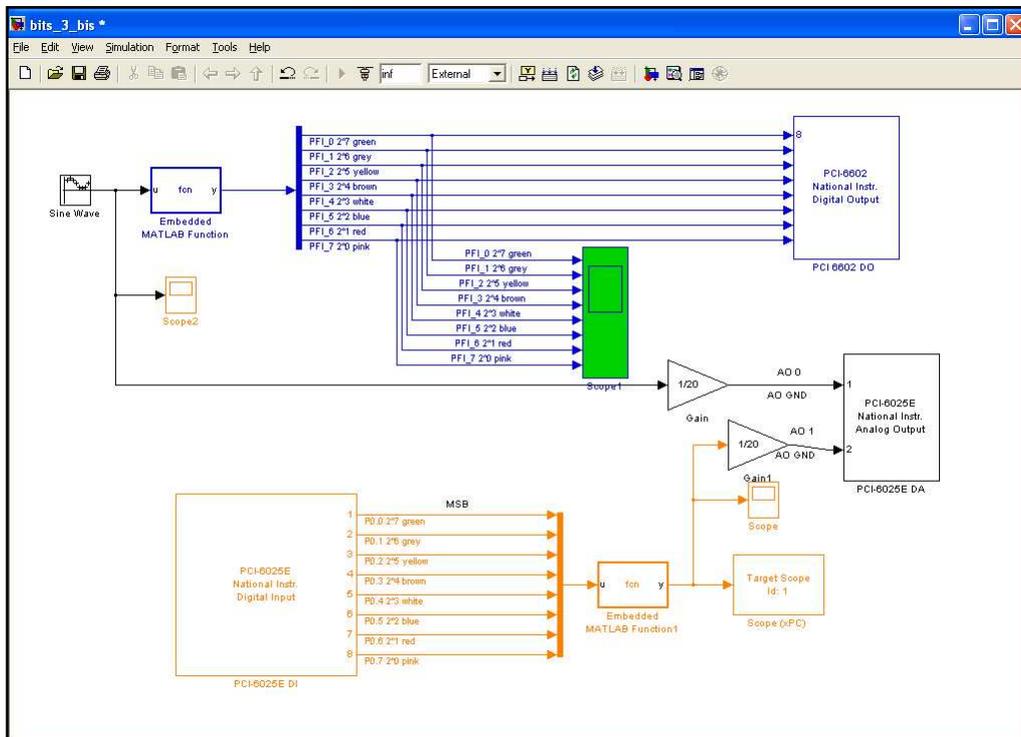


Fig. 73. Envío de onda sinusoidal desde la SCB-68 a la SCB-100. Con salida analógica de la SCB-100 y con ganancias correctoras

Por lo que la señal que se obtiene en el osciloscopio es la que se buscaba:

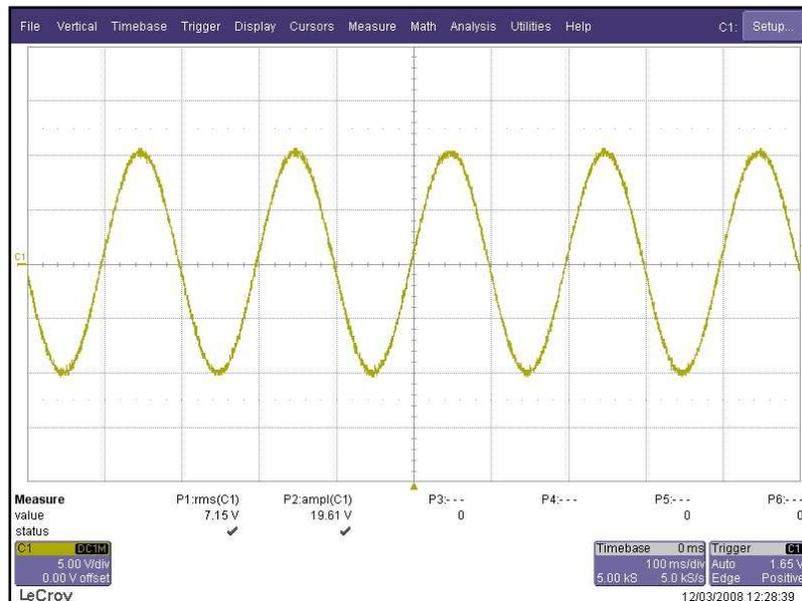


Fig. 74. Osciloscopio. Salida analógica de la SCB-100. Onda sinusoidal con 8 bits con ganancias correctoras

7.2.3.3. Esquemas de conexión en las tarjetas SCB de NI. Cableado, cable de 8 hilos

SCB-68 salidas (entradas) digitales [15]

SCB-68 Quick Reference Label NI 660X DEVICES			
PIN#		SIGNAL	
68	GND	12	PFI_3
34	PFI_31 (SOURCE_2)	46	GND
67	PFI_30 (GATE_2)	13	PFI_4
33	GND	47	PFI_5
66	PFI_29 (UP_DOWN_2)	14	GND
32	PFI_28 (OUT_2)	48	PFI_6
65	GND	15	PFI_7
31	PFI_27 (SOURCE_3)	49	GND
64	PFI_26 (GATE_3)	16	PFI_8 (OUT_7)
30	GND	50	GND
63	PFI_25 (UP_DOWN_3)	17	PFI_9 (UP_DOWN_7)
29	PFI_24 (OUT_3)	51	PFI_10 (GATE_7)
62	GND	18	GND
28	PFI_23 (SOURCE_4)	52	PFI_11 (SOURCE_7)
61	PFI_22 (GATE_4)	19	RG
27	GND	53	PFI_12 (OUT_6)
60	PFI_21 (UP_DOWN_4)	20	GND
26	PFI_20 (OUT_4)	54	PFI_13 (UP_DOWN_6)
59	GND	21	PFI_14 (GATE_6)
25	PFI_19 (SOURCE_5)	55	GND
58	PFI_18 (GATE_5)	22	PFI_15 (SOURCE_6)
24	GND	56	RG
57	PFI_17 (UP_DOWN_5)		
23	PFI_16 (OUT_5)		
		1	+5V
		35	RG
		2	PFI_39 (SOURCE_0)
		36	GND
		3	PFI_38 (GATE_0)
		37	RESERVED
		4	RESERVED
		38	RESERVED
		5	PFI_36 (OUT_0)
		39	GND
		6	PFI_33 (UP_DOWN_1)
		40	PFI_37 (UP_DOWN_0)
		7	PFI_35 (SOURCE_1)
		41	GND
		8	PFI_34 (GATE_1)
		42	GND
		9	PFI_32 (OUT_1)
		43	RG
		10	PFI_0
		44	PFI_1
		11	GND
		45	PFI_2

Fig. 75. NI SCB-68 PCI 6602. Esquema cableado digital

A las salidas digitales de la NI SCB-68 (PFI_0, PFI_1, PFI_2, PFI_3, PFI_4, PFI_5, PFI_6, PFI_7) se les atornillan cables con los siguientes colores:

(PFI_0, PFI_1, PFI_2, PFI_3, PFI_4, PFI_5, PFI_6, PFI_7)

Estos cables tienen su otro extremo atornillado a las entradas digitales de la tarjeta SCB-100.

(PO_0, PO_1, PO_2, PO_3, PO_4, PO_5, PO_6, PO_7)

La conexión del cableado sobre la tarjeta, queda entonces:

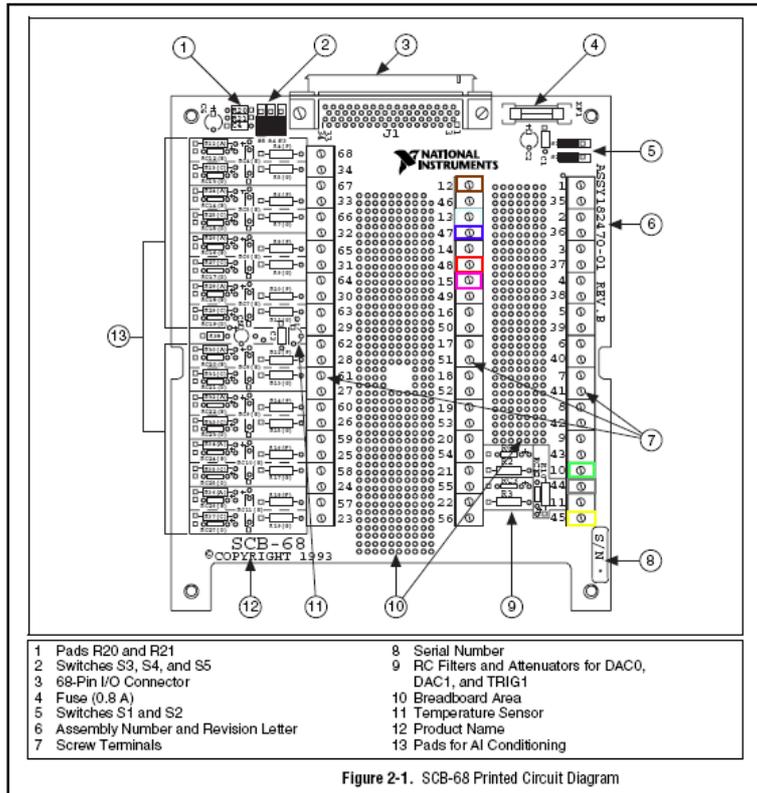


Fig. 76. NI SCB-68 PCI 6602. Esquema cableado digital sobre tarjeta

SCB-100 entradas (salidas) digitales

SCB-100 QUICK REFERENCE LABEL		PIN # SIGNAL		33 D GND		67 AI 32 (P2.7)	
 E-SERIES DEVICES EXTENDED ANALOG (EXTENDED DIGITAL) PIN 183022B-01 FACTORY DEFAULT SETTING S1: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S2: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S3: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S6 S5 S4 * TEMP SENSOR DISABLED * ACCESSORY POWER ON S1: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S2: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S3: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S6 S5 S4 * TEMP SENSOR ENABLED ON SINGLE ENDED CH 0 * ACCESSORY POWER ON S1: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S2: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S3: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S6 S5 S4 * TEMP SENSOR ENABLED ON DIFFERENTIAL CH 0 * ACCESSORY POWER ON S1: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S2: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S3: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> S6 S5 S4 * 100 GENERIC TERMINALS (TEMP. SENSOR AND ACCESSORY POWER OFF)		1	AI GND	34	+5 V	68	AI 40 (D GND)
		2	AI GND	35	+5 V	69	AI 33 (P2.8)
		3	AI 0	36	AI HOLD COMP	70	AI 41 (D GND)
		4	AI 8	37	EXT STROBE	71	AI 34 (P2.5)
		5	AI 1	38	PFI 0/AI START TRIG	72	AI 42 (D GND)
		6	AI 9	39	PFI 1/AI REF TRIG	73	AI 35 (P2.4)
		7	AI 2	40	PFI 2/AI CONV CLK	74	AI 43 (D GND)
		8	AI 10	41	PFI 3/CTR 1 SOURCE	75	AI SENSE 2 (P2.3)
		9	AI 3	42	PFI 4/CTR 1 GATE	76	AI GND (D GND)
		10	AI 11	43	CTR 1 OUT	77	AI 36 (P2.2)
		11	AI 4	44	PFI 5/AO SAMP CLK	78	AI 44 (D GND)
		12	AI 12	45	PFI 6/AO START TRIG	79	AI 37 (P2.1)
		13	AI 5	46	PFI 7/AI SAMP CLK	80	AI 45 (D GND)
14	AI 13	47	PFI 8/CTR 0 SOURCE	81	AI 38 (P2.0)		
15	AI 6	48	PFI 9/CTR 0 GATE	82	AI 46 (D GND)		
16	AI 14	49	CTR 0 OUT	83	AI 39 (P1.7)		
17	AI 7	50	FREQ OUT	84	AI 47 (D GND)		
18	AI 15	51	AI 16 (P3.7)	85	AI 48 (P1.8)		
19	AI SENSE	52	AI 24 (D GND)	86	AI 56 (D GND)		
20	AO 0	53	AI 17 (P3.6)	87	AI 49 (P1.5)		
21	AO 1	54	AI 25 (D GND)	88	AI 57 (D GND)		
22	AO EXT REF	55	AI 18 (P3.5)	89	AI 50 (P1.4)		
23	AO GND	56	AI 26 (D GND)	90	AI 58 (D GND)		
NC		57	AI 19 (P3.4)	91	AI 51 (P1.3)		
24	D GND	58	AI 27 (D GND)	92	AI 59 (D GND)		
25	PO.0	59	AI 20 (P3.3)	93	AI 52 (P1.2)		
26	PO.4	60	AI 28 (D GND)	94	AI 60 (D GND)		
27	PO.1	61	AI 21 (P3.2)	95	AI 53 (P1.1)		
28	PO.5	62	AI 29 (D GND)	96	AI 61 (D GND)		
29	PO.2	63	AI 22 (P3.1)	97	AI 54 (P1.0)		
30	PO.6	64	AI 30 (D GND)	98	AI 62 (D GND)		
31	PO.3	65	AI 23 (P3.0)	99	AI 55 (+5 V)		
32	PO.7	66	AI 31 (D GND)	100	AI 63 (D GND)		

Fig. 77. NI SCB-100 PCI 6625E. Esquema cableado digital

El cableado de la tarjeta SCB-100 queda como se muestra en la Fig. 79:

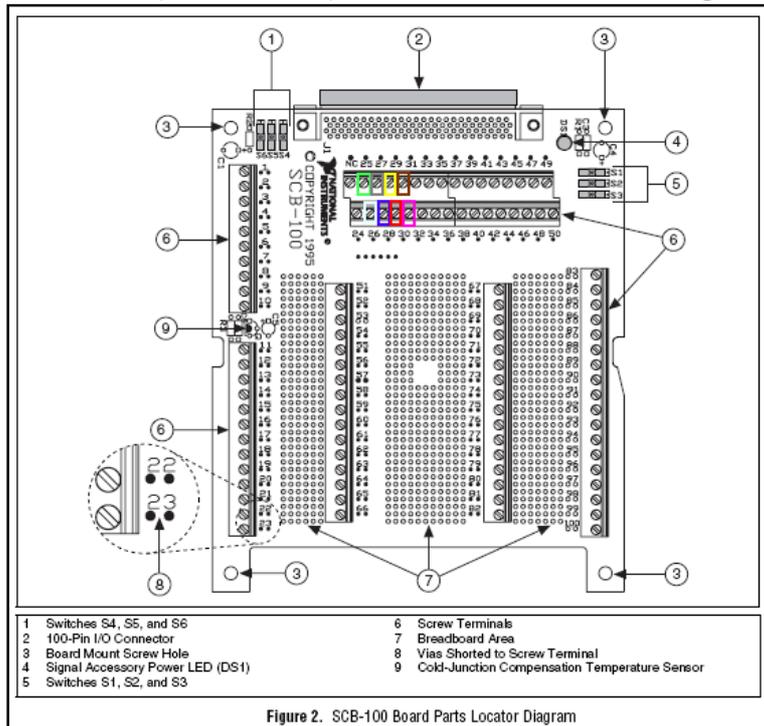


Fig. 78. NI SCB-100 PCI 6625E. Esquema cableado digital sobre tarjeta

SCB-100 salidas analógicas

SCB-100 QUICK REFERENCE LABEL		NATIONAL INSTRUMENTS		E-SERIES DEVICES EXTENDED ANALOG (EXTENDED DIGITAL)	
P/N 183022B-01		FACTORY DEFAULT SETTING		S1 S2 S3	
PIN #	SIGNAL	33	D GND	67	AI 32 (P2.7)
1	AI GND	34	+5 V	68	AI 40 (D GND)
2	AI GND	35	+5 V	69	AI 33 (P2.8)
3	AI 0	36	AI HOLD COMP	70	AI 41 (D GND)
4	AI 8	37	EXT STROBE	71	AI 34 (P2.5)
5	AI 1	38	PFI 0/AI START TRIG	72	AI 42 (D GND)
6	AI 9	39	PFI 1/AI REF TRIG	73	AI 35 (P2.4)
7	AI 2	40	PFI 2/AI CONV CLK	74	AI 43 (D GND)
8	AI 10	41	PFI 3/CTR 1 SOURCE	75	AI SENSE 2 (P2.3)
9	AI 3	42	PFI 4/CTR 1 GATE	76	AI GND (D GND)
10	AI 11	43	CTR 1 OUT	77	AI 36 (P2.2)
11	AI 4	44	PFI 5/AO SAMP CLK	78	AI 44 (D GND)
12	AI 12	45	PFI 6/AO START TRIG	79	AI 37 (P2.1)
13	AI 5	46	PFI 7/AI SAMP CLK	80	AI 45 (D GND)
14	AI 13	47	PFI 8/CTR 0 SOURCE	81	AI 38 (P2.0)
15	AI 6	48	PFI 9/CTR 0 GATE	82	AI 46 (D GND)
16	AI 14	49	CTR 0 OUT	83	AI 39 (P1.7)
17	AI 7	50	FREQ OUT	84	AI 47 (D GND)
18	AI 15	51	AI 16 (P3.7)	85	AI 48 (P1.8)
19	AI SENSE	52	AI 24 (D GND)	86	AI 56 (D GND)
20	AO 0	53	AI 17 (P3.6)	87	AI 49 (P1.5)
21	AO 1	54	AI 25 (D GND)	88	AI 57 (D GND)
22	AO EXT REF	55	AI 18 (P3.5)	89	AI 50 (P1.4)
23	AO GND	56	AI 26 (D GND)	90	AI 58 (D GND)
NC		57	AI 19 (P3.4)	91	AI 51 (P1.3)
24	D GND	58	AI 27 (D GND)	92	AI 59 (D GND)
25	PD.0	59	AI 20 (P3.3)	93	AI 52 (P1.2)
26	PD.4	60	AI 28 (D GND)	94	AI 60 (D GND)
27	PD.1	61	AI 21 (P3.2)	95	AI 53 (P1.1)
28	PD.5	62	AI 29 (D GND)	96	AI 61 (D GND)
29	PD.2	63	AI 22 (P3.1)	97	AI 54 (P1.0)
30	PD.6	64	AI 30 (D GND)	98	AI 62 (D GND)
31	PD.3	65	AI 23 (P3.0)	99	AI 55 (+5 V)
32	PD.7	66	AI 31 (D GND)	100	AI 63 (D GND)

Fig. 79. NI SCB-100 PCI 6625E. Esquema cableado analógico

Las salidas analógicas se conectan a los canales 1 y 2 del osciloscopio.

Las tierras de ambos canales se conectan mediante sus “cocodrilos” a la tierra de la salida analógica de la SCB-100 (AO GND)

El canal 1 que representará nuestra entrada (la onda sinusoidal en este caso) conectará su otro terminal de dicho canal a la salida 1 analógica de la SCB-100 (AO 0).

El canal 2 que representará la salida (la onda sinusoidal que esperamos de la salida de la SCB-100) conectará su otro terminal de dicho canal a la salida 2 analógica de la SCB-100 (AO 1)

El cableado de las salidas analógicas de la tarjeta SCB-100 queda finalmente:

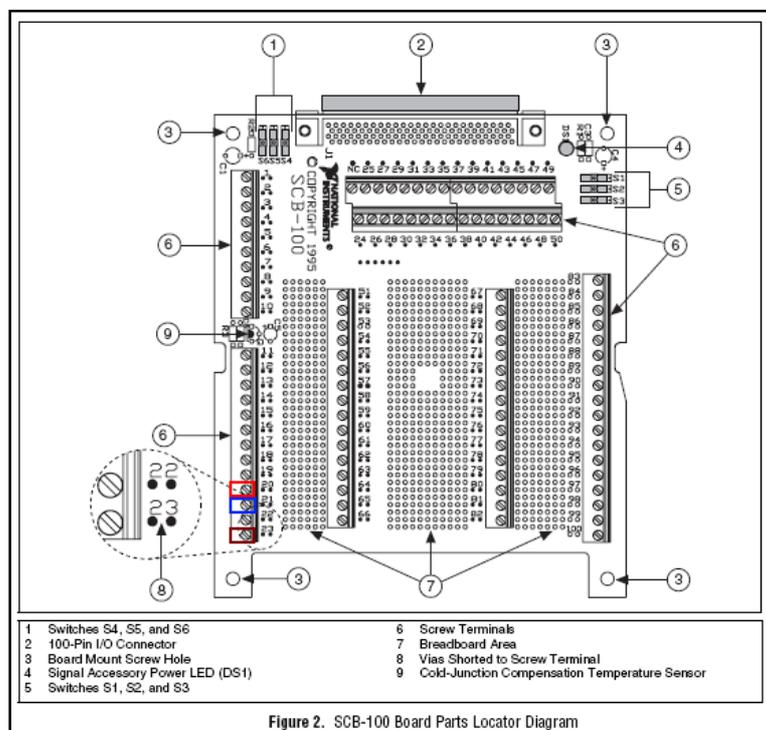


Fig. 80. NI SCB-100 PCI 6625E. Esquema cableado analógico sobre tarjeta

7.2.3.4. Retardos en el envío. *xPC Target*, SCB-68 y SCB-100 en el mismo PC Target

Con el modelo utilizado del envío de la señal sinusoidal, según el esquema:

xPC Target -> SCB-68 -> *xPC Target* -> SCB-100 -> osciloscopio

Estableciendo una amplitud de de 100 y 20 muestras por periodo. Se realizaron diversos ensayos comprobando que la amplitud si se excedía de 255 no representaba bien la señal (por la precisión de 256 antes expuesta) y que las muestras por periodo no afectaban al retardo siempre y cuando fuese mayor que el periodo de cálculo.

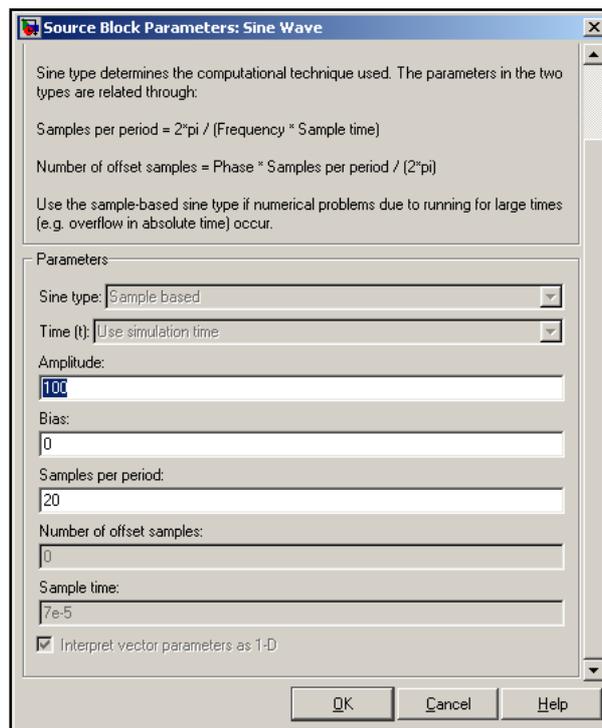


Fig. 81. Parámetros de la señal sinusoidal

Además se fijó el tiempo de muestreo en $3.5e-5s$, debido a que tras varias pruebas se comprobó que el PC Target se colapsaba (aparecía en el monitor del PC Target overload) para tiempos menores, dejaba de calcular y ejecutar el modelo compilado.

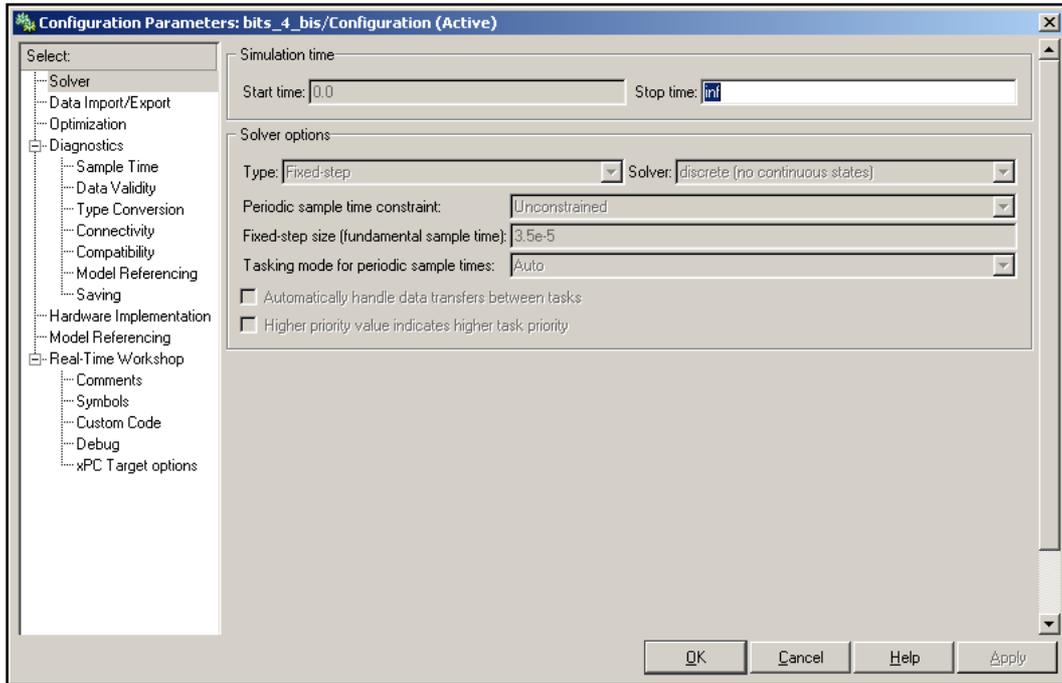


Fig. 82. Tiempos de simulación.

Se midieron en el osciloscopio la señal de salida de la SCB-68 y la señal de salida de la SCB-100 midiendo el desfase.

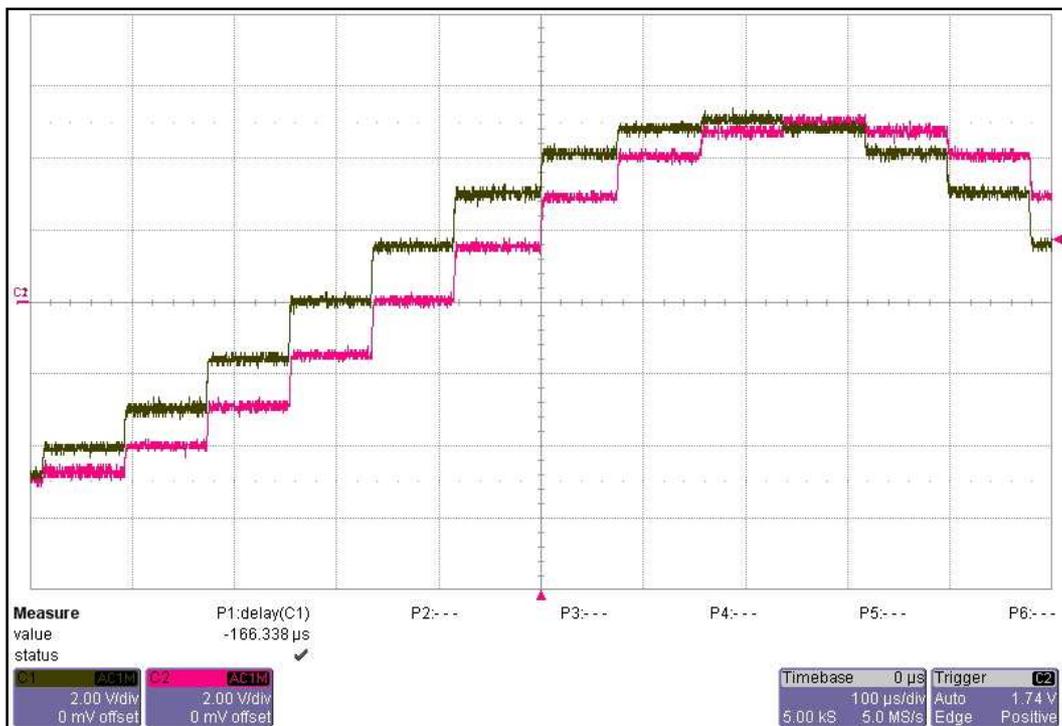


Fig. 83. Retardo del sistema

El mayor desfase fue de 100μ s, por lo que el hardware implementado sería capaz de actuar para tiempos menores que este retardo.

7.2.3.5. Retardos en el envío. xPC Target, generador de funciones SCB-100

Para comprobar cuál era el retardo real se realizó una nueva prueba. En la que se utilizó un generador de funciones y las entradas y salidas analógicas de la tarjeta SCB-100.

Generador funciones -> (entrada analógica) SCB-100 -> xPC Target -> (salida analógica) SCB-100 -> osciloscopio

Entrada analógica (AI 0) de la tarjeta SCB-100. Se le introduce una sinusoidal que viene del generador de funciones. Esta señal va al ordenador y sale por la salida analógica de la SCB-100. Ambas señales se llevan a dos canales del osciloscopio, para poder medir el desfase.

El retardo del sistema se midió con el siguiente modelo:

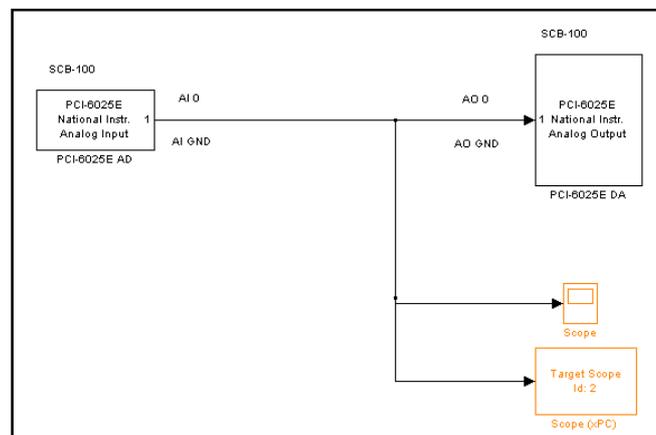


Fig. 84. Entradas y salidas analógicas de la tarjeta SCB-100

Para la conexión, ver los esquemas de las (Fig. 75 a 80) en donde se indican los pines de las entradas (AI 0, AI GND. Pines 3 y 1 respectivamente) y salidas (AO 0, AO GND. Pines 20 y 23 respectivamente) analógicas.

Los ensayos de la prueba corroboraban que el desfase del sistema era de 100µs. Este tiempo se fijaba en la aplicación de *Matlab* debido a que con tiempos menores el PC *Target* se colapsaba deteniendo la ejecución del sistema de control. Se muestran a continuación representaciones de la señal de entrada (en verde) del generador de funciones. Y la señal de salida analógica (en marrón) de la tarjeta SCB-100.

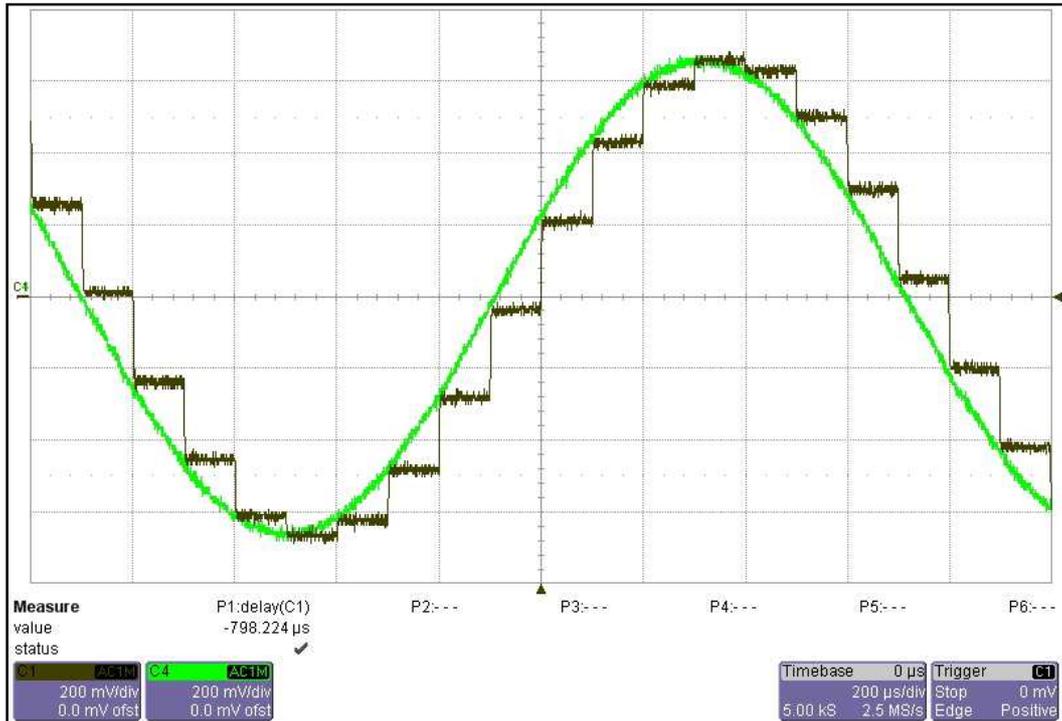


Fig. 85. Señal de entrada (generador de funciones); señal salida analógica (SCB-100)

Para un tiempo de muestreo de 100μs se obtiene una calidad de onda de 100μs.

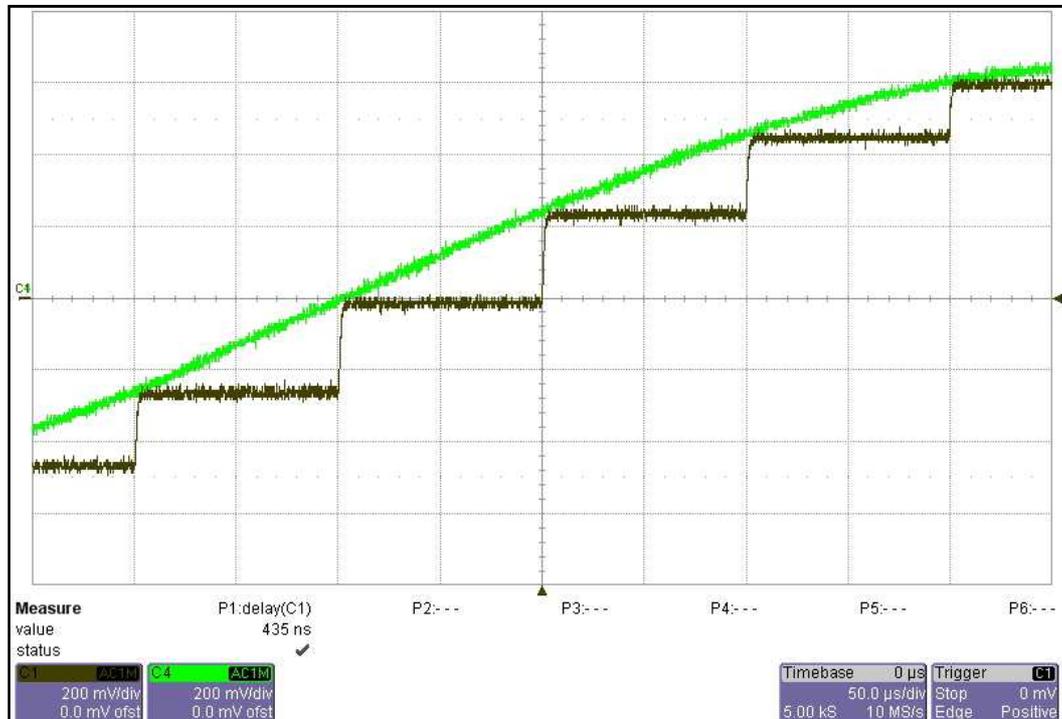


Fig. 86. Ampliación. Señal de entrada (generador de funciones); señal salida analógica (SCB-100)

En el monitor del PC *Target* se muestra un valor denominado TET. Este valor es el tiempo de ejecución de tarea. Es el tiempo que realmente el PC *Target* tarda en

realizar los cálculos exigidos. Mientras que los $100\mu\text{s}$ es un retardo intrínseco de la aplicación *xPC Target*, digamos un tiempo de refresco de datos del sistema.

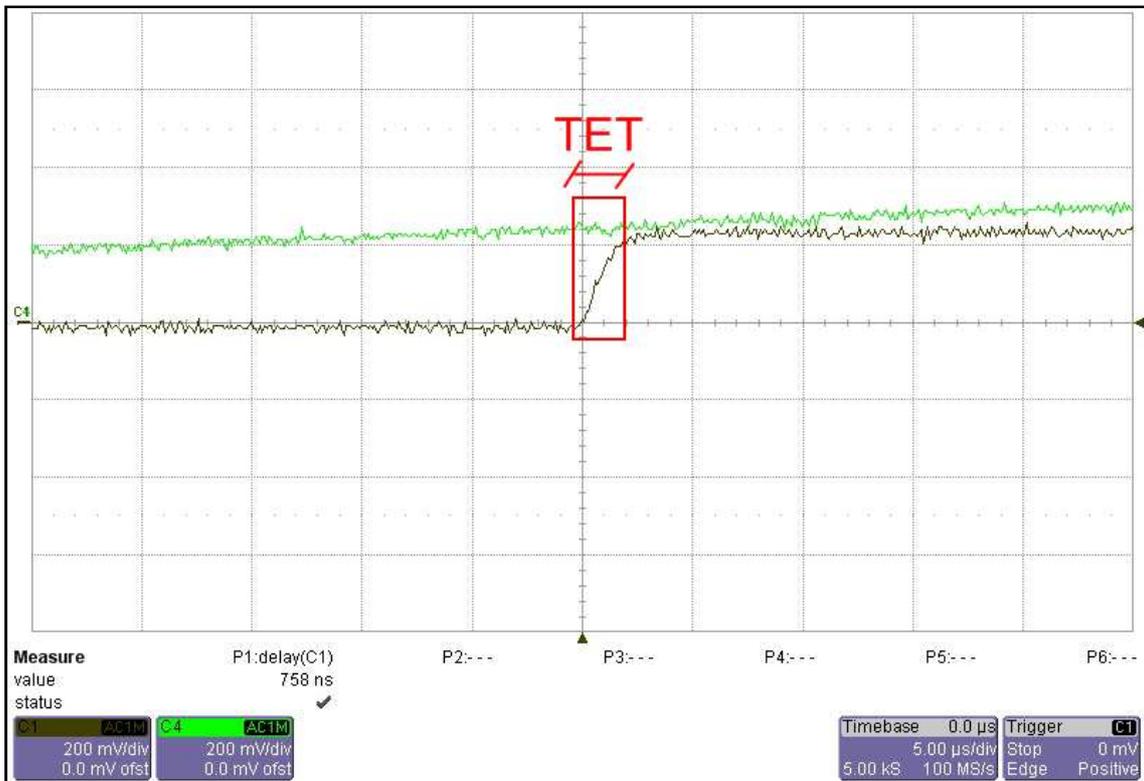


Fig. 87. Medida del Tiempo de Ejecución de Tarea (TET) de la aplicación *xPC Target*

En el osciloscopio se midió un TET de $1,5\text{e-}5\text{s}$. Mientras que en el monitor del *PC Target* se obtuvo $1,494\text{e-}5\text{s}$. Este valor corresponde al *AverageTET* de la aplicación *xPC Target*.

7.2.3.6. ENVÍO DE DATOS POR EL PROTOCOLO UDP A TRAVÉS DE CABLE DE RED RJ45

UDP es un protocolo de envío de datos que no tiene confirmación, ni control de flujo, como sí tiene el protocolo TCP/IP, por lo que los paquetes pueden adelantarse unos a otros; y tampoco se sabe si ha llegado correctamente, ya que no hay confirmación de entrega o recepción.

Se realizó la prueba de envío de datos a través del propio cable de red RJ45. Que además de utilizarse para la comunicación entre el PC *Host* y los PC *Targets*, se utilizó mediante protocolo UDP (similar al TCP/IP) para el envío de datos. Utilizando dicho cable de red en lugar de las tarjetas de adquisición de datos de *National Instruments*.

En el PC *Target1* se compila un modelo que genera una señal sinusoidal que se convierte a binario y que se envía a través del cable de red por protocolo UDP hacia el exterior.

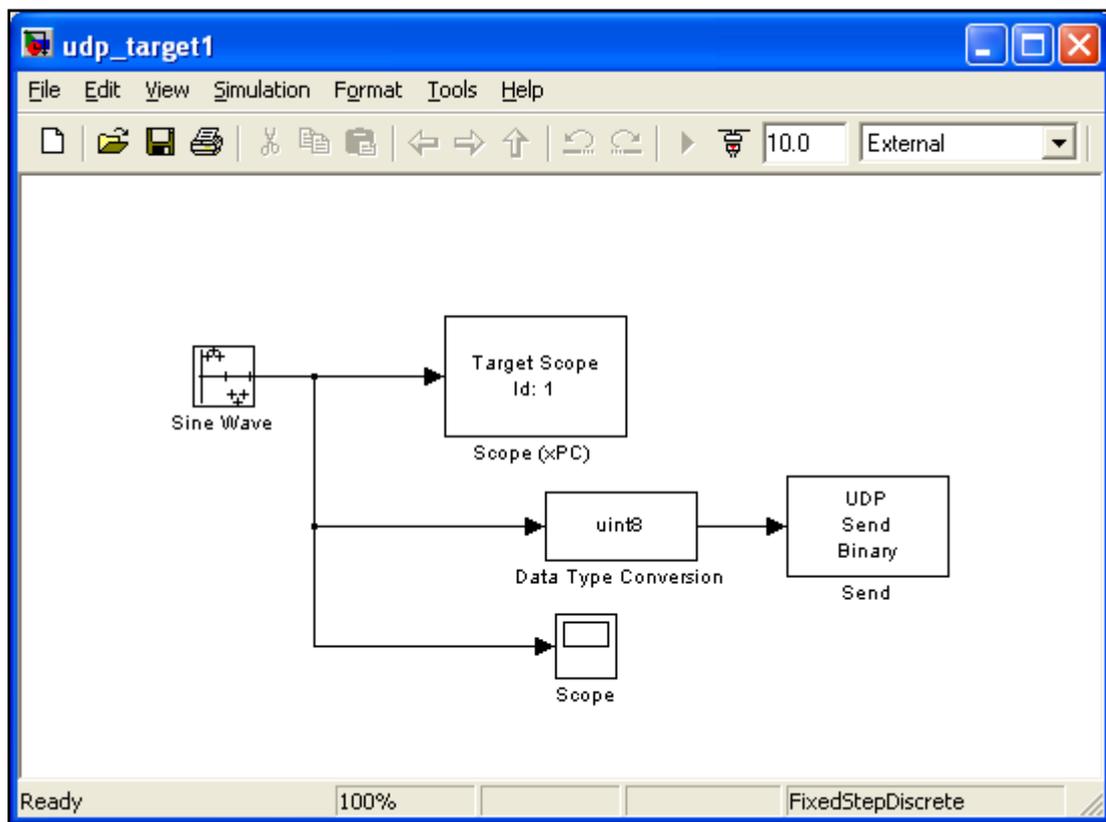


Fig. 88. PC Target1. Señal Sinusoidal en binario enviada por protocolo UDP

En el monitor del PC *Target1* se puede ver la siguiente imagen:

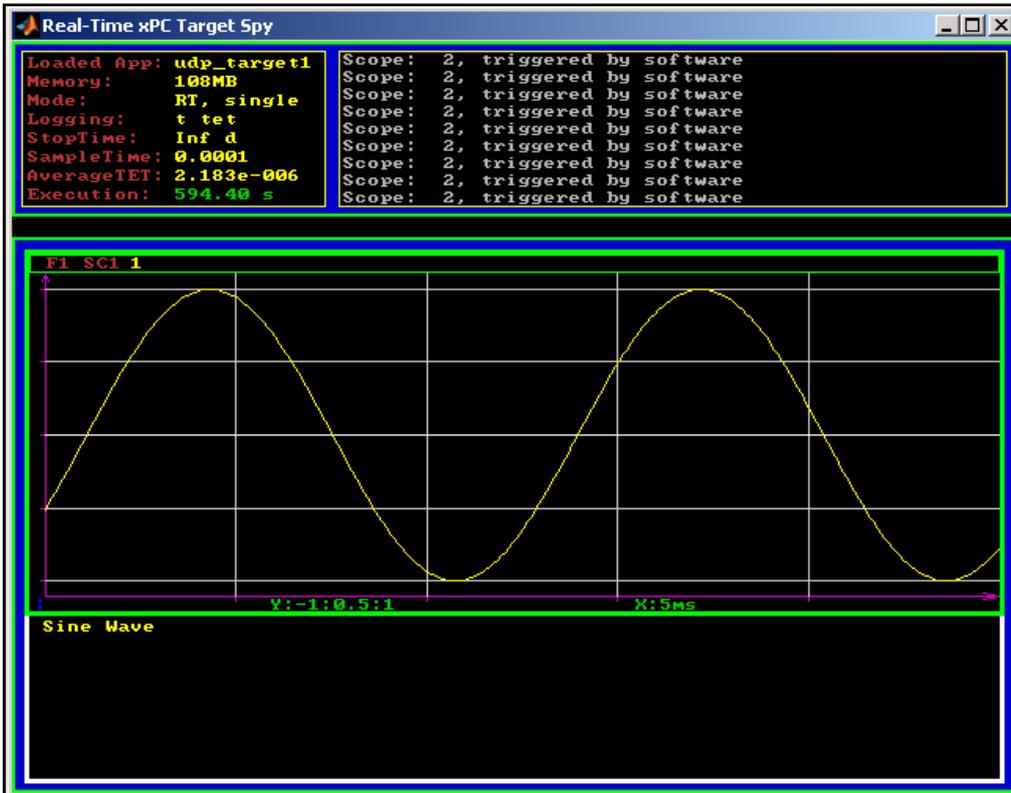


Fig. 89. PC *Target1*. Captura de imagen del monitor

En el PC *Target2* se compila el siguiente modelo:

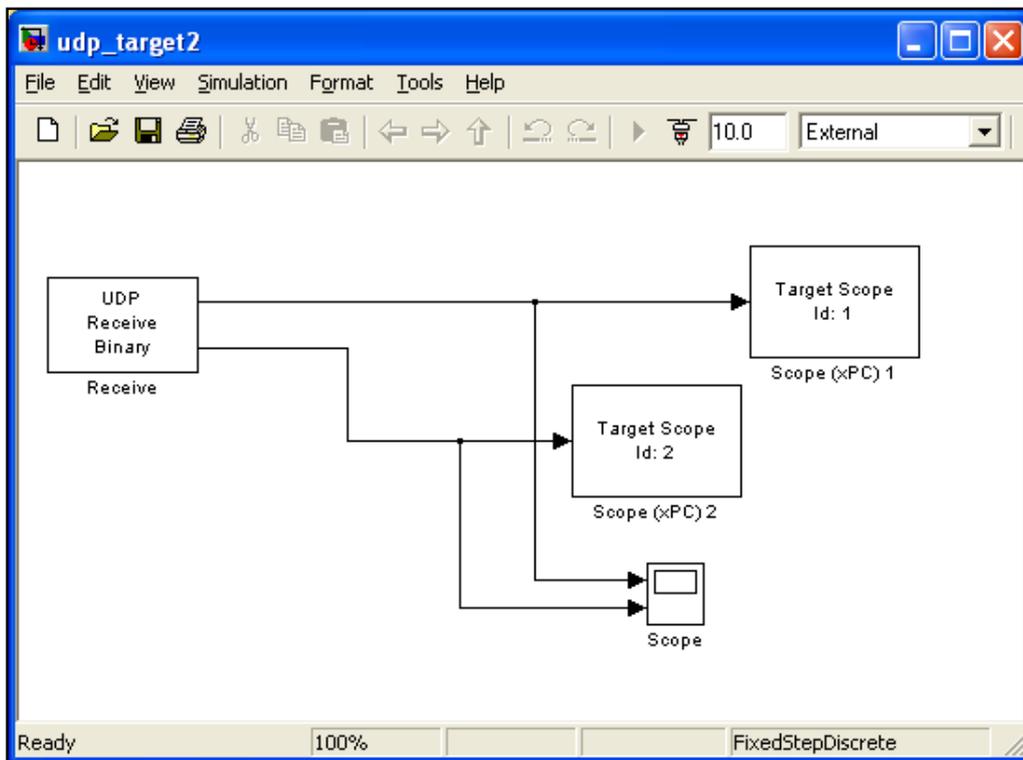


Fig. 90. PC *Target2*. Recepción de señales en binario por protocolo UDP

En el monitor del PC *Target2* se puede ver la imagen capturada que se muestra a continuación:

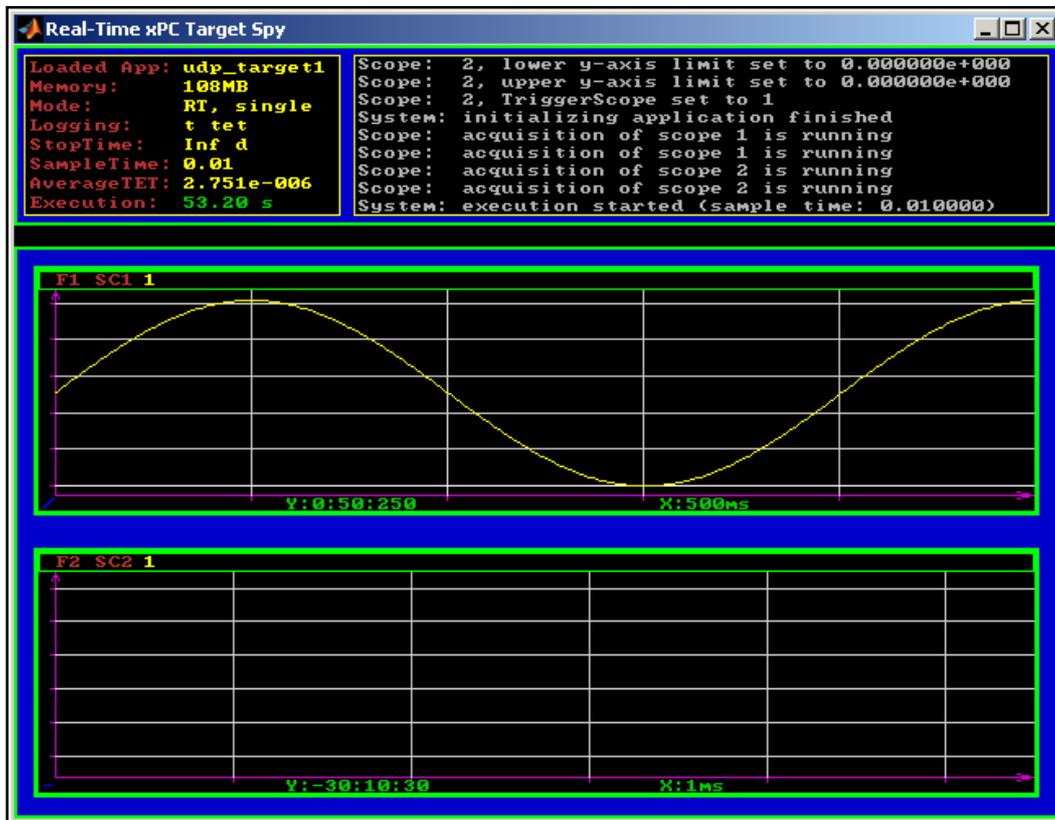


Fig. 91. PC *Target2*. Captura de imagen del monitor

En la práctica se compilaba el programa del PC *Target1*, se desconectaba la conexión con el PC *Target1* para incomunicar el envío de datos con él. Se compilaba el programa de recepción de datos por protocolo UDP al PC *Target2* y posteriormente se volvía a conectar el cable RJ45 del PC *Target1* para facilitar el envío de datos y en consecuencia el funcionamiento del sistema.

Con este protocolo, se ha comprobado que el envío de datos es muy lento, además de inestable e impreciso. El *sample time* (tiempo de muestreo) ha sido de 0.01s en el PC *Target1* y de 0.005s en el PC *Target2*.

Por esto, se llegó a la decisión de que la mejor opción para el envío de señales no era el cable serie RS-232, ni el cable de red RJ45 mediante protocolo UDP. La mejor opción era el uso de las tarjetas de adquisición de datos *National Instruments*.



8. PROGRAMACIÓN DE LA ARQUITECTURA DE CONTROL. ENVÍO DE LA PWM DE CONTROL A TRAVÉS DE LA NI

8. PROGRAMACIÓN DE LA ARQUITECTURA DE CONTROL. ENVÍO DE LA PWM DE CONTROL A TRAVÉS DE LA NI

Posteriormente, se llegaría a la aplicación con objeto de este proyecto. El uso del hardware implementado para el control del disparo del convertidor. La implementación del modelo es objetivo del proyecto [7]. Siendo imprescindible decir que el objetivo del presente proyecto era permitir el uso de dicho control.

La utilidad del presente proyecto para permitir el control, mediante los pulsos al convertidor quedó patente en los ensayos posteriores. Permitiendo controlar el giro del motor, gobernando sus parámetros de control.

La topología hardware utilizada sería la que sigue:

xPC Target1 -> SCB-100 (counter 0) -> Osciloscopio

El modelo utilizado para generar los pulsos que gobiernan al convertidor es el siguiente:

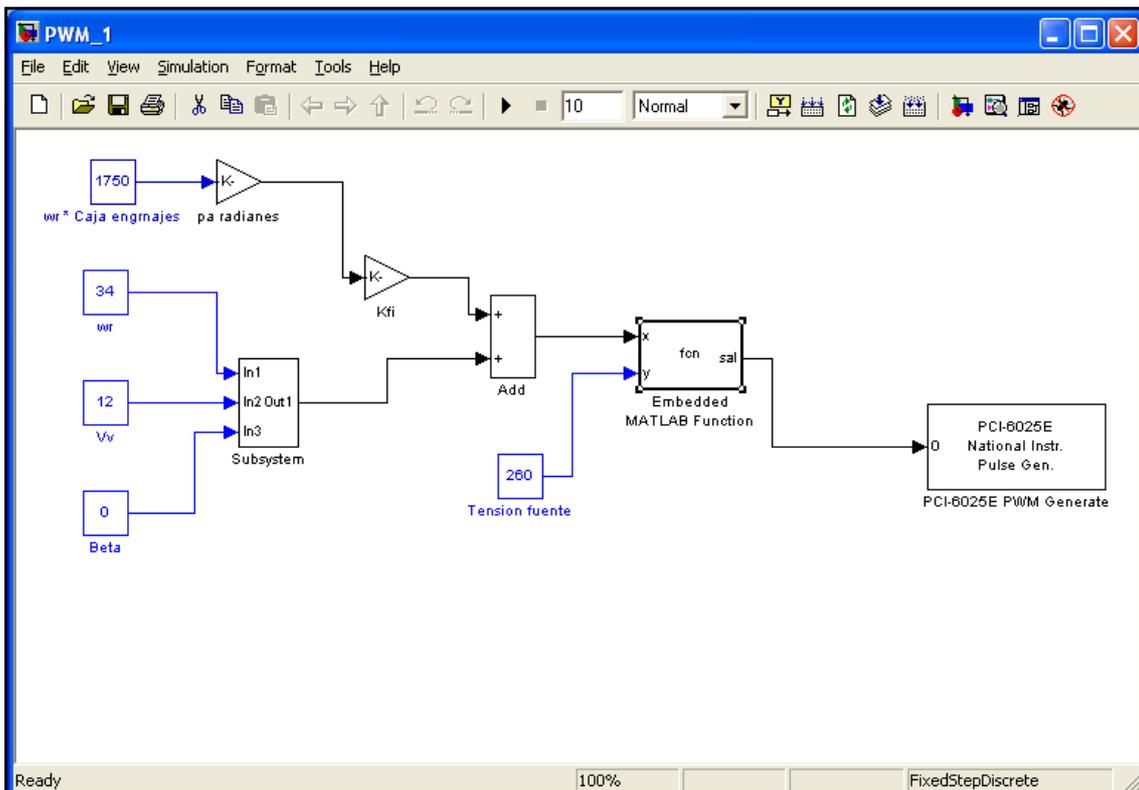


Fig. 92. Modulación PWM para el convertidor a través de la tarjeta NI SCB-100 [7]

Los pulsos se midieron en el osciloscopio. Se muestran a continuación:

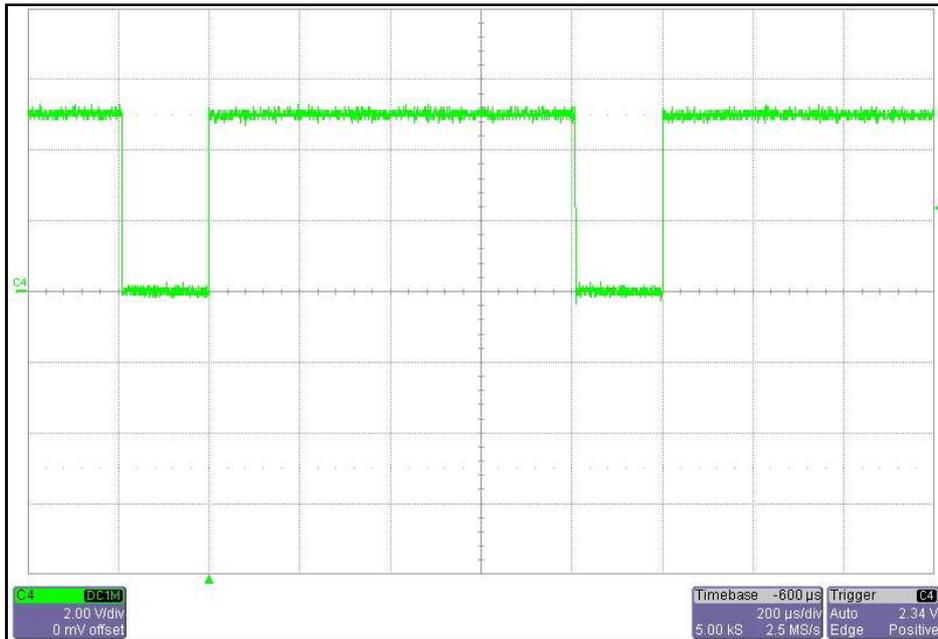


Fig. 93. Representación de la modulación PWM. Modulación de 1000µs.

El sistema permanece un 80% del tiempo activo y un 20% desactivado. El tiempo de muestreo más preciso a utilizar es $7e-5s$ (un tiempo menor, por ejemplo $5e-5s$) satura al ordenador y lo colapsa.

La tarjeta NI SCB-100 tiene un contador, que es el encargado de mostrar y calcular los $1000\mu s$ preestablecidos en nuestro programa, puesto que hemos querido que conmute a $1000\mu s$. El tiempo de muestreo del ordenador nos dará mayor precisión cuanto más pequeño sea, pues tomaremos más muestras en $1000\mu s$.



Fig. 94. Misma gráfica con una escala de tiempos menor



9. SELECCIÓN DE LA ARQUITECTURA HARDWARE DEL CONTROL DEL MOTOR DC

9. SELECCIÓN DE LA ARQUITECTURA HARDWARE DEL CONTROL DEL MOTOR DC

Mediante la compilación del modelo de control del convertidor, en el que se tienen en cuenta las condiciones de viento y posibles situaciones de operación del aerogenerador, se ha conseguido controlar el motor DC.

A continuación se muestra en la imagen el hardware definitivo conectado al prototipo del aerogenerador. Mediante el control del giro del motor DC.

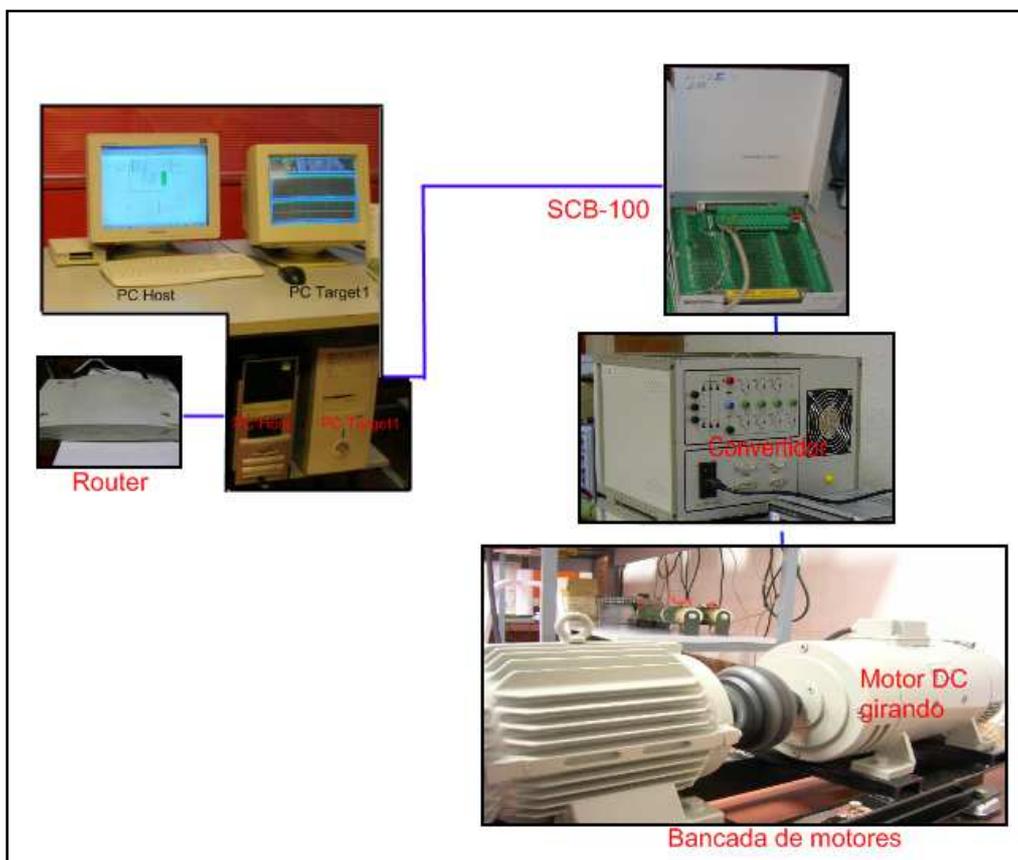


Fig. 95. Hardware definitivo

9.1. Ejemplo de conexión de la máquina DC a través de la xPC Target

Se hizo un ensayo en el laboratorio que consistía en arrancar y parar la máquina DC aplicando una tensión triangular. La señal triangular aplicada comienza en 1 en el instante inicial, llega a 99 en 100 segundos y vuelve a 1 a los 200 segundos. Los valores 1 y 99 son para evitar los extremos en el ciclo de trabajo que puedan saturar la salida. El archivo de la Fig. 96. fue desarrollado en *Simulink* para este ensayo. [7]

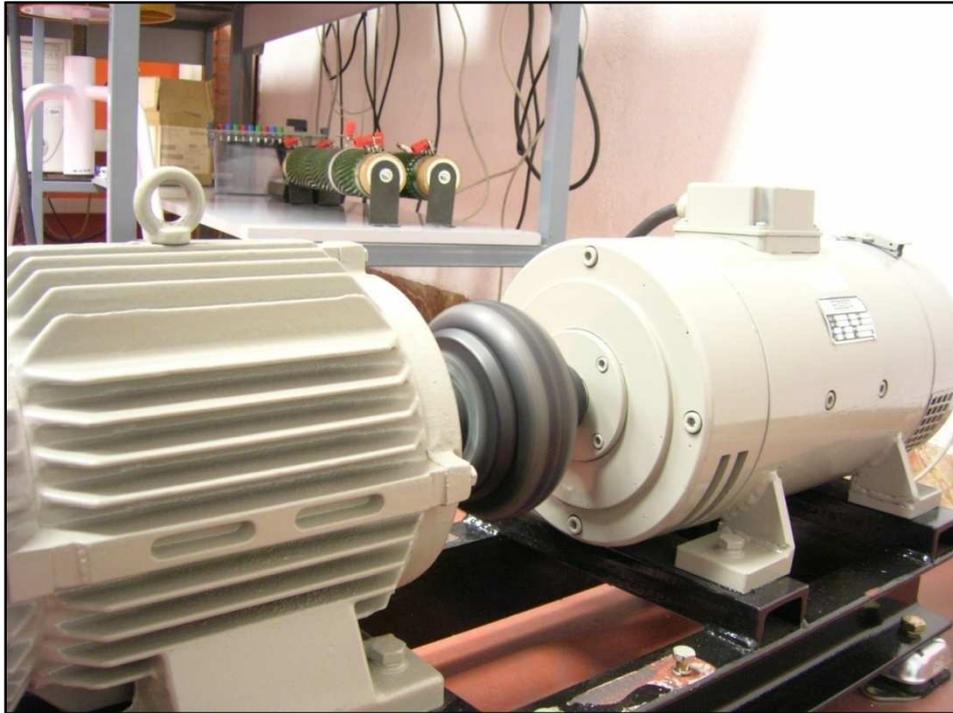


Fig. 97. Fotos del ensayo en el laboratorio

La tensión y corriente del inducido de la máquina DC fueron registradas tanto en el osciloscopio del laboratorio como en un scope del PC Target. Las gráficas obtenidas en ambos casos son presentadas a continuación:

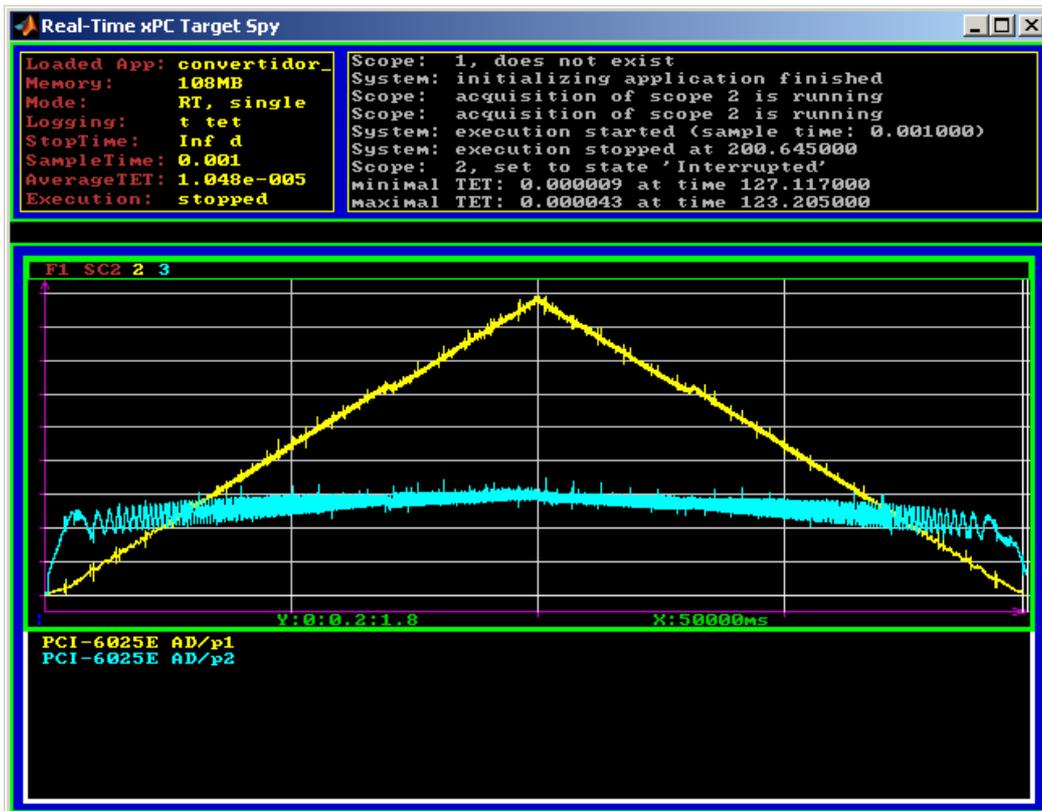


Fig. 98. Scope xPC Target.

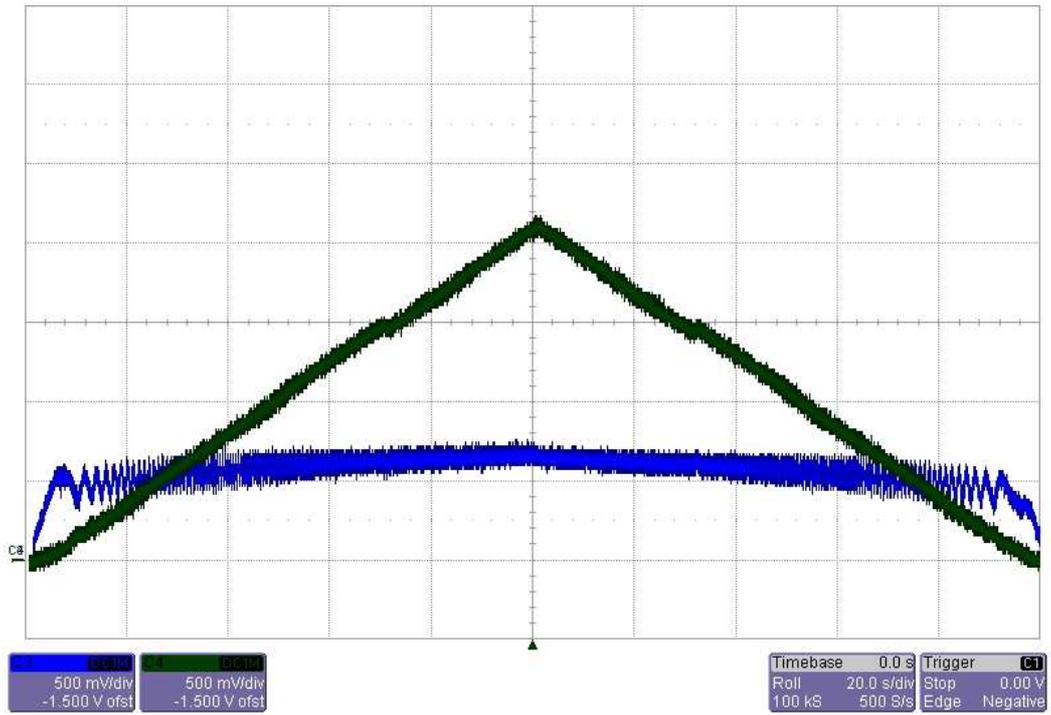


Fig. 99. Osciloscopio laboratorio.



10. CONCLUSIONES



10. CONCLUSIONES

- **PC Host, PC-104. Cable serie RS-232**

Se realizó la comunicación entre un ordenador anfitrión (PC *Host*) y un ordenador destino (PC-104) mediante un cable serie RS-232 consiguiendo:

- Que un ordenador destino (PC-104) realice los cálculos de un modelo de *Simulink* compilado de manera remota por un ordenador anfitrión (PC *Host*). De esta manera se consigue liberar de carga de cálculo a nuestro ordenador principal. Permittiéndonos controlar de manera remota y en tiempo real un modelo real (Fig. 33).
- Mediante el modelo anterior de *Simulink*, compilado en el PC-104 se ejecuta el cálculo de una señal sinusoidal cuyo cálculo se grafica en el monitor del PC-104 (Fig. 36).

- **PC Host, PC Target. Cable de red RJ45 cruzado**

Se comprobó que la comunicación entre PC's se podría realizar no sólo con un ordenador de bajas prestaciones construido para la aplicación en concreto, sino que también era posible comunicar dos ordenadores personales como los que tenemos en nuestras casas o en la Universidad.

Se permiten velocidades de computación de $1,5 \cdot 10^{-5}$ segundos, eficiente para la aplicación para la cual planteamos el objetivo. El control del disparo de convertidores (de frecuencias de conmutación de 1kHz, $1 \cdot 10^{-3}$ segundos) de un aerogenerador.



- **PC Host, PC's Target. Mediante cable de red RJ45 con un Router**

La herramienta *xPC Target* nos permite comunicar un PC *Host* con varios PC's *Target*. Permitiendo una comunicación unidireccional y por turnos entre el PC *Host* y cada PC *Target*. Siendo necesario cambiar tímidamente alguna configuración en los comandos a tal efecto para realizar una correcta comunicación con el ordenador deseado.

El PC *Host* dispone de una tarjeta de red, a la que conectar un cable de red RJ45, así como cada uno de los PC *Target* [12].

Para poder realizar una conexión entre un PC *Host* y varios PC's *Targets* se llegó a la solución de conectarlos a través de un Router con varias entradas para cables RJ45. Por lo que se solucionó la disyuntiva de cómo conexionar un PC *Host* con varios PC *Target*.

- **Conexión mediante Protocolo UDP**

Los ensayos mostraron que la respuesta del hardware no respondía a lo esperado. Por lo que se desechó. Siendo la elección, por velocidad, respuesta, y ajuste la conexión mediante protocolo TCP/IP.

- **Tarjetas de adquisición de datos National Instruments**

Mediante las tarjetas de adquisición de datos de *National Instruments* (SCB-68 y SCB-100) [15] se pudo conexionar un modelo cargado en el PC *Target* con la realidad. Se creó un modelo, como el que se muestra en la (Fig. 60), el cual genera una señal que es convertida a dígitos binarios, es decir, se digitaliza. Y se envía al exterior a través de los pines digitales que dispone la tarjeta de adquisición de datos SCB-68 (Fig. 75 y 76).

Dichas salidas digitales se envían a las entradas digitales de la tarjeta SCB-100. (Fig. 77 y 78). Estos datos los ejecuta de nuevo el ordenador, recomponiéndolos mediante una función que convierte los datos digitales a analógicos, de manera que se reconstruya la señal sinusoidal analógica inicial.

Se obtuvo dicha señal por el osciloscopio, a partir de los pines de la salida analógica de la tarjeta SCB-100 (Fig. 79 y 80).

Al reconstruir la señal se midió un retardo de 100 μ s. Fijado en *Matlab*, debido a que si se reducía este tiempo, la aplicación saturaba al PC *Target* y colapsando al sistema de control. Este era un punto crítico, debido a que si hubiese sido un retardo mayor, es decir, la señal de salida tardara más de 1kHz (1ms) en ser reconstruida, sería imposible conmutar los convertidores con la velocidad suficiente. Y la comunicación a través de la aplicación *xPC Target* no sería útil.



La señal que se obtuvo se midió con el osciloscopio. Se muestra en la Fig. 67. Y el retardo en la Fig. 74.

Observando los resultados se concluyó que se consiguió tener un hardware de comunicación lo suficientemente rápido como para realizar cálculos y dar respuestas en tiempo real, dentro de los márgenes necesarios.

A partir de este trabajo, se podrá controlar cualquier sistema físico (mediante señales de corriente y tensión) siguiendo las pautas aquí descritas y amoldándolas a la problemática en cuestión.

Las conclusiones obtenidas han sido las siguientes:

- El mayor desfase del sistema fue de $100\mu\text{s}$, por lo que el hardware implementado sería capaz de actuar para tiempos menores que este retardo.
- En el caso de comunicación con un PC *Host* conectado con cable RS-232 con dos PC *Targets* es necesario tener dos puertos COM disponibles en el PC *Host*. Estableciendo en *Host* port el COM1 para el primer disquete que se cree para el primer PC *Target* y COM2 para el segundo disquete que se creará para el segundo PC *Target*.
- Al tener en el PC *Target*, como sistema operativo la aplicación *xPC Target*, no permite una conexión inalámbrica, ya sea wifi, en el caso de Internet. (No siendo posible otro tipo de comunicación: infrarrojos, Bluetooth, etc). Siendo el único modo de comunicación entre PC *Host* y PC *Target*, mediante un cable serie RS-232 o mediante cable de red RJ45.
- Se ha conseguido crear un hardware de comunicación que puede ser aplicado a infinidad de aplicaciones reales. Cumpliendo las especificaciones para el que fue creado. Para la comunicación de un aerogenerador de velocidad variable. Por lo que, además, se desecha la posibilidad abierta de utilizar una *DSP*.
- El coste total de nuestro hardware es de 950€ frente a los 4300€ del sistema hardware implementado en [21].



11. BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN



11. BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN

- [1] R. Ovando, J. Aguayo, M. Cotorogea, “Emulation of a Low Power Wind Turbine with a DC motor in *Matlab/Simulink*”. Power Electronics Specialists Conference, 2007. PESC 2007. IEEE. Volume, Issue, Page(s): 859 – 864.
- [2] Plan de Energías Renovables en España, 2005-2010. Ministerio de Industria, Turismo y Comercio. 21 de Julio de 2005. Resumen online disponible en: <http://www.mityc.es/NR/rdonlyres/C1594B7B-DED3-4105-96BC-9704420F5E9F/0/ResumenPlanEnergiasRenov.pdf>
- [3] Weiwei Li; Dianguo Xu; Wei Zhang; Hongfei Ma, “Research on Wind Turbine Emulation based on DC motor”. Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on. Volume, Issue, 23-25 May 2007 Page(s): 2589 - 2593.
- [4] Asociación Empresaria Eólica (AEE). Nota de prensa 17 de Enero de 2008. “La potencia eólica instalada en España alcanza los 15.145 MW con 3.515 MW nuevos en 2007”
- [5] Asociación Empresaria Eólica (AEE). Nota de prensa 16 de Enero de 2008. “Nuevo récord de generación eólica con 9.550MW”.
- [6] www.laprensagrafica.com. Dominical.
- [7] Luis Cabañas Gutiérrez. “Emulador de un aerogenerador de velocidad variable”. Proyecto fin de carrera de la Universidad Carlos III de Madrid. Ingeniería Industrial Electricidad. Junio de 2008.
- [8] <http://www.invap.net/indus/eolica/contemporaneas.html>. Sección Industrial, Energía eólicas, Turbinas eólicas.
- [9] www.repower.com.au. Fabricante de turbinas eólicas.
- [10] www.vestas.es. Fabricante de turbinas eólicas marítimas.
- [11] *Mathworks, Inc. xPC Target User’s Guide*. Año 2003.
- [12] *Mathworks, Inc. xPC Target For Use with Real-Time Workshop®*. Getting Started with *xPC Target*. Año 2006.
- [13] www.mathworks.com (Página oficial de *Matlab*).



- [14] Ricardo Albarracín Sánchez. “Manual de la arquitectura de comunicación Host-Target mediante Simulink de un aerogenerador DFIG”. Manual departamento de Ingeniería Eléctrica de la Universidad Carlos III de Madrid. Diciembre de 2007.
- [15] digital.ni.com (Manuales de las tarjetas *National Instruments* SCB-68, SCB-100).
- [16] *Spectrum Digital DSP 2812*. eZDSP™ F2812. Technical Reference Manual. September 2003.
- [17] Javier Mazón. “Prototipaje rápido de redes neuronales mediante *xPC Target* de *Matlab*”. Artículo para el Congreso Internacional de Ingeniería de Proyectos. AEIPRO. ETSI de Bilbao. Octubre de 2004.
- [18] Modesti M. R.; Vaschetti J. C.; Jazni J. E.; Cova W.; Naguil J. L. Pedroni J. P. “Plataforma de simulación en tiempo real del control del vector de empuje de un vehículo lanzador”. Artículo para el IV Congreso Argentino de Tecnología Espacial. Universidad Tecnológica Nacional, Facultad Regional de Córdoba, Argentina. Mayo de 2007.
- [19] Armando Fernández Pellejero. “Estudio de una aplicación orientada al control y simulación de procesos en tiempo real”. Proyecto fin de carrera de la Universidad de la Rioja. Ingeniería Técnica Industrial, Electrónica. Año 2007.
- [20] Kevin Lynch; Benjamin Stephens. “Getting started with *xPC Target* with your *PC/104* stack”. Practical Manual of the Engineering University of Chicago. January 2006.
- [21] Miguel Álvar Mayor. “Contribuciones al sistema electrónico de control de un convertidor NPC de tres niveles de 250kVA usado en turbinas eólicas”. Proyecto fin de carrera de la Universidad Politécnica de Alcalá de Henares. Ingeniería Técnica de Telecomunicación (Especialidad Sistemas electrónicos). Junio de 2006.
- [22] www.realtek.com (Página de fabricante de tarjetas de red Ethernet).

