



UNIVERSIDAD POLITÉCNICA DE VALENCIA  
MASTER EN AUTOMÁTICA E INFORMÁTICA INDUSTRIAL  
TESINA DE MÁSTER

# Diseño, Identificación y Control de un Prototipo de Horno Continuo

**Alumno** : Alfonso Escrivá Calafat

**Tutor** : Antonio Sala Piqueras

**Fecha** : 30/09/2010



# Agradecimientos

Esta tesina no es fruto de un simple trabajo individual, sino que es el resultado de una suma de apoyos y esfuerzos a lo largo de muchos meses. A todas las personas que, aunque no sean conscientes de ello, han hecho posible finalizar esta tesina, mil gracias de corazón:

A Bea, por ser como es, por quererme como soy, por aguantarme tantos ratos en este último año en los que tenía mi cabeza nublada, por tu amor, . . . por tantas cosas, te quiero.

A mis padres, porque a ellos les debo todo lo que soy y siempre serán mi referencia. A Germán, por que es mi otro yo, nunca dejes de creer en tí. A mis abuelos, y a Matilde, que se sentiría orgullosa de su nieto.

A mis amigos, porque amigos de verdad se encuentran pocos y yo sé bien donde están. Jonathan, por ser un tío genial, sabes que siempre seremos tu familia. Como no a Fran, Ricardo y Noel, amigos de esos que valen la pena y que “han sufrido” esta tesina. Como también han hecho Bonet y Jordi, soportes en la vida diaria. A Ana y a Nuria, también sois mi familia.

A mi tutor, Antonio Sala, por guiarme a lo largo de estos meses de duro trabajo y cederme parte de ese tiempo que no le sobra. También a los técnicos por sus buenos consejos. Sin ellos esta tesina no hubiera sido posible. A mis compañeros de laboratorio, Román y Andreu. Y por último al Programa Prometeo financiado por la Conselleria d’Educació de la Comunitat Valenciana, y dirigido por Pedro Albertos Pérez.



# Índice general

<b>Introducción</b>	<b>III</b>
<b>Objetivos y Motivaciones</b>	<b>III</b>
<b>I Diseño del sistema</b>	<b>5</b>
<b>1. Estado del Arte</b>	<b>7</b>
1.1. Horno continuo	7
1.1.1. Características básicas	8
1.2. Control de Hornos Continuos	9
1.2.1. Otros efectos y característica a tener en cuenta	12
<b>2. Presentación del Proceso</b>	<b>13</b>
2.1. Diseño general del horno con la cinta transportadora	13
2.2. Descripción del Hardware	14
2.2.1. Actuadores	14
Calentamiento. Radiadores	14
Características de Funcionamiento	15
Modelo del Actuador	16
Enfriamiento. Control de ventiladores (motores DC)	17
Funcionamiento Ventiladores	19
2.2.2. Transductores	19
Rangos de Temperatura	20
Escalado y Centrado: Ajuste de Temperatura Máxima y Mínima	21
Calibración. Relacionar Voltios con Grados.	22
<b>3. Interface con el computador</b>	<b>25</b>
3.1. Tarjetas de adquisición de datos AD y DA	25
3.2. Tarjetas	27
3.2.1. Driver DAQ-MTLB	28

3.3. DAQ-LVIEW PnP . . . . .	31
<b>II Identificación y mejoras</b>	<b>35</b>
<b>4. Identificación</b>	<b>37</b>
4.1. Introducción . . . . .	37
4.2. Procedimiento . . . . .	37
4.3. Definir el Comportamiento del Sistema . . . . .	38
4.3.1. Elección de la Estructura . . . . .	39
Representación del Sistema . . . . .	40
4.4. Modelo Horno . . . . .	41
4.5. Diseño del experimento . . . . .	43
4.5.1. Elección del tipo de entrada . . . . .	43
4.5.2. Elección del periodo de muestreo . . . . .	44
4.5.3. Recogida de datos . . . . .	44
4.5.4. Tratamiento de datos . . . . .	45
4.6. Cálculo del Modelo . . . . .	46
4.7. Validación del Modelo. . . . .	49
4.8. Conclusiones iniciales . . . . .	49
<b>5. Rediseño del Horno</b>	<b>51</b>
5.1. Problemas que presenta el montaje . . . . .	51
5.2. Ruido en los termopares . . . . .	52
5.3. Atenuación del Fenómeno de Aliasing . . . . .	53
5.3.1. Detección de Alias . . . . .	53
5.3.2. Eliminación de Alias de Alta frecuencia . . . . .	53
Implementación . . . . .	55
5.3.3. Diseño del Filtro Antialiasing . . . . .	57
Selección de valores del filtro e instalación en la DAQ PCI9111 . . . . .	57
Simulación e Implementación . . . . .	58
5.4. Mejora de la adquisición . . . . .	60
5.4.1. Mejora de precisión en la unión fría. . . . .	60
5.4.2. Mejora de la alimentación de los transmisores. . . . .	61
Simulación e Implementación . . . . .	61
5.5. Mejora de la señal de los actuadores . . . . .	63
Simulación e Implementación . . . . .	64
5.6. Apantallamiento . . . . .	66
5.7. Modificación Estructural del Prototipo . . . . .	66

5.8. Medición en calentamiento . . . . .	67
5.9. Reparación del sistema de ventilación . . . . .	67
<b>III Identificación, Control y Resultados experimentales</b>	<b>69</b>
<b>6. Identificación después del Rediseño</b>	<b>71</b>
6.1. Introducción . . . . .	71
6.2. Modelo de Puesta en Marcha $G1(s)$ . . . . .	71
6.2.1. Condiciones del modelo . . . . .	71
6.2.2. Obtención del modelo . . . . .	72
6.3. Modelo Nominal $G2(s)$ . . . . .	74
<b>7. Control</b>	<b>77</b>
7.1. Introducción . . . . .	77
7.2. Arquitectura de Control . . . . .	77
7.3. Control por Realimentación del Estado . . . . .	79
Reducción del número de estados . . . . .	79
Reducción utilizando Matlab . . . . .	80
Modelo discreto en espacio de estados reducido . . . . .	81
7.4. Observador del estado . . . . .	82
7.4.1. Observabilidad del sistema . . . . .	82
7.4.2. Diseño de un observador óptimo (Kalman Filter) . . . . .	82
Sintonización de parámetros QN ( $q_{obs}$ ) y RN ( $r_{obs}$ ) . . . . .	85
7.5. Diseño del Controlador . . . . .	85
7.5.1. Controlabilidad del sistema . . . . .	85
7.5.2. Controlador LQR . . . . .	86
7.5.3. Diseño LQR . . . . .	87
Sintonización de parámetros. Modelo 3x3, $G1(s)$ . . . . .	87
7.6. Control en Bucle Abierto. Modelo 3x3, $G1(s)$ . . . . .	88
7.7. Control 2-DoF. Modelo 3x3, $G1(s)$ . . . . .	90
7.7.1. Simulación . . . . .	93
7.7.2. Resultados Reales . . . . .	93
7.8. Control 2-DoF. Modelo 8x6, $G2(s)$ . . . . .	96
7.8.1. Simulaciones . . . . .	97
7.8.2. Resultados Reales . . . . .	98
7.8.3. Posibles soluciones a realizar . . . . .	101
<b>8. HMI</b>	<b>103</b>

8.1. Introducción	103
8.2. HMI de Identificación	104
Labview Control Panel	104
Labview Block Diagram	105
8.3. HMI de Control Horno	111
Labview Panel Control	111
Labview Block Diagram	112
<b>9. Conclusiones</b>	<b>119</b>
<b>IV Apéndices</b>	<b>121</b>
<b>A. Conceptos útiles</b>	<b>123</b>
A.1. Definiciones	123
A.2. Acrónimos	125
A.3. Tipos de ruido que podemos encontrar	125
<b>B. Instrucciones DAQ-MTLB</b>	<b>127</b>
<b>C. Programas</b>	<b>129</b>
C.1. Experimentos de adquisición de datos	129
C.1.1. Bucle de Adquisición	129
C.1.2. Función para inicializar PCI9111	130
C.1.3. Función para inicializar PCI6208(6216)	132
C.2. Código de Identificación antes del rediseño	132
C.2.1. Función para identificar escalón positivo y tao ideal	133
C.2.2. Función cálculo del retardo y de tao real	135
C.2.3. Función para pasar de Voltios a Grados	136
C.2.4. Función para pasar de Voltios a Watos	136
C.3. Código de Identificación despues del rediseño	136
C.4. Diseño de Realimentación	139
C.4.1. Ejemplo para proceso 3x3	139
C.4.2. Ejemplo para proceso 6x8	140
C.5. Implementación de Bucle de Control	142
C.6. HMI Control Completo	144
C.6.1. Cargar todos los datos	144
C.6.2. Genero estructura de datos para fase de puesta a punto	145
C.6.3. Implemtación del Bucle de Puesta a Punto	146
C.6.4. Genero estructura de datos para Fase de Control	146

C.6.5. Código Control Modelo Completo, $G^2(s)$ . . . . .	147
C.6.6. Guardar datos . . . . .	149



# Resumen

El presente trabajo describe la implementación de una plataforma multivariable que intenta representar las características de un horno industrial. El diseño general de la plataforma fue realizado por el técnico de laboratorio Manuel Yago.

También se describe la metodología de identificación llevada a cabo y se presentan los resultados de los experimentos preliminares realizados, analizando de forma concurrente todos los problemas de diseño que presenta la plataforma. Además se ha incluido un ejemplo de implementación de control mediante una arquitectura basada en 2 grados de libertad (2-DoF), y se ha descrito las limitaciones de control que presenta el sistema.

El documento se ha estructurado del siguiente modo:

- En primer lugar, en el capítulo 2, veremos la construcción del horno, tanto su estructura como implementación.
- A continuación, se expone en el capítulo 3 los Sistema de adquisición de datos utilizados (PCI-9111 y PCI-6018 de ADLINK) y los drivers utilizados.
- En capítulo 4 se recoge el procedimiento de identificación de la planta.
- En el capítulo 5 se describen los problemas de instrumentación y construcción que tiene el sistema, y las actuaciones de rediseño realizadas.
- Después de estos cambios se ha vuelto a realizar la identificación y control del proceso, optimizando el procedimiento (capítulo 6).
- En el capítulo 7 se describe los bucles de control implementado, y se presenta los resultados los resultados de control obtenidos después del rediseño.
- Seguidamente se resumen y muestran los Scada generados para la realización de los experimentos (capítulo 8)

- Por último, en el capítulo 9, se exponen las conclusiones del trabajo realizado.

En el documento también se han añadido los anexos necesarios para la comprensión completa del trabajo, en éstos se describe:

- Los principales conceptos utilizados, anexo [A](#).
- Descripción de los drivers utilizado para la comunicación con Matlab (Daq-MTLB), anexo [B](#).
- Finalmete se expone los m-files generados para las fases de identificación y control, anexo [C](#).

# Motivación y Objetivos

El siguiente documento se enmarca dentro del Programa Prometeo 2009-0268 de la Generalitat Valenciana, orientado en este caso al Estudio e Implementación de Técnicas de Control Multivariable en procesos con dinámica lenta, concretamente al prototipo de un Horno Industrial. La elección del Horno Industrial como proceso viene motivada por su naturaleza multivariable y por su extensión en la industria de la zona (Cerámica, Alimentación, . . .).

La elevada multiplicidad de actuadores y sensores de este tipo de sistemas hace que su control presente un alto carácter multivariable, donde cada sensor está afectado por varios actuadores, y viceversa. Por este motivo, al intentar gobernar automáticamente la temperatura interior del horno surgen importantes problemas de acoplamiento [3], [6], [13]. El control de temperatura en Hornos



Figura 1: Imagen de un horno industrial

Industriales es un requisito indispensable para obtener un producto final de buena calidad y para optimizar el tiempo de permanencia del producto en su interior, así pues una mejora en los sistemas de control nos acerca al objetivo final de toda industria, mejorar la calidad y la productividad, [6].

Así pues el objetivo final que se persigue en este trabajo es disponer de un sistema capaz de reproducir las características de funcionamiento de un horno industrial, para implementar técnicas de control multivariable que tengan en cuenta los acoplamientos. En la actualidad el control de los hornos se suele realizar mediante el uso de controladores más sencillos, tantos como secciones tenga el horno, considerando cada una de las secciones como un proceso monovariable sin tener en cuenta los acoplamientos entre zonas, [6]. En hornos de dimensiones reducidas los problemas

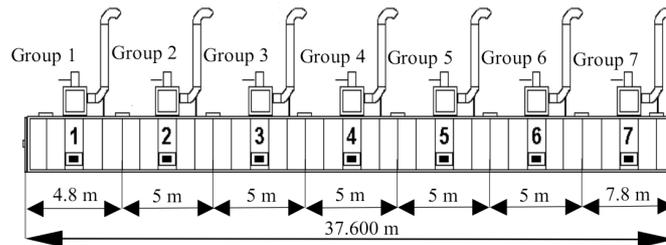


Figura 2: Esquema simplificado de un horno de 7 secciones, [6]

de acoplamiento serán aun más mayores, ya que la diferenciación de fases o compartimentos es más difícil, y por tanto la interacción entre los elementos es mayor, dificultando el diseño de controladores multivariables.

Con estos antecedentes, el diseño de controladores capaces de conseguir que la temperatura interior del horno siga, con el menor error posible, una curva de temperatura de referencia preestablecida se plantea como un problema complejo pero de gran interés.

## **Parte I**

# **Diseño del sistema**



# Capítulo 1

## Estado del Arte

### 1.1. Horno continuo

El horno continuo u horno tipo túnel, es aquel en que el producto a cocer entra por uno de sus extremos y sale por el otro, pudiéndose utilizar como combustible gas, gas-oil o si se trabaja a temperaturas más bajas radiadores eléctricos. También cabe la posibilidad de encontrar ambas fuentes de calor combinadas [35], [14], [5], [7], [29].

La forma de cocción puede ser de dos tipos:

- Radiante (o directo) donde el material está en contacto con la fuente de calor, siendo la transmisión de calor por radiación alrededor del 80 % y, el otro 20 % sería por convección. Esta forma de trabajar se suele utilizar más para hornos de gas, donde la llama estará en contacto directo con las piezas, [35].
- De convección (o indirecto) donde el material está fuera del alcance de la fuente de calor [14], [5], [7], [29]. Este tipo de hornos puede ser de cualquiera mencionados anteriormente, aunque por gas o gas-oil la convección se puede hacer mediante la llama a distancia o por calentamiento de tubos o planchas, [29].

Este tipo de horno tiene una gran cantidad de aplicaciones, tales como temple, recocido, revenido, cementación, precalentamiento, endurecimiento, envejecimiento artificial, recocido blando o revenido, secado, . . . [15], [29]. Pero en cada uno de los ejemplos mencionados existe un objetivo común, adecuar el perfil de la curva de temperatura para que la aplicación minimice el efecto de la mala distribución de temperaturas, [35].

### 1.1.1. Características básicas

En los hornos continuos se establece una cadena transportadora en el interior que, debidamente motorizada, permite un avance continuo de cada unidad de producto a lo largo del horno, de manera que el tiempo de permanencia en el mismo será el adecuado para que el producto alcance la boca de salida del horno perfectamente acabado, [35].

Los tipos de cintas más utilizados son de banda de acero lisa o malla articulada, [7], y de rodillos metálicos, [14]. En la actualidad se pueden cocer ladrillos en tiempos de 2 horas en hornos de rodillos ya que en ellos la calidad de la transferencia de calor es muy buena. Parece ser que ésta es la tecnología que se impondrá ya que permite ahorros energéticos importantes, facilidad de operación y mecanización y, sobretodo, calidad, [14].

El ajuste del tiempo de cocción es esencial para producir resultados consistentes, por tanto la velocidad de la cinta determinará el tiempo preciso que el producto está expuesto al calor, [1]. El avance del producto debe estar combinado con la temperatura del horno, ya sea utilizando resistencias eléctricas o quemadores de gas o gas-oil, [35]. Tradicionalmente se utiliza un taquímetro para medir la velocidad y luego mediante una tabla, se determina la velocidad necesaria para determinado tiempo de cocción. Para implementar una solución de este tipo, se podría acoplar un encóder a uno de los ejes de la cinta transportadora, [1].

Es común dividir el horno en varias zonas de temperatura, [5], [15], [14] donde el uso de circulación de aire forzado para evitar los cambios bruscos es fundamental. El sistema de calor y flujo de aire dentro del túnel del horno genera un efecto de convección logrando que la cocción de las masas de producto sea uniforme, [35]. Se pueden encontrar hornos con circulación de aire horizontal o vertical para adaptación a la carga, [29].

El número y las dimensiones de los quemadores será variable en función de la longitud, anchura y altura del túnel del horno y también de las masas del producto a cocer. Es conveniente una altura considerable y distribuir adecuadamente los calentadores, [35].

La disposición de éstos puede variar en función del tipo de quemadores y de la zona en que se encuentren, por ejemplo [14] indica que en la zona de precalentamiento, o de baja temperatura es preciso mejorar la tasa de transferencia de calor. Para ello es conveniente inyectar, por la parte superior del horno, aire a gran velocidad y colocar fuentes de calor de gran potencia en la parte inferior del horno de esta forma el chorro de aire rompe esta distribución. Mientras que en la zona de alta temperatura, o simplemente, zona de cocción, deben usarse otros tipos de calentadores que potencien la transferencia de calor por radiación. Otro ejemplo se puede encontrar en [35]

donde se indica que los calentadores deben estar dispuestos transversalmente a nivel superior e inferior proyectando el calor hacia las caras superior e inferior de la cadena o cinta transportadora.

## 1.2. Control de Hornos Continuos

Como se ha descrito las piezas son cargadas sobre la cinta y recorren las diferentes zonas del horno, adquiriendo en cada una la temperatura adecuada, [15]. Para medir dichas temperaturas la gran mayoría de los hornos disponen de termopares instalados en las paredes o en la bóveda. Estos indicarán la temperatura correspondiente en los puntos donde están ubicados, formando una curva de temperatura que se conoce como Curva Estática, ver figura 1.1, [14].



Figura 1.1: Curva Estática de Temperaturas

En comparación con otras tecnologías los termopares son los sensores que tienen una respuesta más rápida frente a cambios de temperatura y son los que mayor rango de temperatura pueden medir, ver descripción anexo A.

Sin embargo la temperatura real a la que está sometido el material no es la que indica el termopar de la bóveda sino que corresponde a un valor sensiblemente inferior. Para determinar esta temperatura se debe introducir uno, o más termopares en la propia vagoneta (entre el propio material a cocer) e ir registrando las temperaturas. Esta gráfica se la denomina Curva Dinámica de temperatura [14], ver figura 1.2. Así pues será necesario un sistema de control que trate de seguir, con el menor error posible, una curva de temperatura de referencia. Dicho sistema de control se encar-

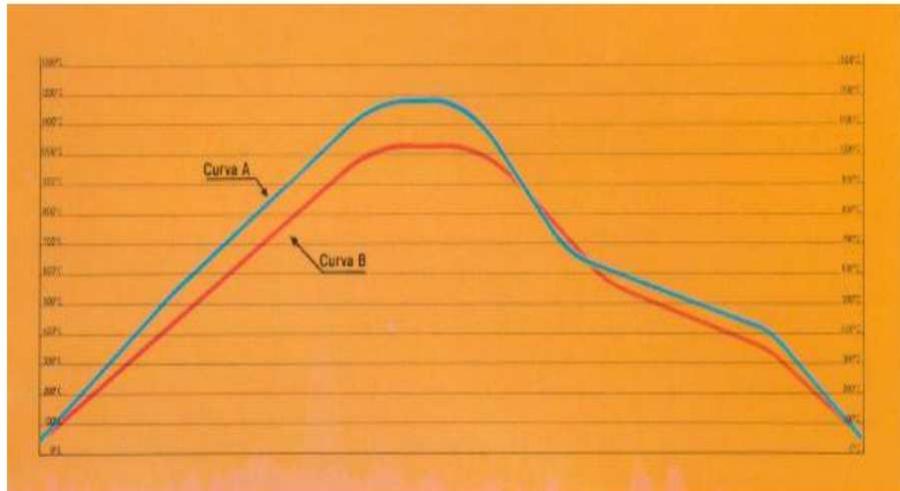


Figura 1.2: Curva Dinámica de Temperaturas

gará de regular la potencia aportada a los calentadores y al sistema de ventilación para intentar conseguir que la temperatura medida se mantenga lo más cercana posible a la deseada, [35].

En la actualidad hay algunos sistemas de control que basándose en el modelo específico de horno, regulan eficazmente la temperatura del horno. Ello conlleva que los aparatos están diseñados a medida del correspondiente modelo de horno así como a la aplicación a la que se destina. Desde controladores sencillos con una temperatura regulable hasta unidades de mando con parámetros de regulación libremente ajustables, programas almacenados e interfaz para conexión a ordenador, [29]. De esta forma se consigue un horno que estará totalmente automatizado, y que bastará con introducir los parámetros correspondientes a las características bajo las que se desee efectuar la cocción para que el sistema de control automáticamente consiga que se alcancen las condiciones óptimas, [35].

En la figura 1.3, se presenta una simulación de control de horno continuo, donde la gráfica roja representa la curva de temperatura deseada y la azul la temperatura real. Las flechas indican la dirección de la circulación de aire.

La implementación de control más extendida en este tipo de procesos esta basada en el uso de PID's, tantos como secciones tenga el horno, considerando cada una de las secciones como un proceso monovariante sin tener en cuenta los acoplamiento entre zonas, [6].

En la figura 1.4 se muestra un esquema de control monovariante con PID utilizado en un horno eléctrico, [6].

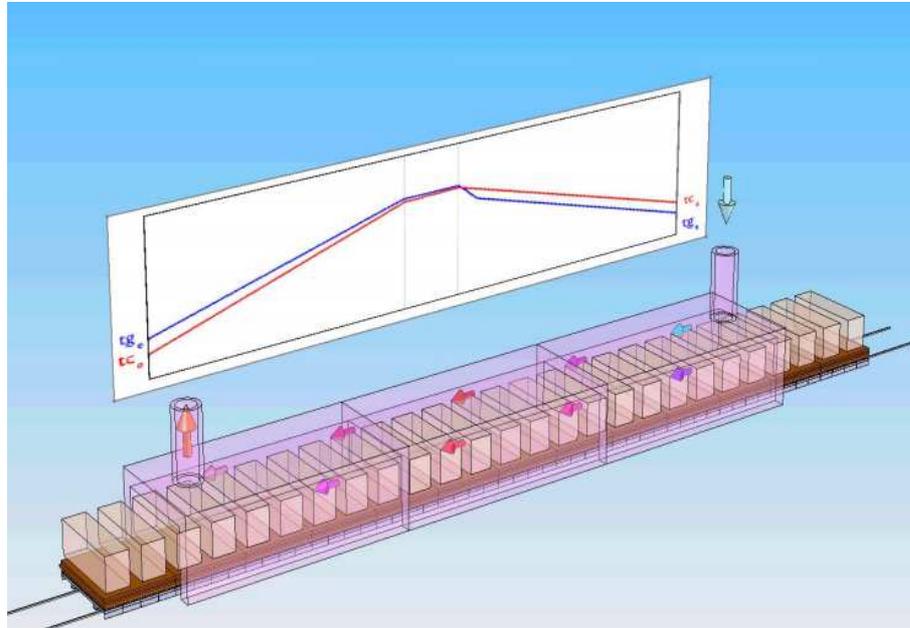


Figura 1.3: Ejemplo de Control de un Horno

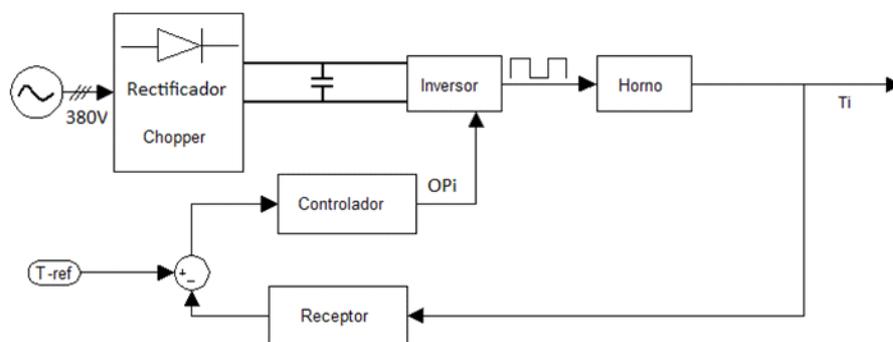


Figura 1.4: Esquema del sistema de control real

### 1.2.1. Otros efectos y característica a tener en cuenta

La clase o calidad del revestimiento, en particular la cara caliente del horno, es determinante para el consumo de los procesos intermitentes. La figura 1.5 representa una composición de una pared de un horno convencional en comparación con otra realizada con fibras, [14].

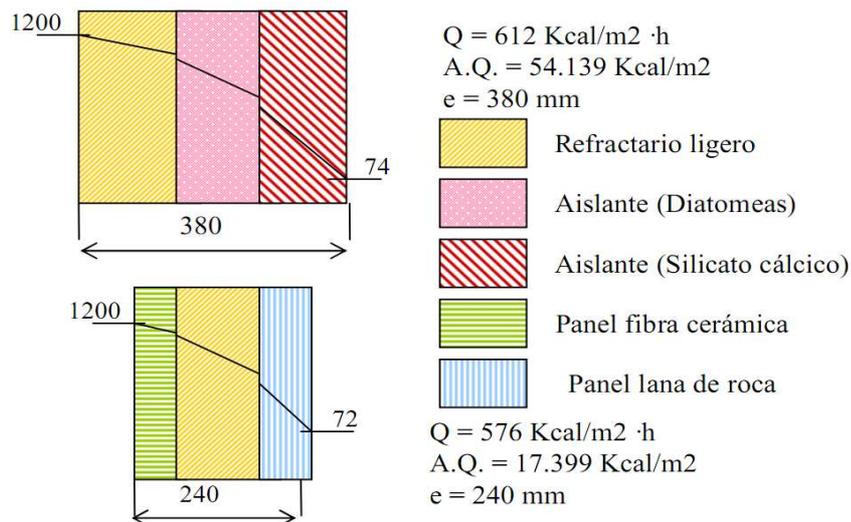


Figura 1.5: Comparación de pérdidas de Calor entre dos revestimientos de composición de materiales diferentes

Otro ejemplos para conseguir temperaturas exteriores bajas, son los siguientes:

- Aislamiento con lana mineral de alta calidad, [29].
- Para hornos pequeños con temperaturas relativamente bajas se pueden instalar patas que eleven el horno, permitiendo que todas las paredes exteriores estén ventiladas y tengan temperaturas bajas, [7].
- Durante el paso por el horno, las piezas conservan sus distancias relativas sin riesgo de golpes y con una duración de la exposición a temperatura rigurosamente igual para todas ellas, con la consiguiente uniformidad en el tratamiento, [7].
- El control de temperatura mejora notable en atmósferas controladas, [5].

## Capítulo 2

# Presentación del Proceso

### 2.1. Diseño general del horno con la cinta transportadora

La figura 2.1 es un esquema básico de la planta piloto, formada por el horno y la cinta transportadora. En el horno disponemos de 3 calentadores cerámicos en la parte superior, y de 6 ventiladores, así como de 2 “sopladores” en los laterales sobre las aberturas de entrada y de salida. Mediante los calentadores y ventiladores podemos controlar el calor que se suministra a los objetos que pasan por la cinta transportadora, y con los sopladores se controla el flujo de aire entre la entrada y la salida, y que incide directamente sobre los calentadores.

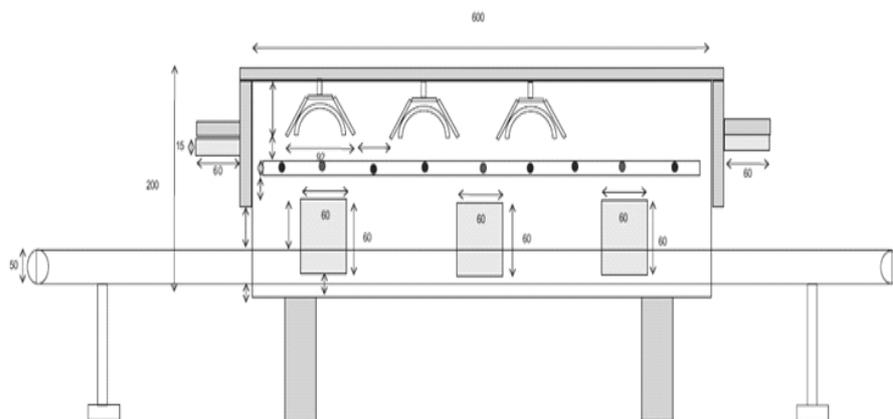


Figura 2.1: Esquema horno con la cinta transportadora

A lo largo del recorrido disponemos de 6 sensores de temperatura (aparte de los 3 que vienen incorporados en los calentadores). La cinta cuenta con un control de velocidad manual para controlar el tiempo que los objetos pasan en el interior del horno y la velocidad a la que lo recorren. El diseño de la cinta se ha realizado de modo que los objetos tardan aproximadamente unos 2.5 minutos en pasar desde la entrada a la salida.

Dispondremos de sensores inalámbricos que se situarán junto a los objetos que recorran la cinta, así estarán en comunicación continua con el PC, transmitiendo la temperatura instantánea en cada punto del recorrido.

## 2.2. Descripción del Hardware

### 2.2.1. Actuadores

#### Calentamiento. Radiadores

Los radiadores utilizados son elementos resistivos de cerámica. La potencia del calentador (o radiador) cerámico es de 500W y su resistencia es de  $100\Omega$ . Se monta tal como aparece en las figuras 2.2 sobre soporte (conectado a tierra) que sirve como reflector.

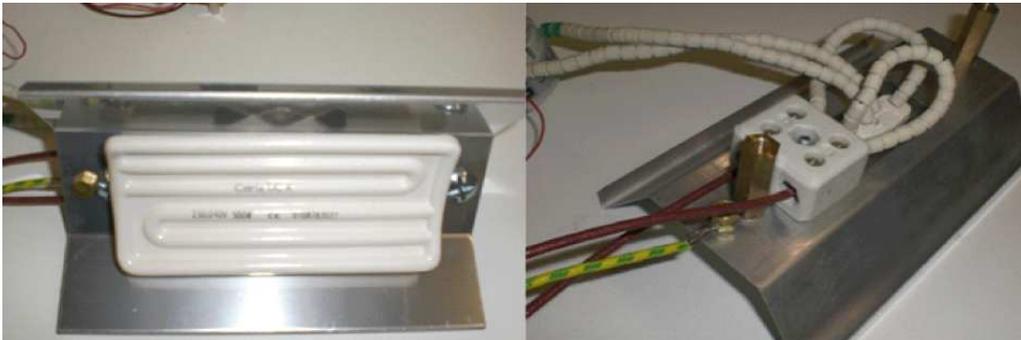


Figura 2.2: Radiador y reflector (Vista superior)

Estos se controlan mediante PWMs, figura 2.3, cuyas señales de control ( $V_c$ ), conectadas a la tarjeta DA, modulan la tensión de Red [6].

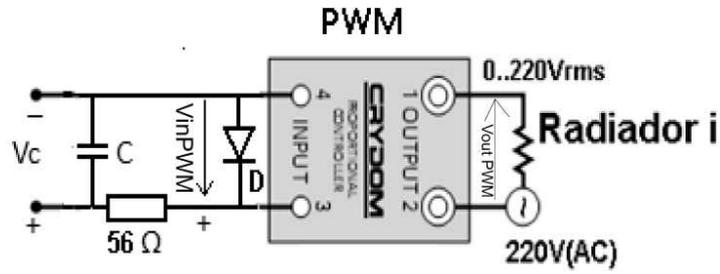


Figura 2.3: Esquema de conexión PWM-10PVC2425 y Radiador 500W/100Ω.

**Características de Funcionamiento**

La relación entre la señal de entrada de control desde PC ( $V_{in PWM}$ ) y la salida del PWM no es lineal y según las hojas de características responde a la función mostrada en la figura 2.4. Dicha no linealidad se puede caracterizar mediante un ajuste por mínimos cuadrados.

Tampoco es lineal la relación entre la tensión de salida del PWM ( $V_{out PWM}$ ) y la potencia en los radiadores ( $W_{Ri}$ ), ésta se puede caracterizar mediante la expresión  $(W_{Ri}) = \frac{(V_{out PWM})^2}{R_i}$ .

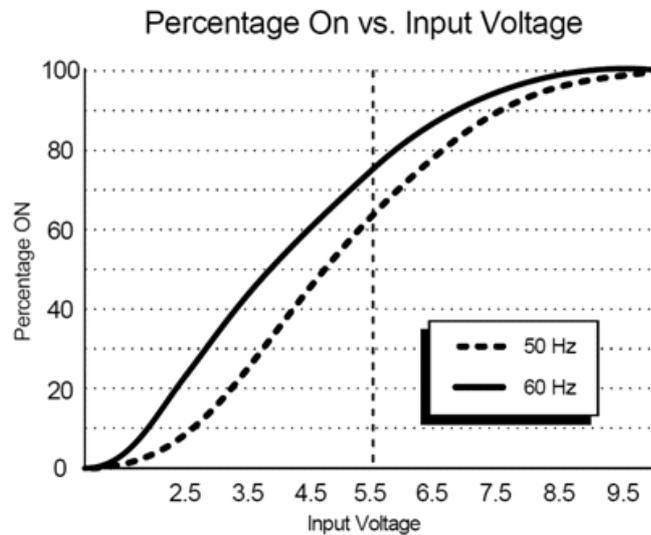


Figura 2.4: Relación Output/Input del PWM

### Modelo del Actuador

Ambas no linealidades se pueden modelizar con un único ajuste tal que  $W_{Ri} = F(V_{outPW})^2 = F(V_c)$  siendo  $W_{Ri} = \frac{V_{outPWM}^2}{Ri}$ . Por tanto, variando  $V_c$  y midiendo  $V_{outPWM}$  podremos obtener  $W_{Ri}$ , que relacionaremos con  $V_c$  mediante ajuste por mínimos cuadrados. Los rangos de la tensión de entrada y salida del PWM son:

- Rango de  $(V_{inPWM}) \rightarrow [2V \dots 10V]$ .
- Rango de  $(V_{outPWM}) \rightarrow [0V \dots 220V]$ .

**Recopilación de datos:** Cambiando el valor de salida de la DAQ a través de PC (Matlab, C, Labview ...) se midió en cada cambio, mediante voltímetro, el valor eficaz de alterna recibido por calentador  $V_{outPWM}$  (salida del PWM); La secuencia de entrada por PC fue: de 5.2V a 10V, de 10V a 2.2V, de 2.2V a 10V y de 10V a 5V. Con pasos de 0.2V y 0.1V (indistintamente).

A partir de los datos recopilados obtenemos la relación entre la potencia eficaz de alterna suministrada al calentador  $W_{Ri} = \frac{V_{outPWM}^2}{Ri}$  con  $Ri = 100\Omega$  y la tensión de entrada al actuador ( $V_c$ ) que es la escrita por la DAQ de salida ( $V_{daq}$ ).

El ajuste se ha realizado con la función `polyfit` y comprobado con `polyval` de Matlab, [27]:

```
1 [P,S,M]=polyfit(Vc,W_Ri,n) % n es el orden del ajuste, P parámetros del ajuste
[M_Ri\_est,d]=polyval(P,[2:10],S,M) %, M(1) media y M(2) desviación
```

El orden del ajuste con menor error escogido es  $n = 8$ :

$$W_{Ri} = b_{1R1} \cdot \bar{V}_{daq}^n + b_{2R1} \cdot \bar{V}_{daq}^{n-1} + \dots + b_{n-1R1} \cdot \bar{V}_{daq} + b_{nR1}$$

donde:

$$\bar{V}_{daq} = \frac{V_{daq} - E(V_{daq})}{\sigma(V_{daq})}$$

$$E(V_{daq}) \equiv \text{media de } (V_{daq}) \quad \sigma(V_{daq}) \equiv \text{desviación estándar de } V_{daq}$$

(2.2.1)

Se utiliza  $\overline{Medida}$  para mejorar el algoritmo de ajuste. Puesto que es la medida centrada y escalada (basado en las funciones `polyfit` y `polyval` de Matlab. El código entero de este ajuste se puede ver en C.2.4.

Como ejemplo se presentan los resultados obtenidos para el Radiador 1 (ver figura 2.5), donde el Error medio en valor absoluto entre el Ajuste y la Función Real es  $4,3 \pm 0,076W$ , obtenidos a partir de:  $mediaErrorR(1)=mean(d)$   $stdErrorR(1)=std(d)$

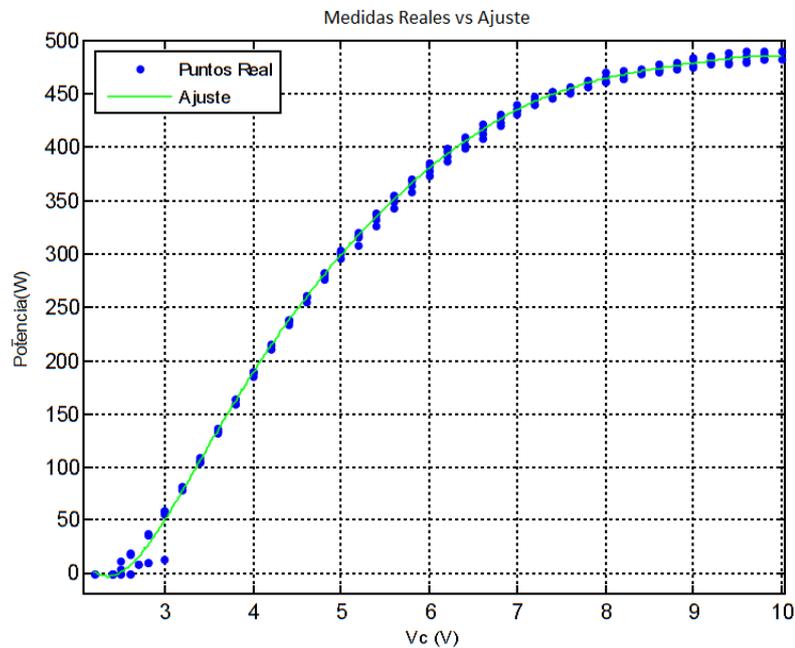


Figura 2.5: Ajuste que relaciona  $V_{daq}$  con la Potencia (en *Wattios*) aplicada al Radiador 3.

### Enfriamiento. Control de ventiladores (motores DC)

Se han preparado pequeñas placas con un control para motores DC pequeños. El diseño se basa en un amplificador operacional que suministra suficiente potencia, ver figura 2.6.

El diseño del circuito consiste en un seguidor no inversor con las protecciones (diodos y fusibles y filtro de la salida) instaladas según marcan las hojas de características del circuito L165, [40]. La señal de entrada ( $V_i$ ) es la señal de control procedente de la tarjeta DA y la salida ( $V_o$ ), ya con suficiente potencia, se envía al motor. El rango de tensión es el mismo para ambas señales: ( $V_i$ ) y ( $V_o$ )  $\rightarrow [3,5V \dots 10V]$ , debido a la zona muerta que presentan los motores DC.

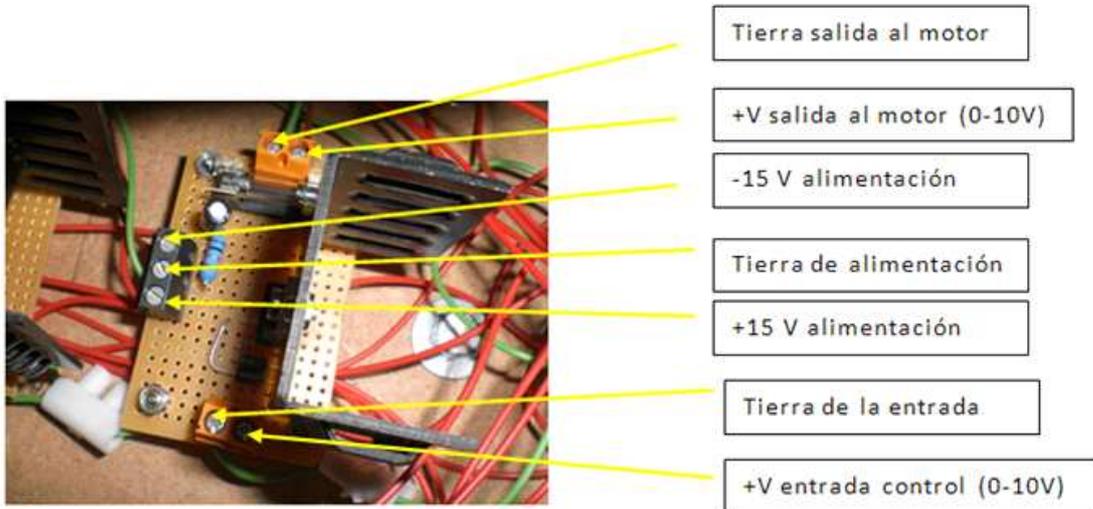


Figura 2.6: Imagen de los Ventiladores

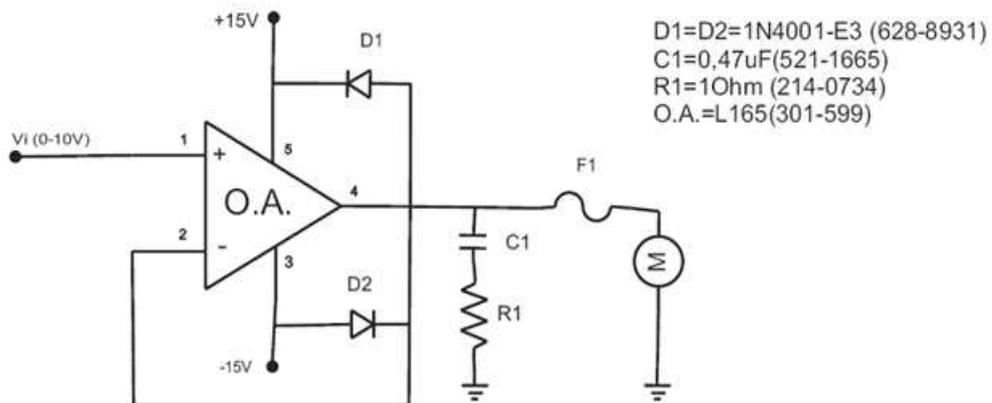


Figura 2.7: Circuito de potencia de los Ventiladores.

### Funcionamiento Ventiladores

El funcionamiento del circuito diseñado no es el deseado. Tiene problemas de calentamiento en el circuito accionador y cualquier cambio brusco (p.e. 3,5 a 10V) en la entrada positiva de alguno de los amplificadores operacionales provoca su ruptura inmediata.

Para intentar solucionar los problemas de calentamiento se han cambiado los amplificadores rotos por otros nuevos y se ha extendido pasta térmica en los radiadores. Al probarlos de manera escalonada no es capaz de hacer funcionar más de 5 ventiladores sin que alguno se rompa.

Por otro lado, resaltar que se inducen corrientes en los demás elementos electrónicos cuando los ventiladores están en funcionamiento. Así, para analizar y reparar esta parte se va a requerir más tiempo, por ello se va a obviar la ventilación y se analizará el funcionamiento del horno sólo con el sistema de calefacción.

### 2.2.2. Transductores

Para medir los cambios de temperatura se han utilizado termopares *tipo K*, y se ha preparado un circuito acondicionador diseñado para que la salida sea de voltaje y esté comprendida entre 2 y 10V que corresponden a la temperatura mínima y la máxima respectivamente (figura 2.8).

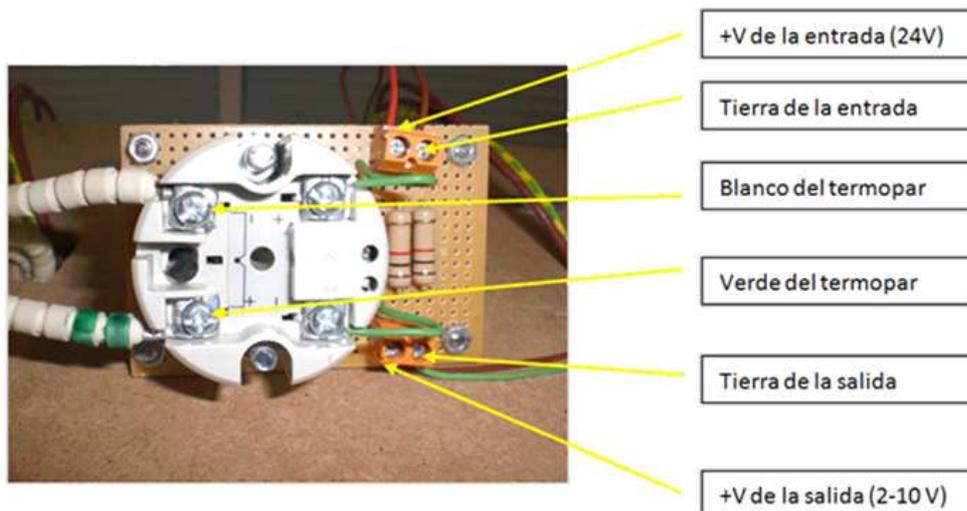


Figura 2.8: Circuito acondicionador de la señal del Termopar. Transmisor de intensidad TX104TC + Acondicionador basado en resistencia.

El TX104TC es un transmisor analógico de temperatura para instalación en cabeza sensora, alojado en un bloque de conectores estándar DIN. Se conecta a sensores de temperatura de termopar tipo *J*, *K*, o *T*, y transforma la señal de medida en una señal de salida de 4 – 20mA. El usuario selecciona en el transmisor el tipo de termopar. Se entrega con escala de 0 – 1,000°C para tipo *K*, pero se puede ajustar la escala mediante puentes soldados y potenciómetros, [34].

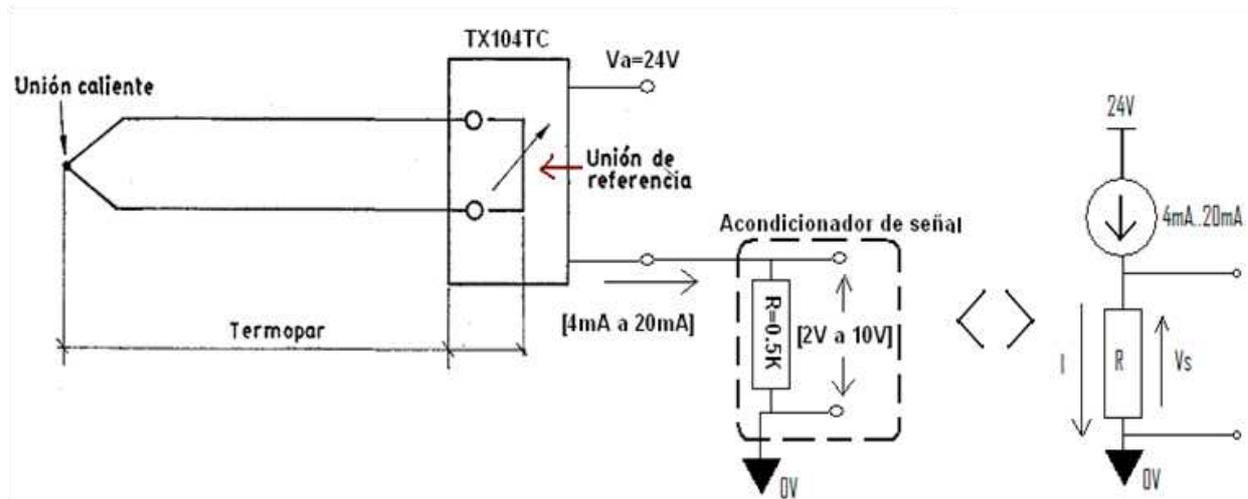


Figura 2.9: Esquema del transductor de Temperatura. Y su equivalente eléctrico.

### Rangos de Temperatura

**Rango de lectura y Precisión de los sensores de temperatura:** Los rangos se pueden variar cambiando las soldaduras internas de los transmisores. En un primer momento han sido instalados para leer desde  $-22/9$  (4mA) hasta  $465/527$  (20mA) por el técnico de laboratorio. Con esta configuración y según nuestro rangos de trabajo, de 20°C a 120°C aproximadamente, el rango de tensión varía entre 2V y 5V, tal y como se ve en la señal azul de la 2.10.

Si la precisión del sensor es del 0,1% del fondo de escala, sin considerar la unión fría (CJ), tal y como indican las hojas de características, por defecto tendríamos como mejor precisión  $(465 - 9) * 0,1\% = 0,456^\circ\text{C}$  (la peor=  $0,55^\circ\text{C}$ ).

Considerando que se puede modificar el rango para tener límites superiores de  $75/148$  (20mA), se podría conseguir una precisión de hasta  $(75 - 9) * 0,1\% = 0,066^\circ\text{C}$ . El problema surge en que si acotamos mucho el rango el sistema puede saturar al pasar de los 100°C, tal y como hace el Sensor 1 (Señal Roja de la Figura 2.10).

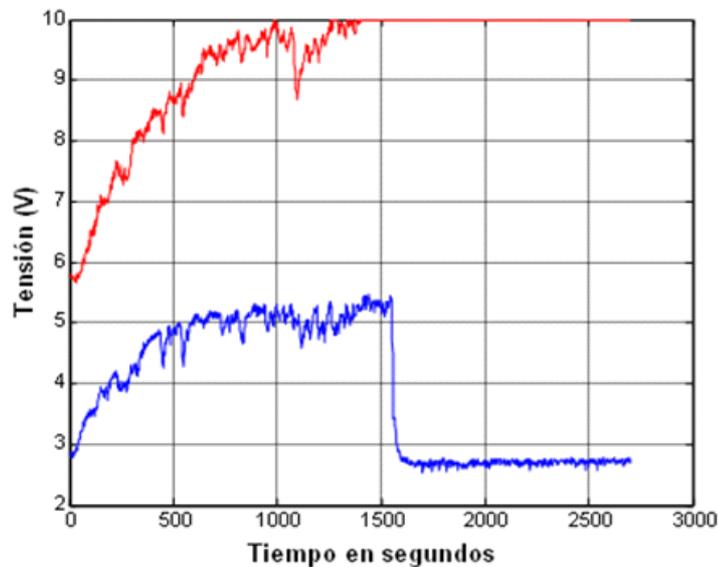


Figura 2.10: Tensión en dos sensores trabajando en rangos de entrada diferentes: Sensor 1 (Señal Azul). Rango de  $-22/9$  (4mA) hasta  $465/527$  (20mA). Sensor 2 (Señal Roja). Rango de  $-22/9$  (4mA) hasta  $75/148^{\circ}\text{C}$  (20mA). En tiempo 1500 segundos (aprox.) el Sensor 2 satura (Sensor 1 se saca del horno y mide temperatura ambiente).

Así, para que no sature, para que sea preciso y para que utilice mejor el rango de entrada de la tarjeta se ha optado por una solución de compromiso donde el límite superior se sitúa en  $138/189^{\circ}\text{C}$ . La peor precisión que se puede tener es de  $0,2^{\circ}\text{C} \leftarrow (189 + 22) \cdot 0,1\%$ . Por último destacar que las señales obtenidas en ambos rangos presentan un nivel de ruido muy alto, sobretodo cuando las medidas se hacen dentro del horno en funcionamiento.

### Escalado y Centrado: Ajuste de Temperatura Máxima y Mínima

Para ajustar los límites del rango se puso la unión caliente de cada una de las sonda en hielo y mediante la variación del potenciómetro ZERO ADJUST se obtuvieron valores de tensión cercanos a 2V (idénticos para los 6 transductores), dicho valor se considero nuestro mínimo. Al mismo tiempo se puso en contacto con el hielo un dispositivo de temperatura ya calibrado el cual nos indicó que la temperatura. Así pues cuando nuestro sensor nos suministre 2V la temperatura rondará los  $0^{\circ}\text{C}$ .

Posteriormente se hizo lo mismo para el límite superior. Las sondas se pusieron en contacto con una zona de alta temperatura del horno y se ajustaron a 9,5V mediante el potenciómetro SPAN ADJUST. El valor de temperatura que indicaba nuestro medidor ya calibrado era de  $185^{\circ}\text{C}$ .

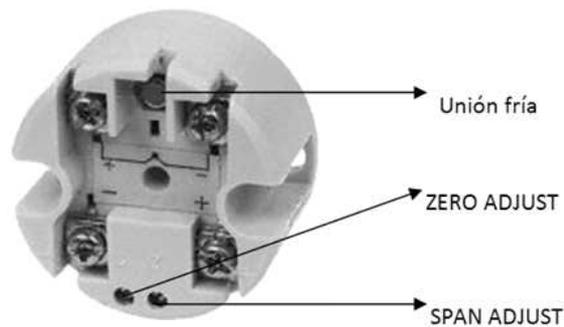


Figura 2.11: Transmisor analógico TX104TC de temperatura para instalación de termopar.

Si el límite superior y el inferior del rango, tanto en temperatura como en voltaje, de todos los sensores coincidiesen el escalado y centrado sería idéntico para todos los termopares. Con lo que la recta que relaciona la temperatura con la tensión del transductor sería la misma.

Pero como no disponemos de un calibrador de termopares conseguir tal coincidencia es muy complicada, en vez de esto utilizaremos la función polyfit de matlab que escala, centra y ajusta los datos a un polinomio de orden  $n$ .

### Calibración. Relacionar Voltios con Grados.

Para relacionar la tensión de los sensores con la temperatura se ha utilizado un multímetro, capaz de medir la temperatura mediante termopar. Dicho termopar se ha ido situando en el mismo punto que cada uno de los 6 sensores que hay en el túnel.

**Descripción del Experimento:** Registra tensión (mediante PC+DAQ) y temperatura (mediante Multímetro+Termopar). Se registraba el voltaje del sensor donde estaba situada la sonda mediante la DAQ y la temperatura que marcaba el multímetro de forma visual.

Posteriormente los datos se ajustan a una función.

- Para recopilar los datos se aplicaron 10V a los tres radiadores, seguido de un escalón de  $-10V$ , es decir se les ha quitado la alimentación.
- Para el ajuste por mínimos cuadrados se ha utilizado 59 lecturas: 51 lecturas de 24 a 80V y 8 lecturas de 80 a 48°C.

**Resultados para todos los sensores:** Como se indicó en el 2.2.2 si tuviéramos un buen escalado y centrado de los termopares (mediante los transmisores) las rectas coincidirían. Como no es posible esto se consigue mediante la función de ajuste polyfit de matlab. De esta forma si el ajuste es una recta de la forma  $Temp_{sensor_i} = a_{1_{sensor_i}} \cdot V_{sensor_i} + a_{2_{sensor_i}}$  tendremos que:

$$\begin{aligned}
 a_{1_{sensor_1}} &= a_{1_{sensor_2}} = a_{1_{sensor_3}} = a_{1_{sensor_4}} = a_{1_{sensor_5}} = a_{1_{sensor_6}} = \mathbf{a_1} = 13,56 \\
 a_{2_{sensor_1}} &= a_{2_{sensor_2}} = a_{2_{sensor_3}} = a_{2_{sensor_4}} = a_{2_{sensor_5}} = a_{2_{sensor_6}} = \mathbf{a_2} = 58,42 \\
 Temp_{sensor_i} &= a_{1_{sensor_i}} \cdot \overline{V}_{sensor_i} + a_{2_{sensor_i}} \overline{V}_{sensor_i} = \frac{V_{sensor_i} - \mathbf{E}(V_{sensor_i})}{\sigma(V_{sensor_i})} \quad (2.2.2) \\
 \mathbf{E}(V_{sensor_i}) &\equiv \text{media de}(V_{sensor_i}) \quad \sigma(V_{daq}) \equiv \text{desviación estadar de } V_{daq}
 \end{aligned}$$

Se utiliza  $\overline{Medida}$  para mejorar el algoritmo de ajuste. Puesto que es la medida centrada y escalada (basado en las funciones polyfit y polyval de Matlab).

$$\begin{aligned}
 \text{Sensor 1} &\rightarrow \mathbf{E}(V_{sensor_1}) = 3,97 \quad \sigma(V_{sensor_1}) = 0,50 \\
 \text{Sensor 2} &\rightarrow \mathbf{E}(V_{sensor_2}) = 4,31 \quad \sigma(V_{sensor_2}) = 0,60 \\
 \text{Sensor 3} &\rightarrow \mathbf{E}(V_{sensor_3}) = 4,38 \quad \sigma(V_{sensor_3}) = 0,65 \\
 \text{Sensor 4} &\rightarrow \mathbf{E}(V_{sensor_4}) = 4,11 \quad \sigma(V_{sensor_4}) = 0,55 \\
 \text{Sensor 5} &\rightarrow \mathbf{E}(V_{sensor_5}) = 3,98 \quad \sigma(V_{sensor_5}) = 0,50 \\
 \text{Sensor 6} &\rightarrow \mathbf{E}(V_{sensor_6}) = 4,23 \quad \sigma(V_{sensor_6}) = 0,60 \quad (2.2.3)
 \end{aligned}$$

En la figura 2.12 se muestra el ajuste para cada uno de los sensores.

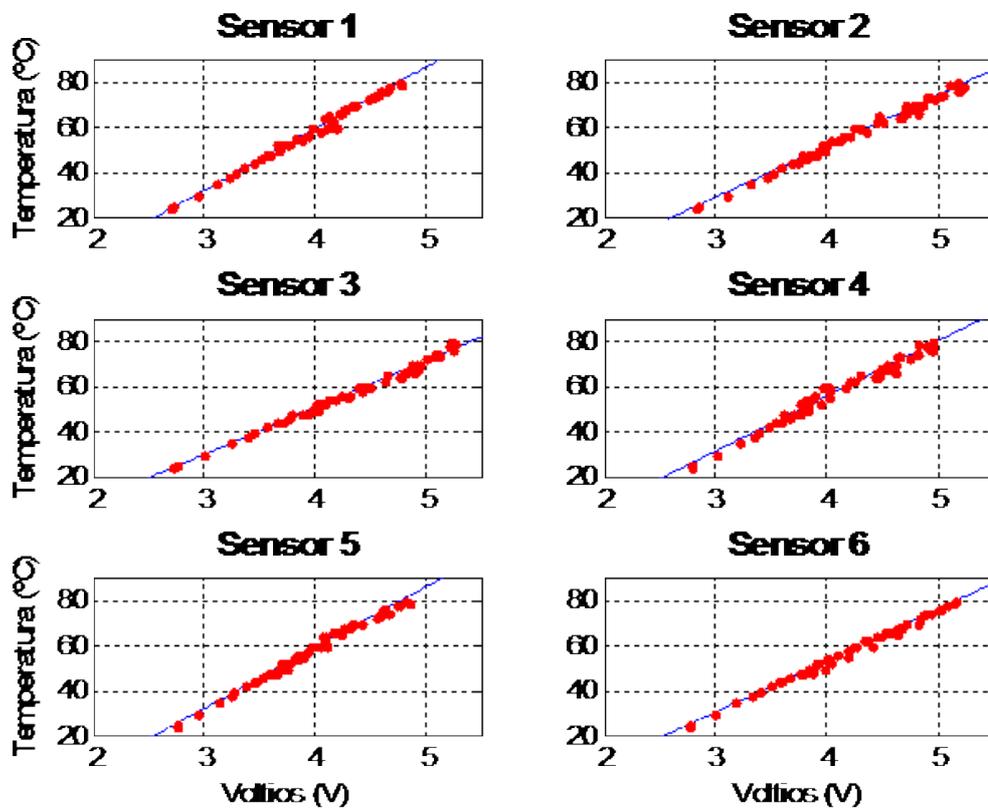


Figura 2.12: Ajuste de los datos

El error de cada uno de los sensores se puede ver en la tabla 2.1.

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6
$1,79 \pm 0,36^{\circ}\text{C}$	$1,76 \pm 0,25^{\circ}\text{C}$	$1,46 \pm 0,18^{\circ}\text{C}$	$2,51 \pm 0,42^{\circ}\text{C}$	$1,82 \pm 0,36^{\circ}\text{C}$	$1,15 \pm 0,17^{\circ}\text{C}$

Tabla 2.1: Error medio en valor absoluto entre la recta de ajuste y los valores reales

## Capítulo 3

# Interface con el computador

### 3.1. Tarjetas de adquisición de datos AD y DA

Se han instalado en el PC ACER-PROMETEO, dedicado al proyecto, dos tarjetas de adquisición de datos (DAQ's) que se conectan al proceso mediante cable y bornera, ver figuras 3.1.

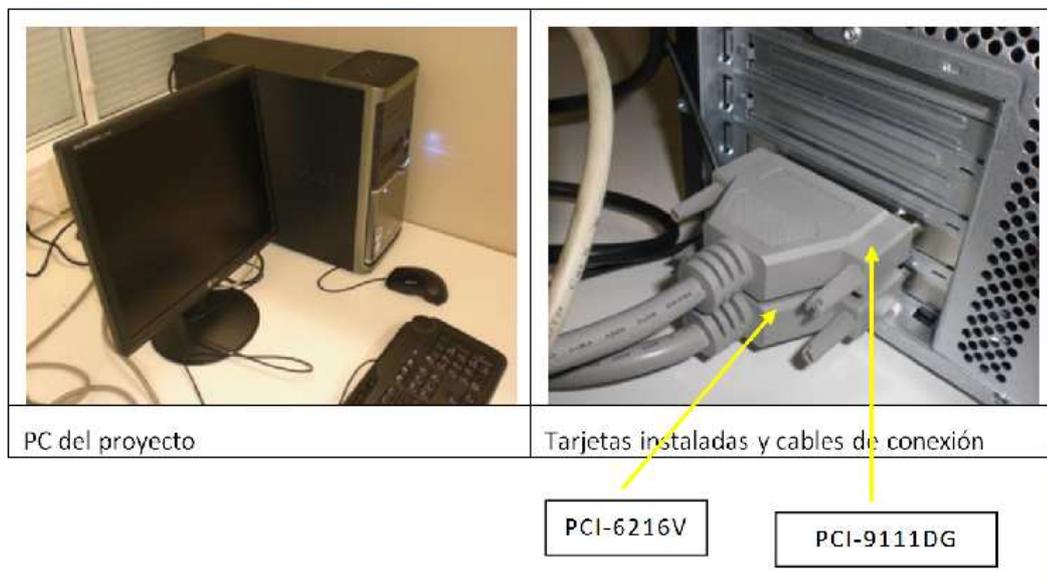


Figura 3.1: PC del proyecto, Tarjetas instaladas y cables de conexión

La tarjeta superior en el PC es la de entradas de datos analógicos (Sensores de Temperatura) y la inferior es la de salida (Acciones de Control).

**Entrada Analógica (A/D):** La tarjeta de entradas de datos es la PCI-9111DG, de ADLINK.

- Convertidor ADS7805/ADS7804 de sucesivas aproximaciones.
- Tiempo de conversión  $8\mu s$ .
- 16 entradas no diferenciales en voltaje.
- Resolución 12 – bit.
- Rangos de Entrada de la señal Analógica:
  - ▷ Bipolar  $\pm 10V$ ,  $LSB = \frac{FondoEscala}{2^{Resolucion}} = \frac{20}{2^{12}} = 0,0049V$ .
  - ▷ Bipolar  $\pm 5V$ ,  $LSB = \frac{FondoEscala}{2^{Resolucion}} = \frac{10}{2^{12}} = 0,0024V$ .
  - ▷ Bipolar  $\pm 2,5V$ ,  $LSB = \frac{FondoEscala}{2^{Resolucion}} = \frac{5}{2^{12}} = 0,0012V$ .
  - ▷ Bipolar  $\pm 1,25V$ ,  $LSB = \frac{FondoEscala}{2^{Resolucion}} = \frac{2,5}{2^{12}} = 0,0006V$ .
  - ▷ Bipolar  $\pm 0,625V$ ,  $LSB = \frac{FondoEscala}{2^{Resolucion}} = \frac{1,25}{2^{12}} = 0,0003V$ .
- Frecuencia de muestreo máxima  $100KHz$  (acumulativa).

**Salida Analógica (D/A):** La tarjeta de señales de control es la PCI-NuDAQ6216V, de ADLINK.

- 16 canales.
- Conversor: B.B PCM56U o equivalente, tipo Monolithic multiplying.
- Resolución: 16-bit (14-bit garantizados)
- Rango de salida en voltaje:  $\pm 10V$ .
- Tiempo de establecimiento:  $2\mu s$  second ( $-10V$  to  $+10V$ ).
- Estado inicial:  $0V$  (después de resetear o power-on).
- Error lineal diferencial:  $\pm 0,001\%$  en todo el rango.

Las características técnicas completas de ambas tarjetas se puede encontrar en los respectivos manuales, [21] y [20]. En la figura 3.2 se muestra el esquema de conexiones de ambas tarjetas.

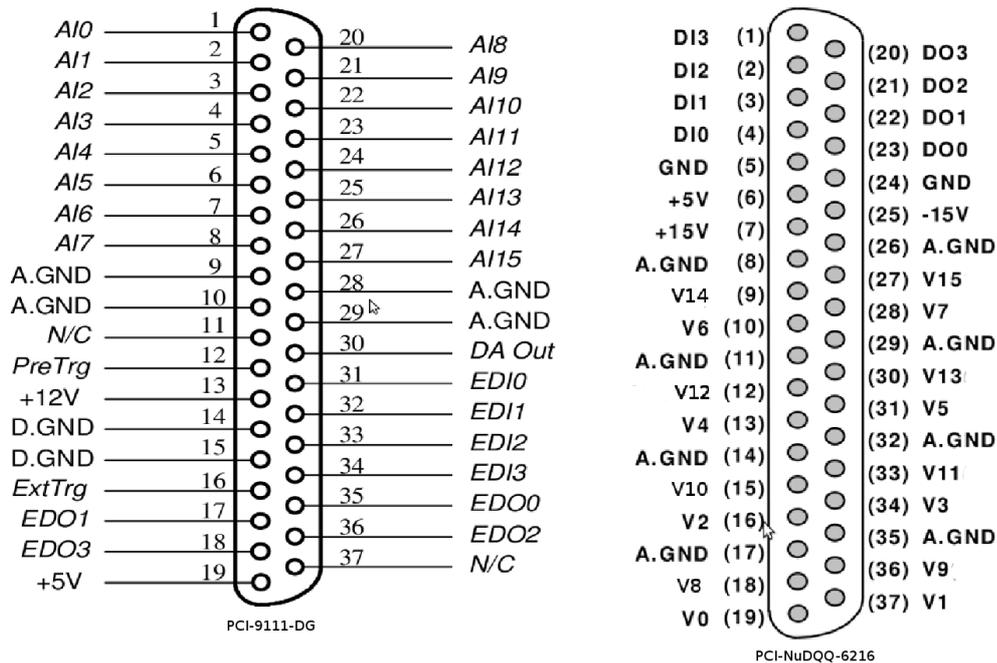


Figura 3.2: Configuración de conexiones. Izquierda PCI-9111DG. Derecha PCI-NuDAQ6216V

### 3.2. Comunicación con la tarjeta de adquisición de datos

Para interactuar con el sistema de adquisición se ha utilizado *Matlab* y *LabVIEW*. En ambas plataformas se necesita la instalación de drivers de adaptación para que se puedan comunicar con las tarjetas.

En este punto se presentan las soluciones utilizadas:

- a.- Librería de ADLINK para matlab "Data acquisition library for matlab"(DAQ-MTLB) que se puede descargar directamente desde <http://www.adlinktech.com/MAPS/DAQ-MTLB.html>. Manual de usuario: [18].
- b.- DAQ-LVIEW PnP es un conjunto de VIs de DAQ de LabVIEW que dan soporte a un rango completo de tarjetas de adquisición de ADLINK. Manual de usuario: [19].

Matlab se ha utilizado para analizar los datos obtenidos, identificar los modelos del horno y simular el sistema de control mediante Simulink. Mientras que LabVIEW se ha utilizado para diseñar una interface de usuario (HMI) e implementar el bucle de control, cargando los resultados obtenidos en Matlab mediante el bloque VIs MathScript, el cual permite programar m-files dentro de Block Diagram, [23].

### 3.2.1. Driver DAQ-MTLB. Data Acquisition Toolbox Adapter

El adaptador DAQ-MTLB es un Toolbox-Adptador-driver estándar de adquisición de datos de Matlab que da soporte a una línea de tarjeta de adquisición de datos (DAQ's) de ADLINK.

DAQ-MTLB está basada en la MATLAB DAQ Toolbox interface y se utiliza para el desarrollo de drivers. Una vez instalado el MATLAB DAQ Toolbox, se puede controlar desde matlab las ADLINK DAQs. Antes de utilizar DAQ-MTLB, tenemos que tener instalado:

- Al menos una tarjeta de ADLINK, que sea soportada por DAQ-MTLB.
- MATLAB 6.5 o superiores.
- MATLAB Data Acquisition Toolbox 2.2 o superiores.

#### Componentes de la Toolbox

La toolbox de adquisición de datos consiste en tres componentes: funciones M-File, el motor de adquisición de datos y el adaptador de drivers, que permiten pasar información entre nuestros hardware de adquisición de datos y MATLAB, observar figura 3.3.

#### M-File Functions

Para realizar cualquier tarea se hace mediante llamada a funciones M-file desde Matlab. Entre otras cosas nos permite crear objetos los cuales proveen acceso a las capacidades del software.

#### The Data Acquisition Engine

The data acquisition engine (o simplemente motor ) es un MEX-file que almacena los objetos del dispositivo y las propiedades asociadas al mismos que controlan su comportamiento: sincronización de eventos, almacenamiento de datos.

**El flujo de adquisición de datos.** El flujo de adquisición de datos desde el dispositivo hasta el motor consiste en dos pasos, ver figura 3.4:

- Adquisición desde el hardware que se almacena en el motor de adquisición.
- Extracción de los datos desde el motor que se almacenan en MATLAB.

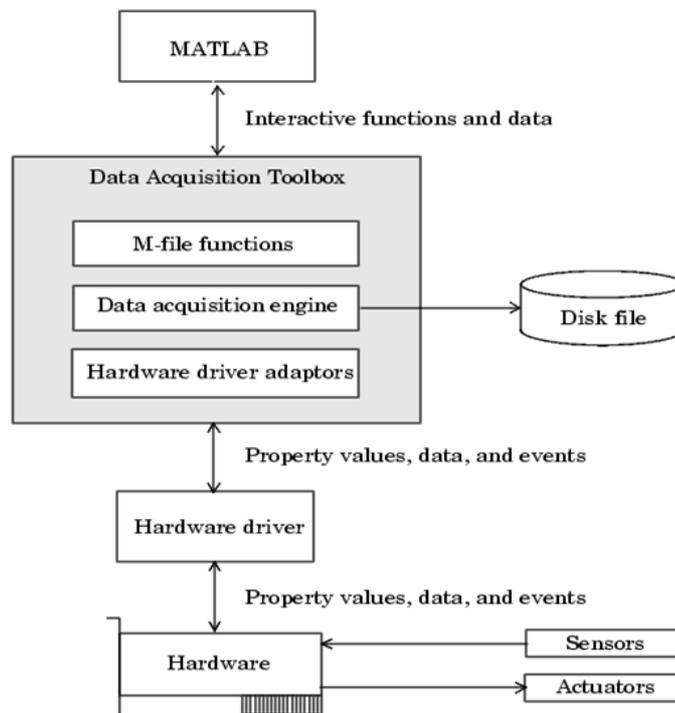


Figura 3.3: Esquema de Adquisición de datos MFile, Motor y adaptador

El almacenamiento de datos no se hace de forma automática a workspace de MATLAB, para ello se debe extraer explícitamente desde el motor de adquisición mediante la función `getdata`.

**El flujo de datos de salida** El flujo de datos de salida desde el motor hasta el dispositivo hardware consiste en dos pasos, ver figura 3.5:

- Desde MATLAB poner en cola en el motor.
- Escribir los datos en el dispositivo.

Para ello, antes se deben poner en cola en el motor con la función `putdata`. La cantidad de datos en cola dependerá de la memoria disponible, el número de canales del hardware y el tamaño de cada muestra.

### The Hardware Driver Adaptor

The hardware driver adaptor (o simplemente adaptor) es la interface entre el motor y el driver del hardware. Su propósito principal es pasar información desde MATLAB al hardware a través de sus

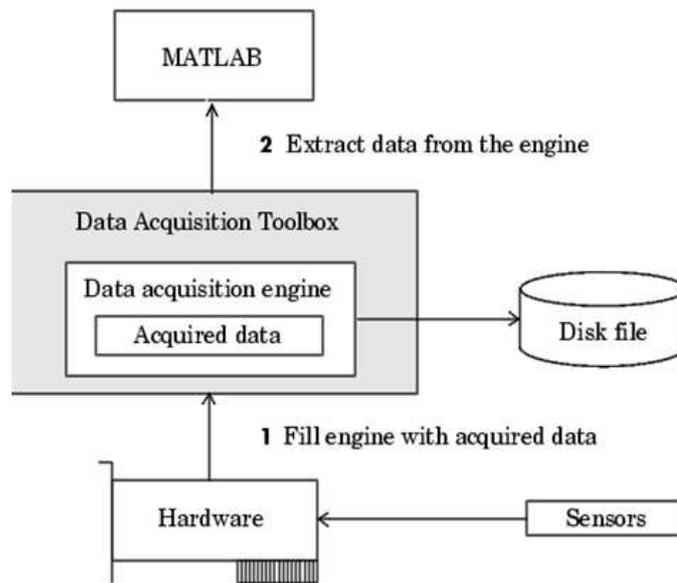


Figura 3.4: Esquema de flujo de adquisición de datos.

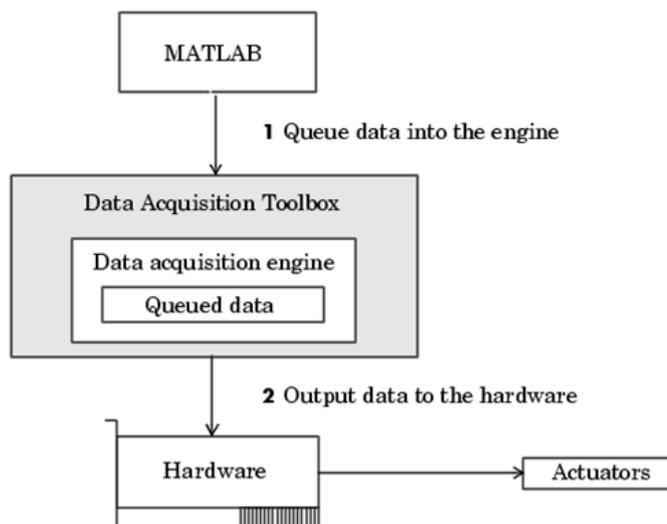


Figura 3.5: Esquema de flujo de datos de salida

drivers. En el anexo B se muestran los ejemplos utilizados para la programación de las rutinas de adquisición.

Nota: Los drivers no son parte de la Data Acquisition Toolbox.

### 3.3. DAQ-LVIEW PnP. Drivers para comunicación con Labview

DAQ-LVIEW PnP .exe se puede descargar de la pagina web de ADLINK. Se ejecuta el archivo y tendremos acceso a una serie de VI que permite la comunicación directa con Labview, ver figura 3.6.



Figura 3.6: Bloques VI para acceso a las DAQ

En nuestro caso utilizaremos los bloques VIs de los submenús AI (Analog Input) y AO (Analog Output). Como entrada analógica utilizaremos el bloque **AI Sample Channels** y **AO Update Channels**. Los parámetros a definir en **AI Sample Channels** son:

1. **input config**: configuración de la tarjeta, en nuestro caso será la básica, no `change:0`.
2. **reference source** (Device input): localiza de la tarjeta en el PC (conexión PCI):1.
3. **channels**: Número de canales de entrada 0:5, corresponde al número de sensores a leer.
4. **high limit input**:10 y **low limit input**:0. Define el rango de actuación.
5. **sub type input**: indica el subtipo de la familia de la tarjeta, en nuestro caso: 0.
6. **value**: valor del vector de entrada.

Los parámetros a definir en **AI Sample Channels** son:

1. **sub type input**, indica el subtipo de la familia de la tarjeta, en nuestro caso: 0.
2. **reference source** (Device output): localiza de la tarjeta en el PC (conexión PCI):2.
3. **channels**: Número de canales de entrada 0:10, corresponde al número de actuadores.
4. **high limit input**:10 y **low limit input**:3.5. Define el rango de actuación.
5. **value**: valor del vector de salida.

En la figura 3.7 se muestra la configuración de comunicación que utilizaremos para implementar los bucles de control. En el anexo 8 se puede ver como se han utilizados estos bloques para generar los Scadas.

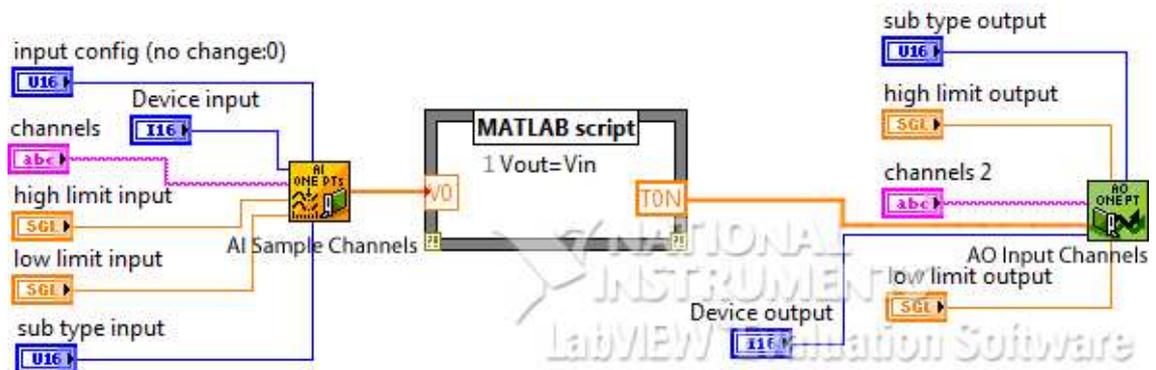


Figura 3.7: Esquema de comunicaciones en Labview

El MATLAB script permite ejecutar todas las funciones del último Matlab que tengamos instalado dentro de Labview.



## **Parte II**

# **Identificación y mejoras**



## Capítulo 4

# Identificación

### 4.1. Introducción

La identificación llevada a cabo en este capítulo no es completa, estando limitada por los problemas con la instrumentación. Así pues considerando que en este punto aun estamos en fase de diseño de la plataforma el objetivo será comprobar que el comportamiento de la plataforma (sensores, radiadores ,materiales, comunicaciones, plataforma de software, . . . ) funcionan correctamente. Por tanto, considerando que los ventiladores tienen problemas de funcionamiento, el modelo indentificado tendrá como variable de salida la señales de los termopares (temperatura) y como entrada la potencia suministrada a los radiadores.

### 4.2. Procedimiento de identificación del Sistema. Concepto y Etapas

Se entiende por identificación de sistemas a la obtención, experimental o mediante primeros principios, de la estructura y de los parámetros de un modelo que reproduce con suficiente exactitud las características dinámicas de interés del proceso.

En nuestro caso la obtención del proceso será experimental, con lo que necesitaremos [42], [24]:

- Un conjunto de datos, el cual obtendremos de las señales de los termopares(6 dentro del horno y 3 sobre los radiadores), y las tensiones aplicadas a los calentadores y a los ventiladores.

- Una Estructura de modelo cuyos parámetros se ajusten a partir de los datos recogidos.
- Y una regla para evaluar los modelos candidatos (Validación del modelo).

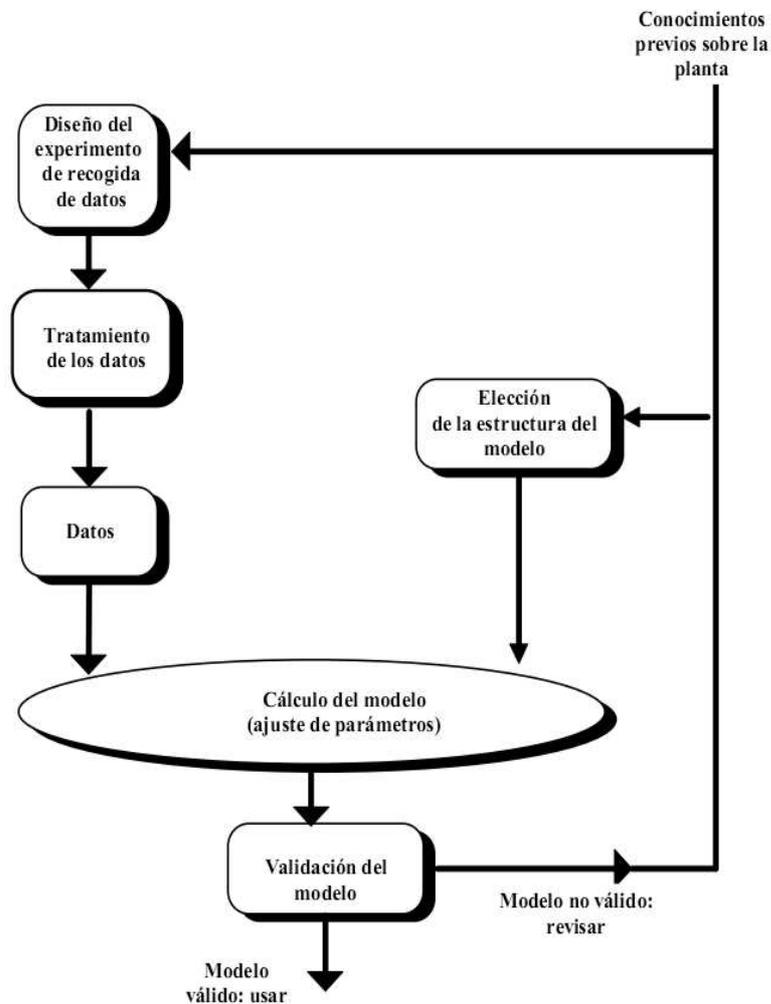


Figura 4.1: Etapas para obtener un modelo válido.

### 4.3. Definir el Comportamiento del Sistema

Para seleccionar la estructura del modelo es conveniente definir el comportamiento del sistema. Para ello asumiremos que nuestro proceso tiene las características de un LTI (Lineal Time Invariant), que es la aproximación más utilizada en la práctica y en literatura. Esta idealización asume que nuestro sistema cumple:

- **Linealidad:**

1. La respuesta del proceso es proporcional a la entrada, esto es si con  $u(t) \rightarrow y(t)$  con  $\alpha \cdot u(t) \rightarrow \alpha \cdot y(t)$  ( $\alpha$  es  $\mathbb{R}$ );
2. El efecto de varias entradas es aditivo. Si tenemos que para  $u_1(t) \rightarrow y_1(t)$  y que para  $u_2(t) \rightarrow y_2(t)$ , entonces para  $u = \alpha_1 \cdot u_1 + \alpha_2 \cdot u_2$  la respuesta es  $y = \alpha_1 \cdot y_1 + \alpha_2 \cdot y_2$ .

- **Invariancia temporal:** Implica que los parámetros y funciones no varían con el tiempo.

En el caso de nuestro prototipo como su propósito no implica un uso continuo se ha considerado que cumple el requisito de invarianza temporal. Sin embargo, al ser un proceso térmico no podremos asumir el requisito de linealidad en todo su rango. Así que para trabajar con el sistema como si fuese un LTI aplicaremos las siguientes técnicas de linealización:

- **Acotaremos los rangos de entrada a los actuadores.** De esta forma fijaremos una zona de funcionamiento sin zonas muertas.
- **Trabajaremos con incrementos sobre un punto de funcionamiento.** Para tener un modelo que consideraremos lineal en un rango acotado.
- **Utilizaremos como entrada el cuadrado de la tensión eficaz aplicada a los radiadores.** Ver sección 2.2.1.

#### 4.3.1. Elección de la Estructura

Como se indico en la sección 4.2 la identificación se realizara de forma experimental. Para ello el proceso será considerado como una caja negra (Black-box) a la cual le introducimos entradas y obtenemos salidas, que más tarde relacionaremos mediante la estructura seleccionada.

Asumido que nuestro proceso es LTI (ver sección 4.3), las estructuras más utilizadas de identificación se dividen en dos grupos: Métodos Paramétricos y Métodos No Paramétricos.

**Métodos Paramétricos:** se describen en el dominio discreto, puesto que los datos para la identificación se han obtenido por muestreo. Consiste en el ajuste de una estructura de modelo a los datos obtenidos mediante variación de parámetros, lo que implica la necesidad de algún conocimiento previo de la estructura del proceso.

Principales de estructuras basadas en estos métodos:

- ARX (Autoregressive extra input).

- ARMAX (Autoregressive moving average extra input): es uno de los más utilizados, puesto que describe mejor el ruido que el ARX.
- ARIMA(X): Es una versión de ARMA (X) con un integrador en la descripción del ruido. El cual es útil para describir sistemas con ruidos de dinámica lenta.
- Otros: ARARMAX, FIR (finite impulse response), BJ (Box-Jenkins), OE (output error),...

**Método No Paramétricos:** se denominan así, porque no emplean (explícitamente) un vector n-finito de parámetros que busquen la mejor descripción. En estas técnicas se asume muy poco acerca de la estructura del sistema, sólo si es lineal o no.

La identificación no-paramétrica se utiliza cuando se tiene una visión muy limitada de los mecanismos que rigen el proceso y por tanto resulta muy difícil determinar el sistema de ecuaciones que rige el proceso. En tales casos se evalúa el sistema con estímulos conocidos, registrando la respuesta del sistema ante tales entradas. Las principales estructuras basadas en estos métodos son:

- Ajuste de la respuesta temporal: Caracteriza la respuesta temporal del sistema frente a entrada conocida. Mención especial para el modelo de primer orden con retardo, que es uno de los más utilizados en la industria. Este modelo está basado en el hecho de que la mayoría de procesos se comportan de forma sobre-amortiguada y su respuesta ante escalón puede ser aproximada mediante cuatro parámetros. Si el proceso es sub-amortiguado se deben considerar otros modelos de orden superior.
- Ajuste de la respuesta en frecuencia: Las señales de entrada son senoidales, que nos permitirá representar la respuesta frecuencial mediante el diagrama de Bode. Y mediante aproximación de la frecuencia de corte y el retardo se puede obtener un modelo similar al del ajuste de la respuesta temporal. Esta técnica es útil para procesos rápidos.
- Otros: Análisis mediante correlación de señales, Análisis de Fourier,...

### Representación del Sistema

Para caracterizar el sistema a partir de los datos experimentales lo haremos por medio de representación *Entrada/Salida*. Esta representación solo utiliza los valores de entrada (a los actuadores) y salida (de los sensores) para la identificación (posteriormente si las estrategias de control lo exigimos podremos realizar las transformaciones necesarias para cambiar de representación).

Se pueden distinguir tres tipos de representación entrada/salida:

1. Ecuaciones en diferencias:  $d_n \cdot \frac{d^n y}{dt^n} + \dots + d_2 \cdot \ddot{y} + d_1 \cdot \dot{y} = n_n \cdot \frac{d^n u}{dt^n} + \dots + n_2 \cdot \ddot{u} + d_1 \cdot \dot{u}$
2. Representación polinomial:  $D(s) \cdot Y(s) = N(s) \cdot U(s)$ .
3. Función de transferencia:  $Y(s) = D^{-1}(s) \cdot N(s) \cdot U(s) = G(s) \cdot U(s)$ . (Matriz de transferencia en el caso MIMO).

En nuestro caso utilizaremos la Función de transferencia, o mejor dicho la Matriz de transferencias puesto que trabajamos con un sistema multivariable. Las principales características de esta representación son:

- Sus elementos representan la conexión de una-entrada a una-salida, facilitando experimentos de identificación y validación parcial del proceso. De la misma forma, en el caso multivariable puede representar la interacción entre diferentes entradas y diferentes salidas.
- Se pueden considerar retardos fácilmente  $e^{-\theta \cdot s}$ .
- Puede ser utilizado como operado para combinar subsistemas.
- Los polos y ceros aparecen explícitamente, lo que permite identificar fácilmente el comportamiento del sistema.
- El número de parámetros está minimizado.
- La representación es única.
- Existen aproximaciones *ad-hoc* para el diseño de sistemas de control.

Sin embargo

- Sus elementos son funciones racionales.
- Algunas propiedades globales no aparecen claramente reflejadas.
- No permite analizar por completo las características del sistema.
- Sólo es válido para sistemas lineales.

#### 4.4. Características del Modelo del Horno

En nuestro caso trabajaremos con un proceso linealizado alrededor de un punto de funcionamiento. En un principio el punto de funcionamiento se ha seleccionado  $5V$  (o  $300W$ ) en todos los

radiadores para tener un rango de actuación más o menos lineal, desde 100W a 450W (ver figura 2.5) evitando además saturaciones.

En cuanto a la estructura que utilizaremos estará basada en métodos no-paramétricos, concretamente seleccionaremos el modelo de primer orden con retardo, puesto que el proceso se comporta de forma sobre-amortiguada, ver figura 4.2 y la respuesta ante escalón se puede aproximar mediante cuatro parámetros  $(K, \tau, n, \theta)$ , [30], [2]:

$$G(s) = (Y(s))/(U(s)) = \frac{K}{(\tau s + 1)^n} e^{-\theta \cdot s} \quad (4.4.1)$$

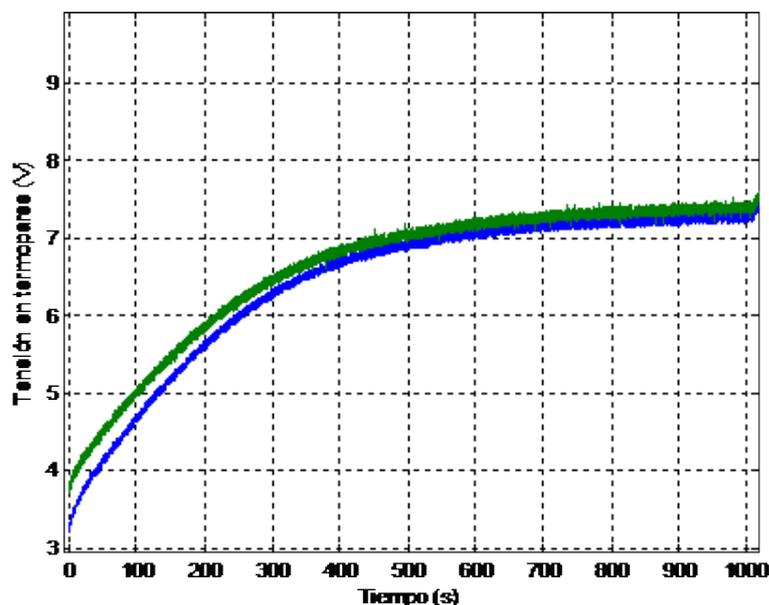


Figura 4.2: Transitorio de la Temperatura. Señales de los termopares situados en los radiadores, por ello apenas se observa retardo.

En sistemas MIMO, este experimento se puede realizar para cada entrada manteniendo el resto constante. Por lo tanto y resumiendo, las razones para seleccionar esta estructura son:

- Por simplicidad. Como primera opción intentar lo más simple.
- Porque estudia la relación Entrada/Salida con técnicas basadas en el dominio del tiempo.
- Porque se pueden parametrizar en términos de función de transferencia continua:  $Y(s) = G(s) \cdot U(s)$ .

- Por no incorporar ninguna interpretación física: 'Black Box models + Métodos no-paramétricas'.
- Por ser la técnica no-paramétrica más directa para identificación de sistemas lineales.
- Por la comodidad para generar la matriz  $G(s)$  multivariable.

## 4.5. Diseño del experimento

### 4.5.1. Elección del tipo de entrada

En general para sistemas lineales es suficiente con dos niveles de entrada, que contenga el mayor número de frecuencias posibles (figura 4.3.a). Si quisiéramos identificar las no-linealidades deberíamos utilizar entradas con más niveles (figura 4.3.b).

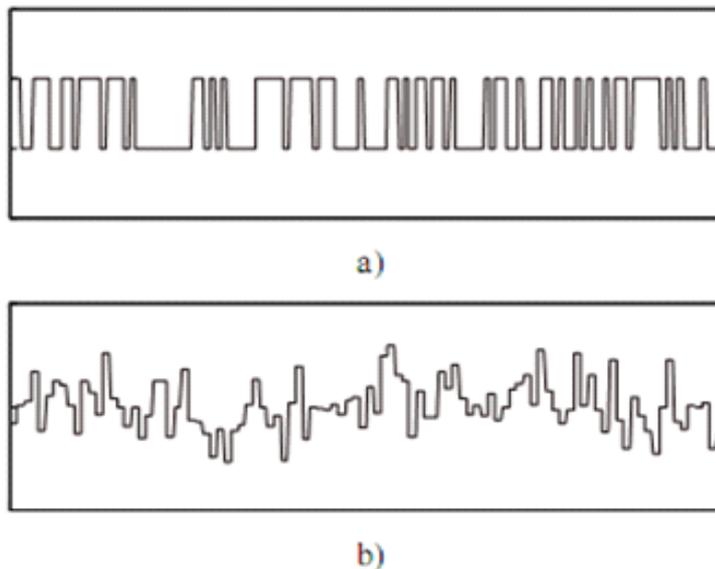


Figura 4.3: a) Dos niveles de entrada. b) Más de dos niveles.

En la estructura que hemos seleccionado (Expresión 4.4.1) es suficiente con identificaremos el modelo mediante una única entrada escalón. Y puesto que el sistema va a trabajar en torno a un determinado punto de funcionamiento (por ejemplo:  $Canal\_DAQ\_Radiador1= Canal\_DAQ\_Radiador2= Canal\_DAQ\_Radiador3=5V$ ), dichos escalones partirán desde ese punto.

Se utilizarán pequeños escalones para garantizar el comportamiento lineal, (aunque si son demasiado pequeños tendrán un ratio de señal ruido reducido, [42]).

#### 4.5.2. Elección del periodo de muestreo

Para identificar el sistema utilizaremos como periodo de muestreo  $0,1 \text{ segundos}$  ( $F_s = 10 \text{ Hz}$ ) que aunque aportará datos redundantes sobre el comportamiento del proceso, nos permitirá detectar posibles dinámicas inesperadas (derivadas térmicas en los termopares, ruido electro-magnético,...) que podremos filtrar digitalmente.

*Nota:* Por otro lado, puesto que el método de identificación seleccionado modela una función continua, con  $10 \text{ Hz}$  podemos ser más precisos en la identificación de la gráfica de la respuesta transitoria, por ejemplo en el retardo. Otra ventaja es que posteriormente se puede obtener el sistema discretizado a la frecuencia deseada.

#### 4.5.3. Recogida de datos

Para recoger los datos llevamos el sistema a un punto de funcionamiento (por ejemplo: todos los calentadores a  $5 \text{ V}$ ). Como el sistema de ventilación no funciona correctamente estos permanecerán parados.

Una vez estabilizada por completo la tensión de todos los sensores, aplicamos un escalón de tensión a uno de los radiadores hasta que se estabilicen de nuevo todos los termopares. Esta operación se repetirá para cada uno de los radiadores. Un ejemplo de respuesta ante un escalón de  $3 \text{ V}$  en el Radiador 2 se puede ver en la Figura 4.4.

La adquisición de los experimentos se realiza con el bucle `for` que se muestra a continuación (para ver el código completo ver sección C.1).

```

Texp=100.0;
2 tmuestreo=0.1;
for i=1:1:Texp/tmuestreo;
4     t(i,:) = clock;
     if (i < (0.5*Texp/tmuestreo))
6         accion_control_rad(i,:)=[0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 5.0 5.0 5.0];
     else
8         accion_control_rad(i,:)=[0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 5.0 8.0 5.0];
end
10 wait(ai_device,0.6*tmuestreo);

```

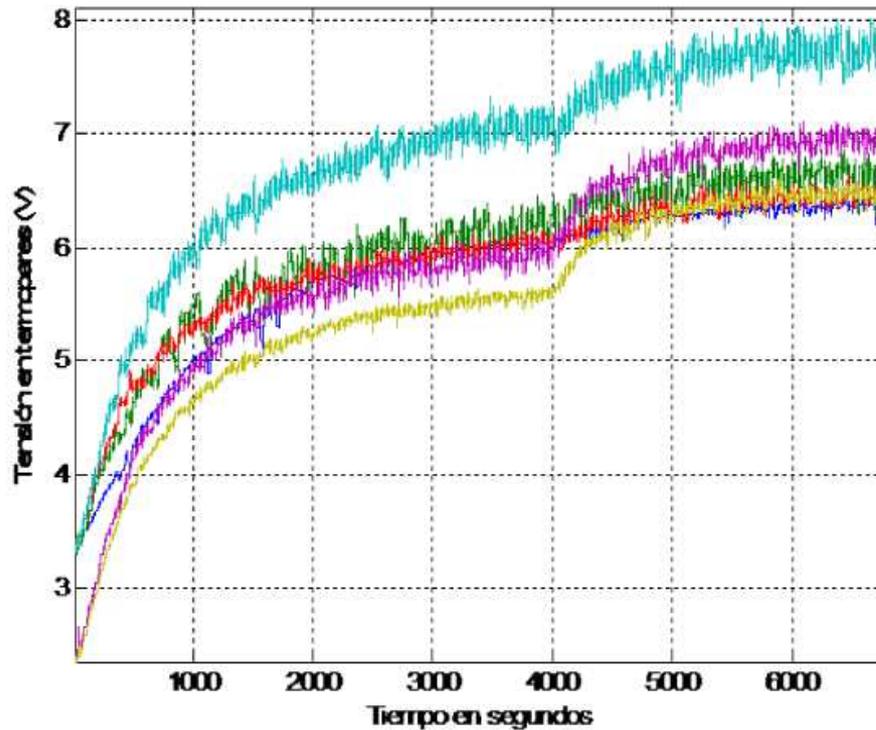


Figura 4.4: Respuesta frente a escalón de 3V en el Radiador 1  $R_1$ , desde punto de funcionamiento  $R_1 = R_2 = R_3 = 5V$

```

input_rad(i,:) = getsample(ai_device);
12 putsample(ao_device, accion_control_rad(i,:));
while etime(clock, t(i,:)) < tmuestreo end;
14 end;

```

#### 4.5.4. Tratamiento de datos

Los datos obtenidos tras el experimento son tensión aplicada a la tarjeta de salida PCI-6216-GL ( $V_{daq}$ ) y tensión de los sensores a la tarjeta de entrada PCI-9111DG/HR ( $V_{termopares}$ ). Los pasos para adecuar los datos obtenidos son:

**Datos de Salida o Lectura (sensores).**

1. Paso de las señales obtenidas por los termopares de Voltios a Temperatura ( $^{\circ}\text{C}$ ), ver sección 2.2.2 el código se puede encontrar en la sección C.2.3 (función V2C.m):

$$Temp_{sensor_i} = a_1 \cdot V_{sensor_{i_{hat}}} + a_2$$

2. Filtramos las señales para obtener una señal más limpia. Para el diseño del filtro utilizamos la función `butter` de Matlab, a la que le pasaremos el `Orden` del filtro y la relación entre la frecuencia de corte y la frecuencia de muestreo ( $F_c/F_s$ ), teniendo en cuenta que se debe cumplir  $F_c \leq 2 \cdot F_s$  ( $F_c \equiv$  frecuencia de corte y  $F_s \equiv$  frecuencia de muestreo). En nuestro caso las señales tiene mucho ruido siendo el ancho de banda bastante grande con lo que necesitaremos un relación ( $F_c/F_s$ ) mayor que 2. Se ha seleccionado 800 lo que indica una mala adquisición. A continuación se muestra el código utilizado para el diseño del filtro y el filtrado de la señal con `filfilt`, (el código se puede encontrar la sección C.3) y el resultado en la figura 4.5:

```

Fc=Fs/800;
Orden=5;
[b,a]= butter(Orden,Fc/Fs);
Tf = filtfilt(b, a, T);

```

#### Datos de Entrada (actuadores).

1. Por último trataremos los datos de entrada al actuador ( $V_{daq}$ ). Esto consiste en obtener el equivalente de tensión de salida en el *PWM* a partir de la señal que enviamos desde el PC ( $V_{daq}$ ), (ver sección 2.2.1) y posteriormente pasarlo a potencia mediante el cuadrado de la tensión:

```

[v,d]=polyval(P,vc,S,M)';
v2=v.^2;

```

El código se puede encontrar en la sección C.3.

## 4.6. Cálculo del Modelo

La identificación de modelo de primer orden frente a entrada escalón, se ha realizado en base a la obtención de los parámetros  $\tau$ ,  $K$ , representados en la figura 4.6, y  $\theta$  en la figura 4.7. A

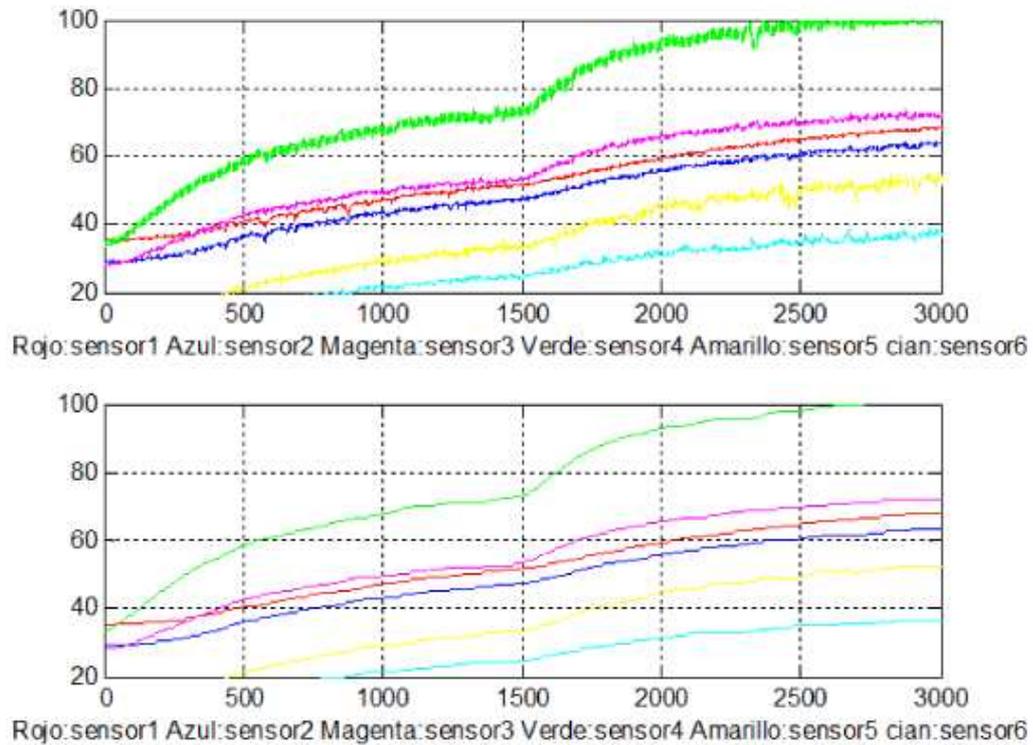


Figura 4.5: Respuesta sin filtrar y respuesta filtrada

continuación se explica como se han obtenido dichos parámetros mediante *m-files* (no se ha hecho de forma gráfica por la cantidad de ruido que presenta el sistema).

1. Cálculo de la ganancia  $K = (T_{\infty} - T_0) / (w_{\infty} - w_0)$ , siendo  $T$ : la temperatura y  $w$ : potencia.
2. Cálculo de la constante de tiempo **ideal**,  $\tau_{ideal}$  (si nuestro sistema no tuviera retardo): Para ello se ha de obtener el valor del tiempo ( $t$ ) cuando la Temperatura Filtrada digitalmente ( $T$ ) alcanza el 63,2% de su valor final.
  - Primero se obtiene  $T_{63,2\%} = T_0 + 0,632 \cdot (T_{\infty} - T_0)$ .
  - Seguidamente para obtener ( $t$ ) se recorre el vector de la señal filtrada hasta llegar al valor  $T_{63,2\%}$ , entonces almaceno el instante de tiempo  $t_{632} = \tau_{ideal}$ .
3. Cálculo de la constante de tiempo **real**,  $\tau_{real}$ .
  - a) Obtener el valor del tiempo ( $t$ ) cuando la Temperatura filtrada ( $T$ ) alcanza el 28,4% de su valor final.

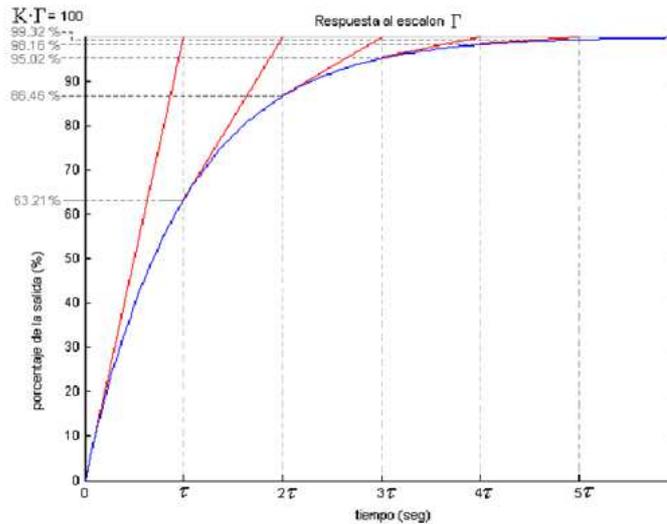


Figura 4.6: Respuesta en  $t$  para un sistema de primer orden a una entrada escalón de magnitud  $\Gamma$

- Primero se obtiene  $T_{28,4\%} = T_0 + 0,284 \cdot (T_\infty - T_0)$ .
  - Seguidamente para obtener  $(t)$  se recorre el vector de la señal filtrada hasta llegar al valor  $T_{28,4\%}$ , entonces almaceno el instante de tiempo  $t_{284} = \tau_{284}$ .
- b) Finalmente se determina la constante de tiempo real,  $\tau_{real} = \tau_{ideal} - \tau_{284}$ , según [2], ver el código en la sección C.2.2.
4. Cálculo del retardo  $\theta$ . Se ha hecho de dos formas diferentes debido a la gran cantidad de ruido que presentan las señales:
- a) Transcurrirá un cierto tiempo hasta que las variables que caracterizan la conducta dinámica ( $T_1 \dots T_6$ ) empieza a modificar su valor en relación al que tenían antes del cambio de la señal de entrada ( $w_i, i = 1, 2$  o  $3$ ). Esta situación se muestra en la 4.7.
  - b) La otra forma esta basada en un método gráfico, según [2], que consiste en obtener el tiempo de establecimiento ( $t_{est}$ ) y posteriormente restamos  $\tau_{real}$  a  $t_{est}$ , ver código en la sección C.2.2.

Los programas y funciones completas se encuentran en el anexo C.

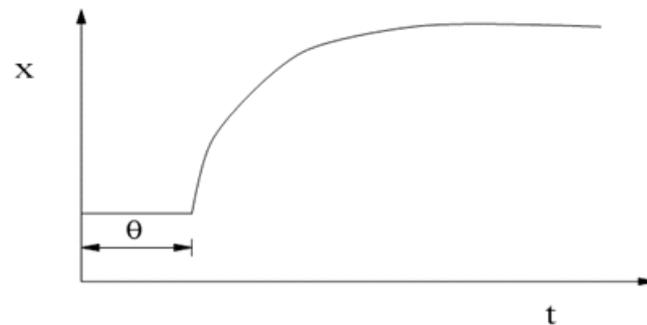


Figura 4.7: Figura que representa retardo

#### 4.7. Validación del Modelo.

La validación se ha llevado a cabo comparando la respuesta real del sistema frente a escalón ( $\Delta U + \hat{5}V$ ) con la respuesta del modelo frente a  $\Delta U$  y analizando el valor absoluto del error  $|Y_{real} - Y_{iden}|$ . La comparación se puede ver en las gráficas de figura 4.8, donde se superponen las  $Y_{real}$  e  $Y_{ideal}$ , para un experimento en el que se realiza el escalón con el *Radiador 2*.

Se repite varias veces la misma operación para cada radiador y nos quedándonos con los modelos que menor error tengan.

Los comandos principales para llevar a cabo el análisis del error son;

```

%Respuesta Real del Horno desde el punto de funcionamiento
2 Yreal=Datos(tpto_fun:tm:tfinal)-Datos(pto_fun);
%Respuesta del modelo a partir de la aplicación del escalón
4 Yiden_1=lsim(G1,entrada,tini:tm:tfinal);
Yiden=Y\_iden(tpto_fun:tm:tfinal)-Yiden(pto_fun); %

```

#### 4.8. Conclusiones iniciales

Aunque la respuesta del modelo dibuja una curva parecida (en valor medio) a la respuesta real, se puede observar que el sistema presenta mucho ruido y esto complicará la sintonización del bucle de control. Por otra parte la aplicación del filtro no mejora las diferencias entre el proceso y el modelo.

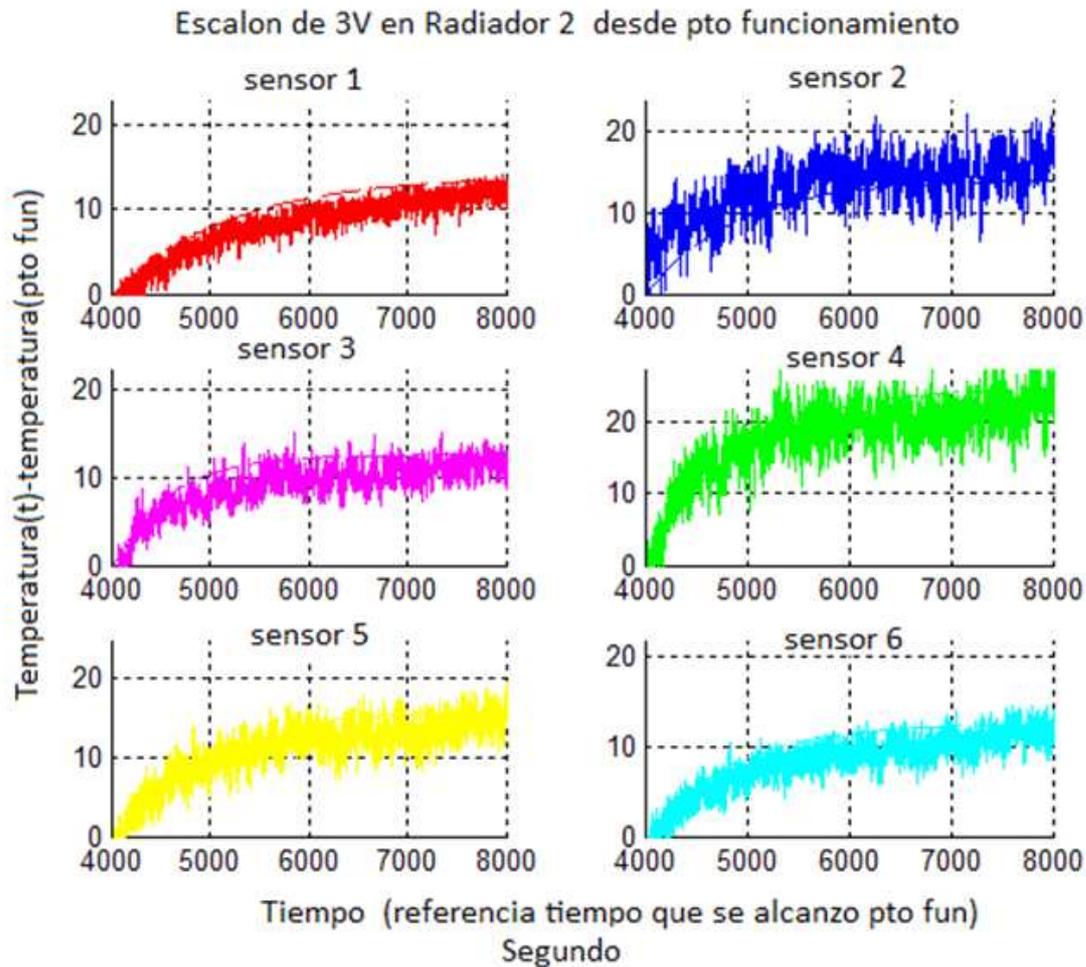


Figura 4.8: Validación del modelo: Respuesta de los sensores frente a escalón de 3V en el Radiador 2

El punto de funcionamiento se ha seleccionado un rango de actuación más o menos lineal, desde 100W a 450 (ver figura 2.5). El problema en este caso es que la zona central (sensores 3 y 4) alcanza 30°C más de temperatura que el resto con lo que la distribución de temperaturas no es homogénea. Además de que es demasiado elevada para utilizar la cinta transportadora y los sensores inalámbricos.

Por todo esto, en esta fase del diseño se ha tomado la decisión de rediseñar la planta.

## Capítulo 5

# Rediseño del Horno

### 5.1. Problemas que presenta el montaje

- Las señales de los termopares tienen mucho ruido.
  - Degradación de la precisión con forma aumenta la Temperatura en la unión fría.
  - Ruido electromagnético producido por las fuentes de tensión, ruido de red, ruido de alta frecuencia producido por los PWM, ruido de alta frecuencia inherente de la circuitería. . .
  - Variaciones en la alimentación de los termopares (Pequeño Rizado).
- La tensión eficaz que llega a los calentadores presenta ruido.
- El sistema de refrigeración se sobre-calienta y deja de funcionar.
  - El diseño del circuito no es el adecuado.
  - El sistema no es operativo.
  - Cualquier cambio o comprobación se hace costosa.
  - Problemas de alcance para encendido y apagado de equipo.
- La Seguridad no es buena.
  - No hay circuito de protección.
  - La tensión de red al alcance.

## 5.2. Ruido en los termopares

Aunque tiene la respuesta más rápida que otras tecnologías, la medida del voltaje en los extremos de la sonda tipo  $K$  se hace poco precisa a causa del mismo sistema de medida, (para más información ver anexo A). Por ejemplo se podría generar dicha diferencia de tensión en diferentes condiciones:

1. Temperatura unión caliente  $100^{\circ}\text{C}$ , Temperatura unión fría  $20^{\circ}\text{C}$ . Diferencia de  $80^{\circ}\text{C}$ .
2. Temperatura unión caliente  $90^{\circ}\text{C}$ , Temperatura unión fría  $10^{\circ}\text{C}$ . Diferencia de  $80^{\circ}\text{C}$ .

Por lo tanto, se obtendrían  $80^{\circ}\text{C}$  de diferencia con 2 temperaturas. Por esta razón es muy importante la capacidad de determinar la temperatura de la unión fría con la mayor precisión posible, puesta que esta capacidad genera la precisión del sistema de medida. De ahí que se va a alejar los transmisores de la zona caliente del horno.

Otros errores que se han tenido en cuenta en la medida son:

- Errores debido a la resistencia del instrumento de medición.
- Los termopares son propensos a tener deriva a lo largo de un cierto periodo de tiempo, producidos por la propia naturaleza de construcción.
- La salida del termopar es de pequeña señal a la cual se le sumaría el ruido electromagnético. Este ruido es debido al termopar trabajando como una antena. De ahí que se requiera que los cables de conexión estén retorcidos y dentro de una funda metálica conectada a tierra y que el amplificador tenga una buena relación señal/ruido.

**Modificación:** *Los errores debidos a la resistencia del instrumento de medida no son compensables y los de deriva para el tiempo de funcionamiento de nuestra planta se pueden considerar despreciables. En cuanto a lo que se ha hecho para eliminar la radiación electromagnética es reducir la longitud de los cables de los termopares y se han enfundado con una funda metálica conectada a tierra.*

Estos errores secundarios suelen ser más importantes que los del propio sensor. Teniendo en cuenta las limitaciones de calibrado de nuestra aplicación asumiremos dicha limitación en la exactitud. Así pues, será difícil obtener errores del sistema inferiores  $\pm 1^{\circ}\text{C}$ .

### 5.3. Atenuación del Fenómeno de Aliasing

El fenómeno de aliasing es propio de la conversión A/D, en el cual la frecuencia de la señal reconstruida es menor que el de la señal original, lo cual ocurre cuando la frecuencia de muestreo es demasiado baja; En las figuras 5.1 se representa dicho efecto en el dominio del tiempo y en el de la frecuencia.

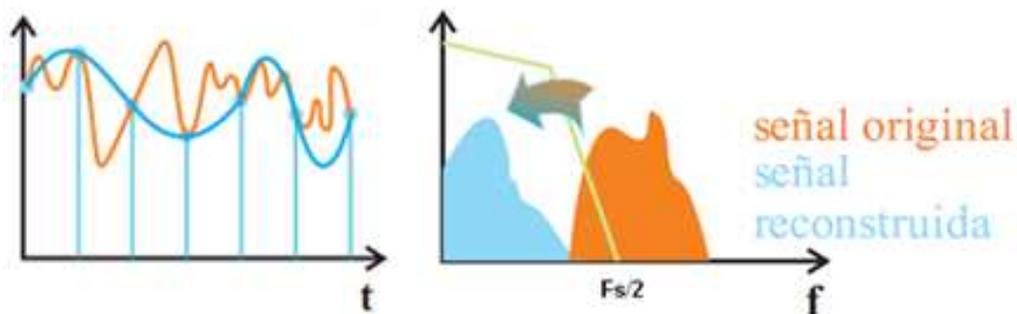


Figura 5.1: Parte de la energía de la señal de alta frecuencia se puede colar como energía de baja frecuencia.

La energía asociada a las componentes de alta frecuencia que no alcanzamos no desaparece, sino que se convierten en parte de la señal erróneamente reconstruida de baja frecuencia (alias).

#### 5.3.1. Detección de Alias

Para detectar este fenómeno muestrearemos a diferentes frecuencias durante 1000 segundos y compararemos los espectros obtenidos. La frecuencias de muestreo utilizadas serán  $1Hz$  (la utilizada en el sistema de control),  $10Hz$  (la utilizada en la identificación) y  $100Hz$ . Como ejemplo se muestra el espectro de frecuencias del sensor 4, para las tres frecuencias, en la figura 5.2.

Si comparamos la figura 5.2.a) con 5.2.b) se puede ver como muestreando a  $1Hz$  o a  $10Hz$  la parte de la energía de la señal de alta frecuencia se cuela como energía de baja frecuencia.

#### 5.3.2. Eliminación de Alias de Alta frecuencia

Una forma de eliminar alias de alta frecuencia es mediante un condensador en paralelo con la tensión medida. En la figura 5.3 se muestra una simulación y en la figura 5.4 los resultados, la fuente de ruido genera un ruido aleatorio de alta frecuencia.

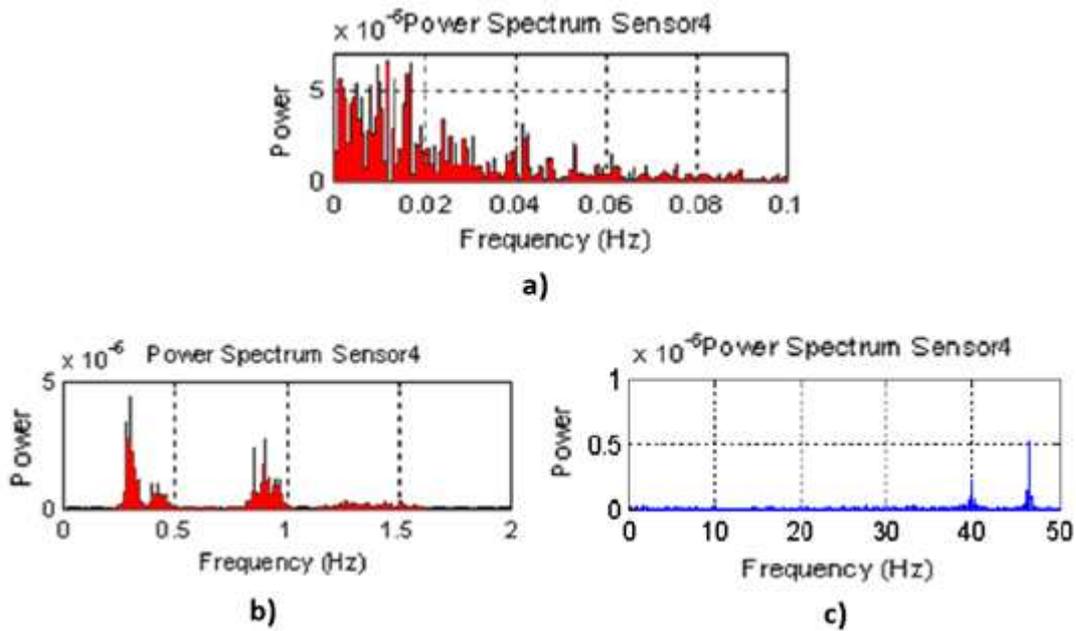


Figura 5.2: a) Frecuencia Muestro =  $1\text{Hz}$ . B) Frecuencia de muestreo= $10\text{Hz}$  y c) Frecuencia de Muestreo= $100\text{Hz}$ .

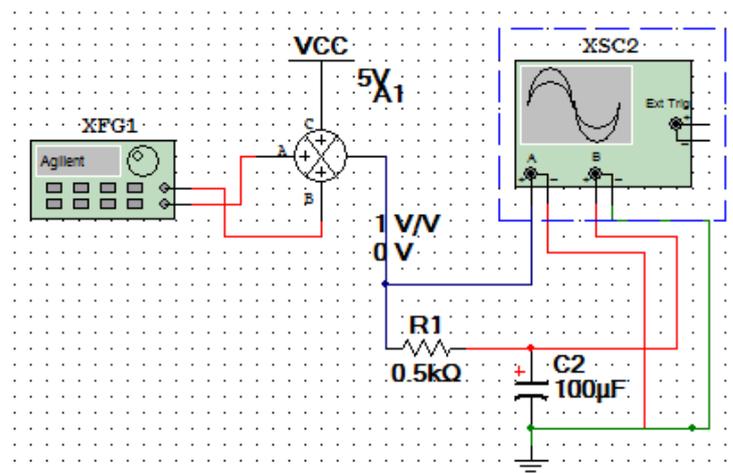


Figura 5.3: Circuito de MultiSim del circuito equivalente. ( $C = 100\mu F$ ).

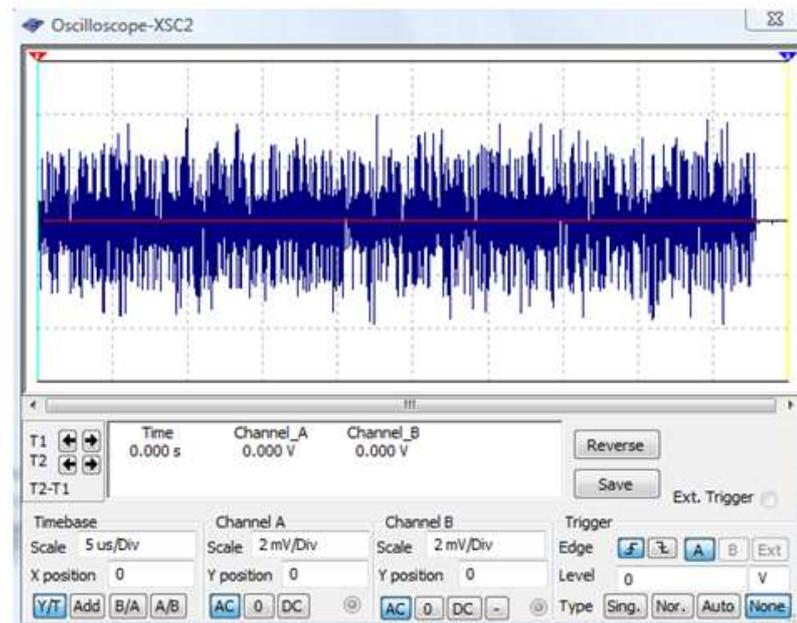


Figura 5.4: Resultados de la simulación. En rojo con el condensador y en azul sin condensador.

### Implementación

El paso siguiente ha sido la implementación (figura 5.5) y comprobación mediante osciloscopio (figura 5.6).

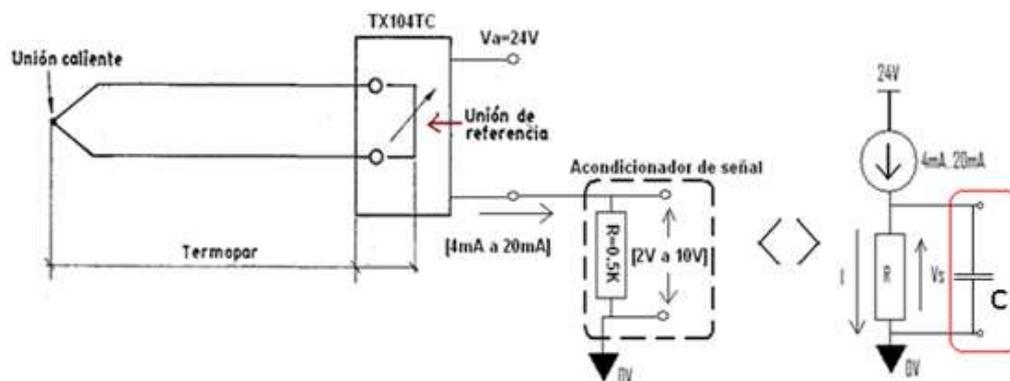


Figura 5.5: Implementación del condensador en paralelo con la salida del transmisor.

Se hace una medición estable de la temperatura ambiente con dos termopares. Y se compara, mediante osciloscopio en acoplamiento AC, un sensor con condensador ( $C = 100\mu F$ ) en el canal 1 y sin condensador canal 2 (figura 5.6). En la figura 5.6 se puede ver que el canal 2 tiene mucho más ruido.

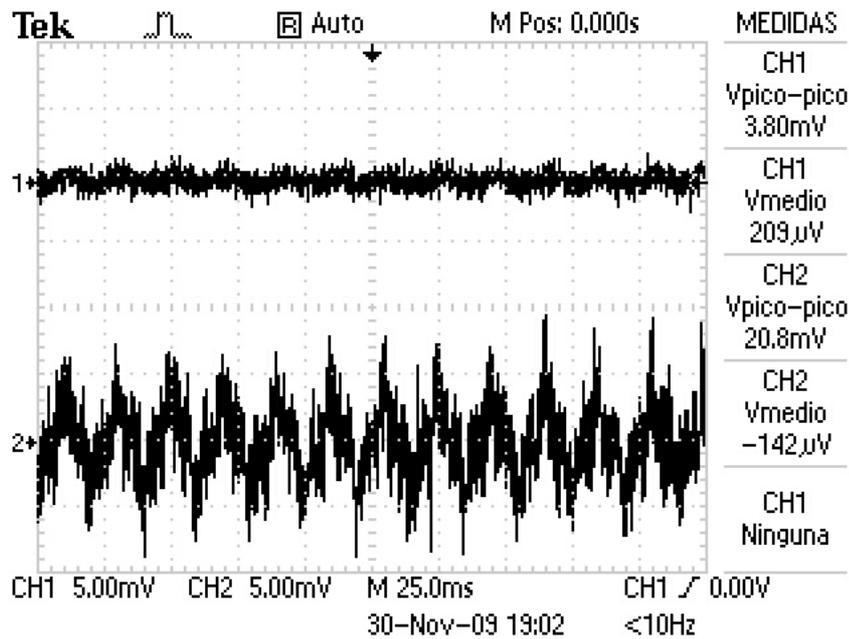


Figura 5.6: Comparación de la señal de salida con condensador en la salida (canal 1) y sin condensador en la salida (canal 2), ambas en acoplamiento AC. ( $C = 100\mu F$ ).

Esto también es evidente en los espectros de frecuencia de ambos canales (figura 5.7). Donde distinguimos componentes de 50Hz y armónicos (150Hz y 250Hz) que posiblemente sean de la red. Aunque con el condensador se han atenuado los alias de alta frecuencia el ancho de banda aun

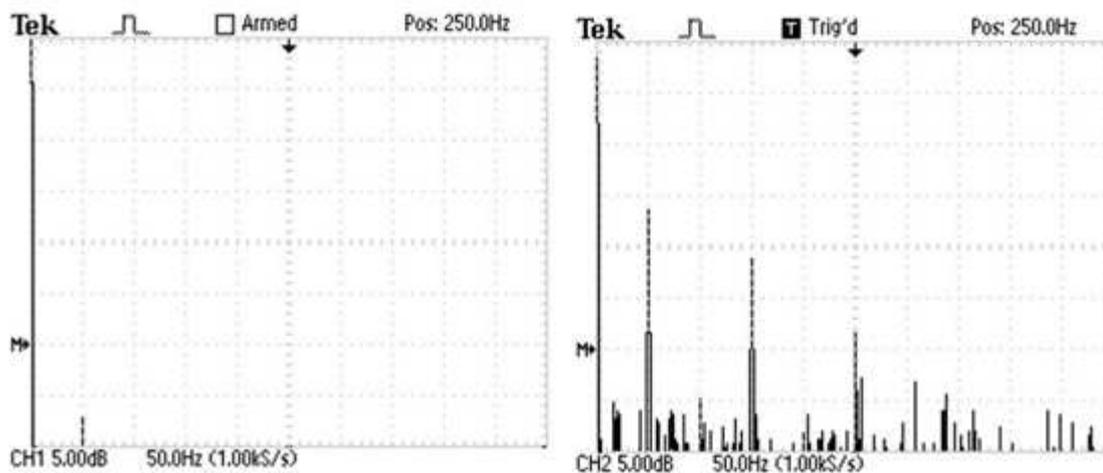


Figura 5.7: Comparación del espectro en canal 1 y canal 2 en acoplamiento AC. Frecuencia de muestreo de 1KHz.

tiene componentes que no son de nuestro interés, por ello vamos a instalar un Filtros Analógicos Paso-Bajo o Filtro Antialiasing en la tarjeta de adquisición de datos.

### 5.3.3. Diseño del Filtro Antialiasing

Para el diseño del filtro utilizaremos el criterio de Shannon-Nyquist, el cual establece que la frecuencia mínima de muestreo necesaria para evitar el efecto de aliasing, debe ser 2 veces mayor que el ancho de banda:

$$\blacksquare F_s > 2BW. \text{ Con } BW = F_{max} - F_{min}$$

Para señales con  $F_{min} = 0$ , la expresión queda:

$$F_s > 2 \cdot F_{max} \quad (5.3.1)$$

Por tanto la primera consideración para diseñar un filtro antialiasing es el periodo de muestreo de la A/D y las frecuencias presentes en la señal ( $BW$ ). En nuestro caso el ancho de banda ( $BW$ ) tras situar el condensador de  $100\mu F$  es de  $50Hz$ , con lo que deberíamos muestrear como mínimo a  $100Hz$  para no tener aliasing.

Para poder muestrear a frecuencias más bajas debemos establecer un ancho de banda deseado ( $BW_{deseado}$ ) que será función de la Frecuencia de muestreo deseada ( $F_{s_{deseada}} > 2 * BW_{deseado}$ ). Establecido  $BW_{deseado}$ , nuestro filtro antialiasing tendrá como frecuencia de corte ( $F_c$ ) la Frecuencia máxima del ancho de banda deseado.

$$F_c < F_s/2 \text{ o también } F_{s_{deseada}} > 2 * BW_{deseado}.$$

Utilizaremos un filtro analógico pasobajo  $RC$  cuya frecuencia de corte es  $f_c = 1/2\pi \cdot R \cdot C$ .

Por ejemplo si quisiéramos muestrear a  $1Hz$  sin aliasing la frecuencia de corte debe ser como máximo de  $0,5Hz$ . Siendo el producto  $R \cdot C = 0,32s/rad$ .

#### Selección de valores del filtro e instalación en la DAQ PCI9111

Teniendo en cuenta la estructura interna de la tarjeta, ver figura 5.8.

$R_{Bn}$  está abierta mientras que  $R_{An}$  debe ser bastante mayor que la resistencia en paralelo con el transmisor TX104TC,  $500\Omega$ , (ver figura 5.5), para asegurarnos que no se deriva corriente al convertidor. Pero tampoco demasiado grande porque podría atenuar mucho la señal. Seleccionamos  $R = 3K3\Omega$  y calculamos  $C = 0,32/3300 = 100\mu F$ .

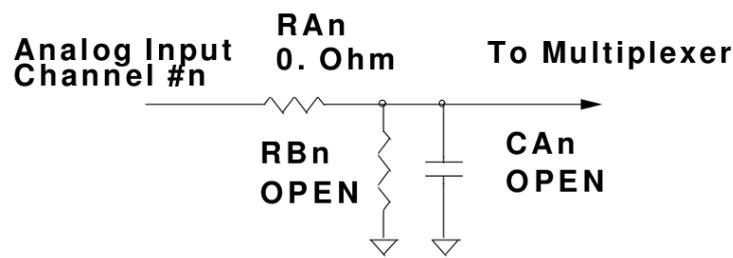


Figura 5.8: Filtro en la entrada analógicas #n

Para la instalación de las resistencias en las  $n$  entradas se ha sustituido  $RAn$  por las resistencias calculadas (ver figura 5.8). Mientras que para el condensador se ha soldado en el espacio dedicado para cada entrada los condensadores calculados. La estructura completa del sistema de adquisición quedará tal cual se muestra en la figura 5.9.

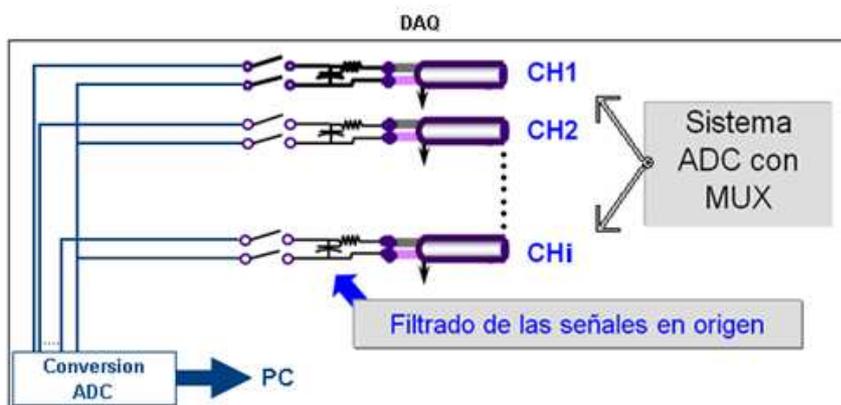


Figura 5.9: Estructura interna del sistema de adquisición para los  $n$ , teniendo en cuenta los filtros RC.

### Simulación e Implementación

Para asegurarnos que los diseños funcionan adecuadamente se ha utilizado el software Multisim de Texas Instruments que permite la simulación de circuitos eléctricos y electrónicos. El circuito equivalente se puede ver en la figura 5.10. Donde la fuente de ruido simulada genera un ruido aleatorio tanto en amplitud como en frecuencia.

Ver resultados en la figura 5.11.

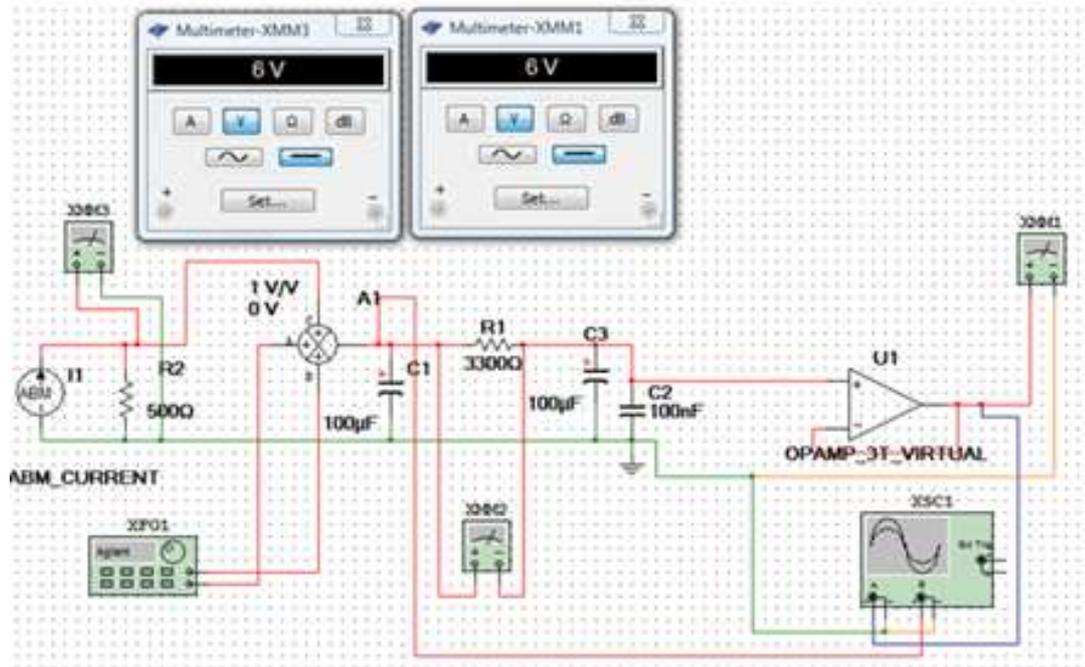


Figura 5.10: Circuito simulado del sensor+transmisor simulado.

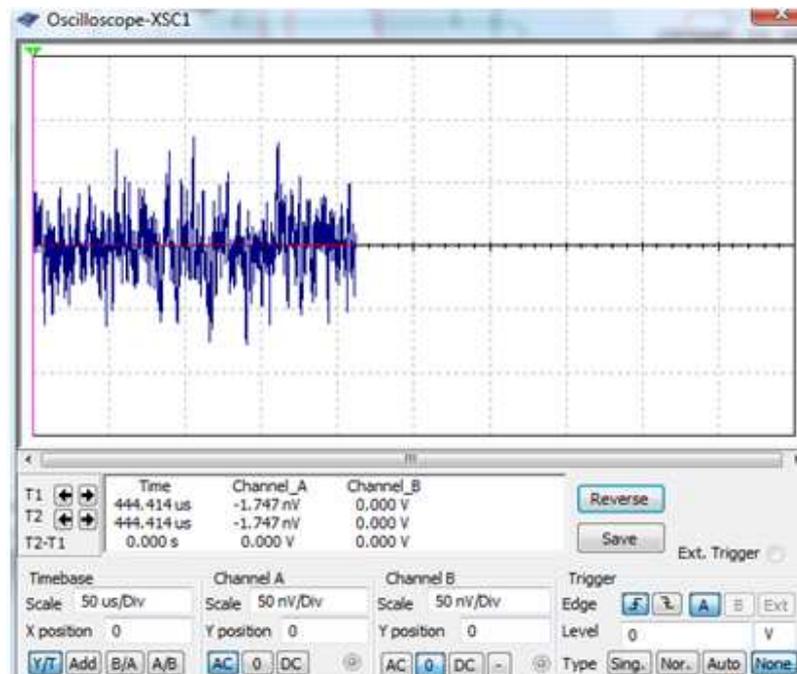


Figura 5.11: Resultados obtenidos en la simulación: rojo señal a la salida del filtro, azul la señal antes del filtro

Para probar el funcionamiento compararemos la señal de entrada al filtro y la de salida con el osciloscopio, ver figura 5.12, frente a pequeños cambios de temperatura en la sonda. Se observa que la señal anterior al filtro es más sensible a cualquier cambio, incluso genera un transitorio no deseado.

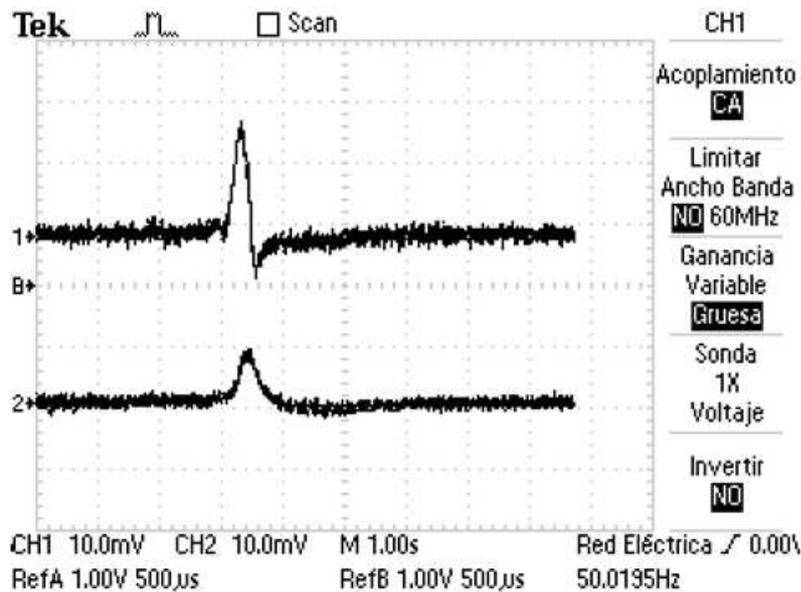


Figura 5.12: Señales reales del circuito sensor en acoplamiento AC. Canal 1 sin filtro  $RC$ . Canal 2 con filtro  $RC$

## 5.4. Mejora de la adquisición

### 5.4.1. Mejora de precisión en la unión fría.

Según las hojas de características del transmisor la precisión del transductor empeora a razón de  $0,05^{\circ}\text{C}/^{\circ}\text{C}$  (Precisión a  $20^{\circ}\text{C}$  de  $0,2^{\circ}\text{C}$ ), es decir un aumento de temperatura en la unión fría del transmisor repercute en la precisión. Para solucionar este problema y también la accesibilidad se ha sacado el conjunto de transmisores que había debajo del horno, y se ha situado en un lateral mediante un soporte de madera, ver Figura 5.13. De esta forma evitaremos que aumente la temperatura en la unión fría de los termopares (CJ), puesto que debajo del horno la temperatura llegaba a alcanzar de  $35^{\circ}\text{C}$  a  $40^{\circ}\text{C}$ .

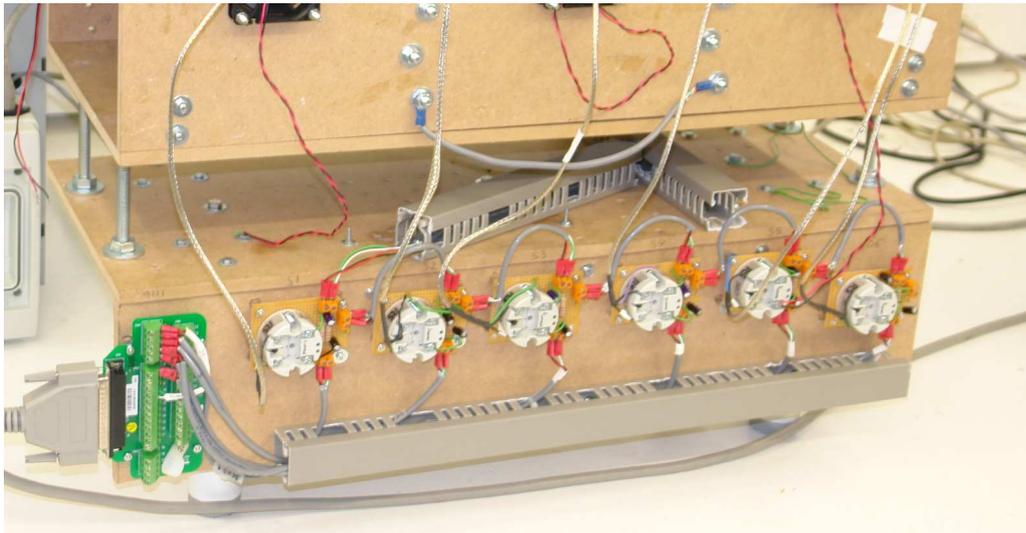


Figura 5.13: Reubicación de los sensores de temperatura.

#### 5.4.2. Mejora de la alimentación de los transmisores.

La fuente de alimentación de  $24V$  que alimenta los transmisores TX104T, que acondiciona la señal de los termopares, tiene un rizado mayor a  $3mV$ , que es el indicado por las hojas de características.

#### Simulación e Implementación

Para eliminar este rizado de alta frecuencia y evitar interferencias se ha utilizado un regulador de tensión variable LM117HVH. Simulado mediante el programa Multisim, ver implementación en figura 5.14 y diseñado según hojas de características del propio regulador.

La fuente de alimentación de la simulación genera un onda senoidal de amplitud igual al rizado indicado en las hojas de características, comparando así la señal de la fuente con la obtenida en la salida del regulador (ver resultado de la simulación en la figura 5.15).

El siguiente paso es la implementación en circuito integrado y comprobación con el osciloscopio, ver resultados en la figura 5.16.

Se puede observar en la figura 5.16 que el rizado desaparece eliminando gran parte del ruido que afectaba a los sensores.

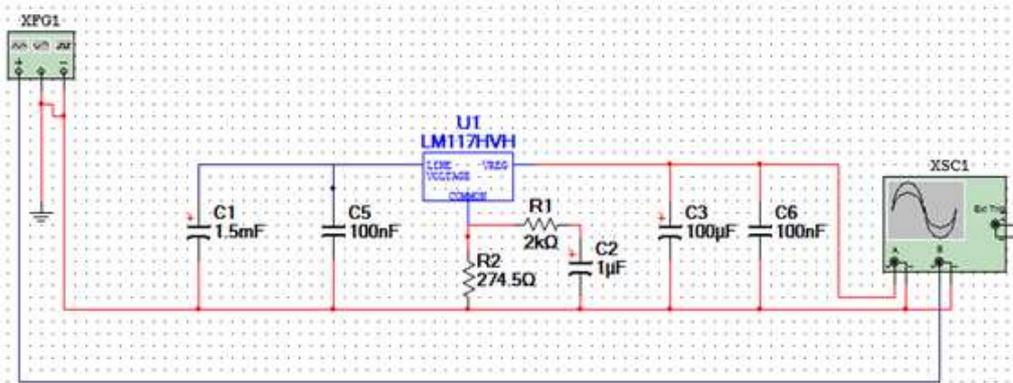


Figura 5.14: Implementación del regulador de tensión.

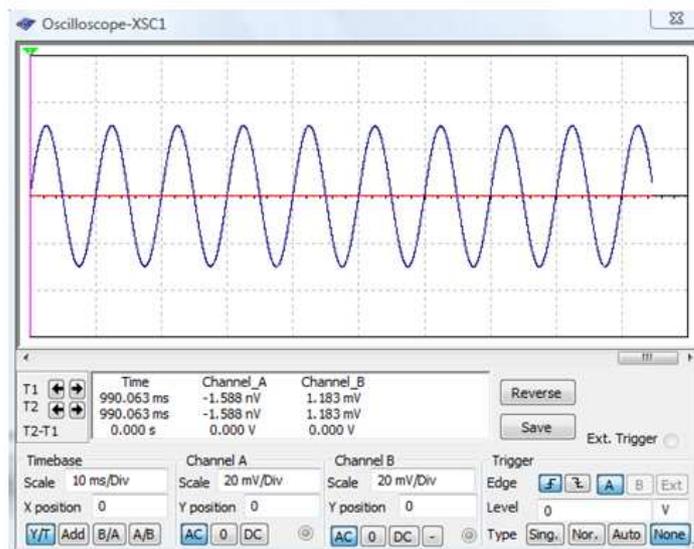


Figura 5.15: Comparación simulada de señal de salida del regulador (rojo) con la señal de la fuente (azul).

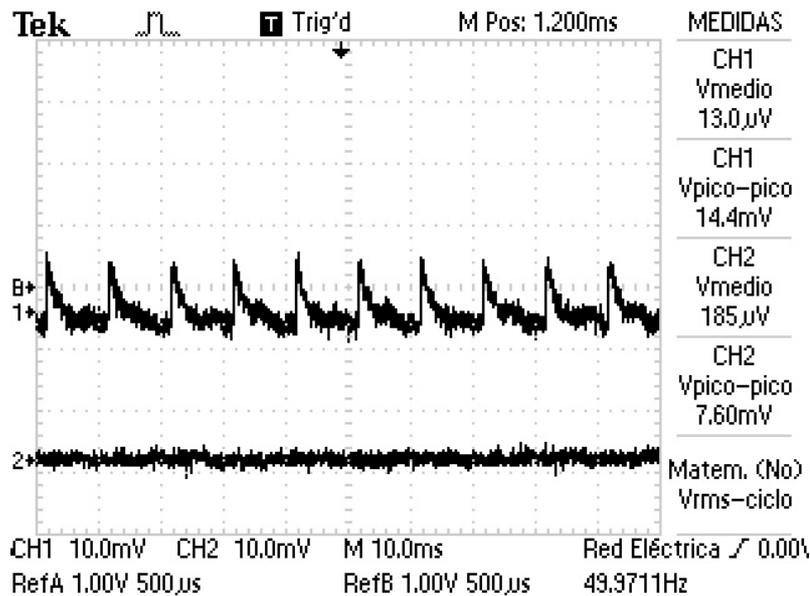


Figura 5.16: Comparación de las señales reales en acoplamiento AC. Señal de salida del regulador canal 2 (inferior) y señal de salida de la fuente canal 1 (superior).

## 5.5. Mejora de la señal de los actuadores

Si visualizamos la señal de la tarjeta sin estar conectada al PWM, esta no presenta ninguna oscilación en cambio al conectarlo al PWM con la señal de red conectada aparece un rizado con picos de tensión, ver gráfico de la derecha de figura 5.17. Los picos de alta frecuencia se pueden eliminar con un condensador de  $C = 100\mu F$  en paralelo a la entrada (C1) (ver figura 5.20) el resultado se presenta en el gráfico de la izquierda de la figura 5.17.

Por otro lado el rizado lo filtraremos con un paso-bajo (RC), aprovechando la resistencia de potencia instalada. Como en el diseño de la sección 5.3 lo primero que haremos será obtener el ancho de banda del sistema tras haber instalado el condensador, observando el periodo del rizado en se deduce que  $F_{max} = 100Hz$ . Se optará por un condensador de gran tamaño que atenúe mucho dicha frecuencia. Como solución se propone un condensador de  $1500\mu F$  (C2) que con R tendrá una frecuencia de corte de  $2Hz$  (ver figura 5.20).

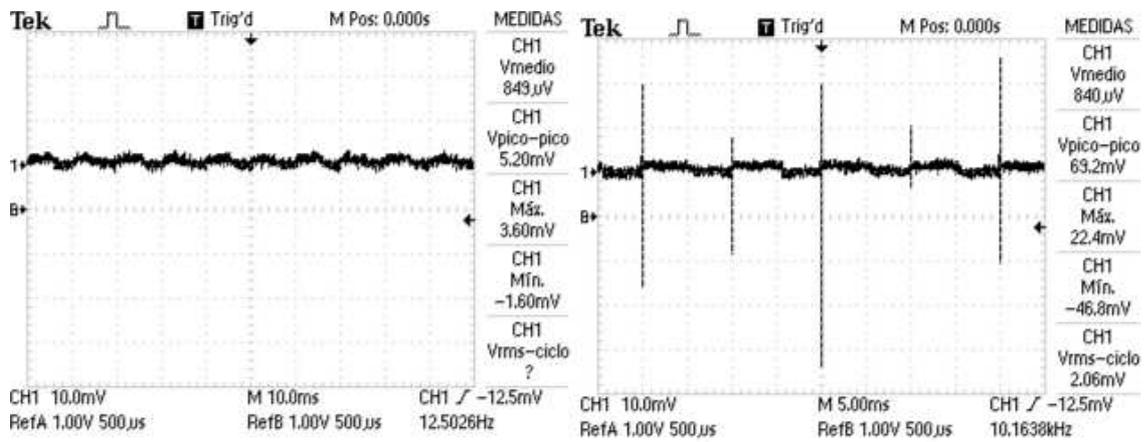


Figura 5.17: La gráfica de la izquierda representa el resultado con Condensador en paralelo en la entrada mientras que la de la derecha representa la señal en el diseño inicial sin Condensador.

### Simulación e Implementación

Para simular las características del PWM se ha utilizado una resistencia equivalente a la impedancia de entrada del PWM, ver R1 en figura 5.18. La fuente de alimentación genera una señal cuadrada de amplitud similar a la que se produce en al entrada del PWM desde la tarjeta (figura 5.18).

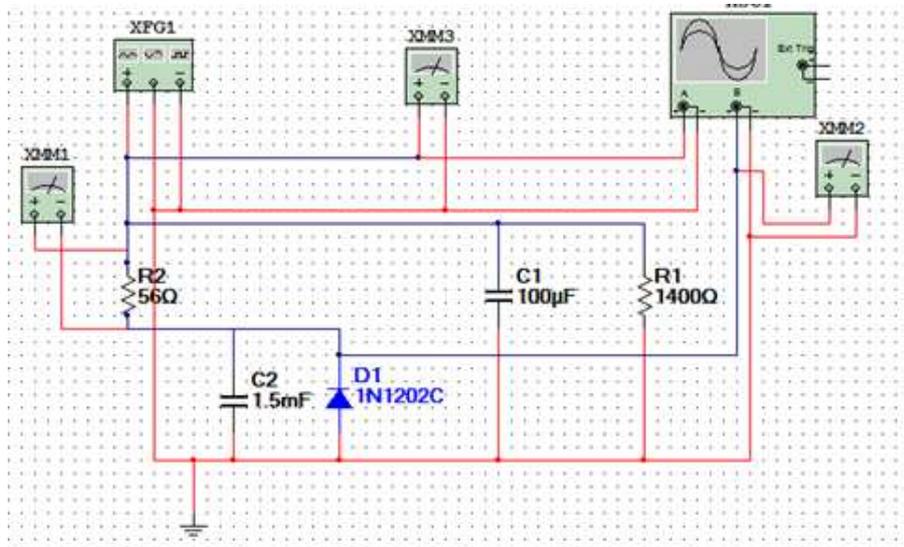


Figura 5.18: Circuito equivalente a la entrada de la etapa de potencia que controla los radiadores

En la figura 5.19 se muestran los resultados de la simulación. Donde se observa que existe un transitorio que se estabiliza a los 0.4 segundos, lo cual es menor al tiempo de muestreo utilizado (1 segundo).

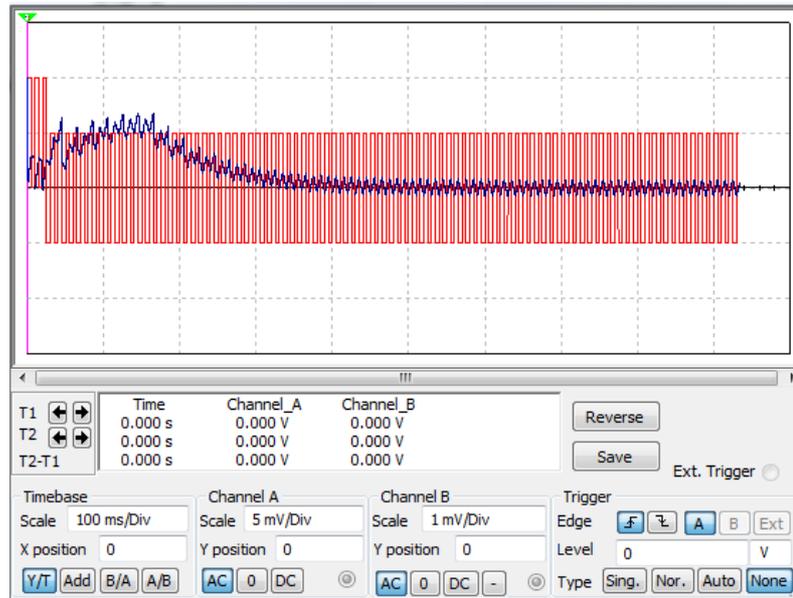


Figura 5.19: Resultado de la implementación. En azul la señal filtrada y en rojo la señal de la fuente de alimentación

La implementación se ha llevado a cabo sobre el circuito integrado ya implementado, simplemente se ha soldado los condensadores C1 en paralelo con la entrada y C2 en paralelo con el diodo (D), ver figura 5.20.

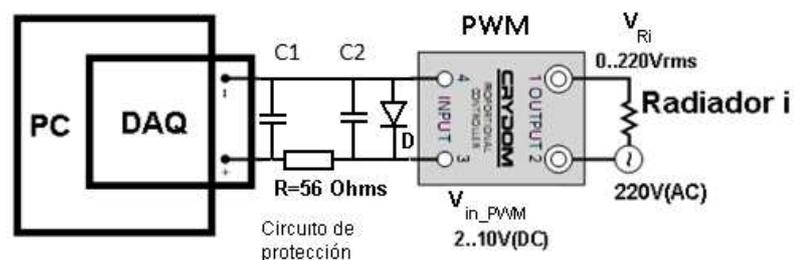


Figura 5.20: Implementación del condensador en paralelo y filtro entre la tarjeta PCI 6216 y el PWM

En la figura 5.21 se compara la señal de entrada del PWM pasando por el filtro (canal 2) y la misma señal sin filtrar (canal 1). Se puede ver que se ha reducido los picos y el rizado, aunque la mejora no es significativa.

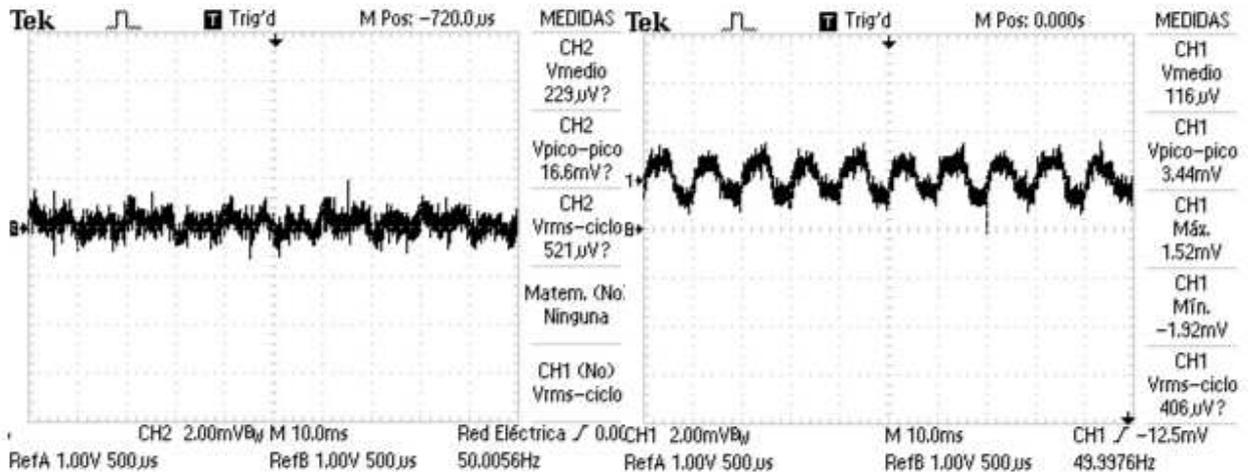


Figura 5.21: La gráfica de la izquierda representa el resultado con el filtro mientras que la de la derecha representa la señal en el diseño inicial sin filtro.

## 5.6. Atenuación del ruido Electromagnético. Apantallamiento

Aprovechando las modificaciones se ha cambiado el cableado por otro de señal apantallado, que evitará interferencias provocadas por campos electromagnéticos externos.

Todas las pantallas están conectadas a las masas flotantes:

- La de los sensores a la masa de la fuente que alimenta a los transmisores. Que también es la del PC a través de la DAQ PC9111.
- La de los actuadores a la masa de la DAQ6216 que será la misma que la del PC.

## 5.7. Modificación Estructural del Prototipo

Además de las modificaciones orientadas a la eliminación de interferencias se han llevado a cabo otros cambios para conseguir un sistema más *Operativo y Seguro*. Ver figura 5.22.

- El cable de alimentación se ha conectado a un interruptor automático de seguridad. Se mejora la seguridad y la facilidad para conectar y desconectar el sistema.
- Se han trasladado todas las señales de potencia, fuentes de alimentación y actuadores a una carcasa cerrada y apantallada de PC, con lo que:

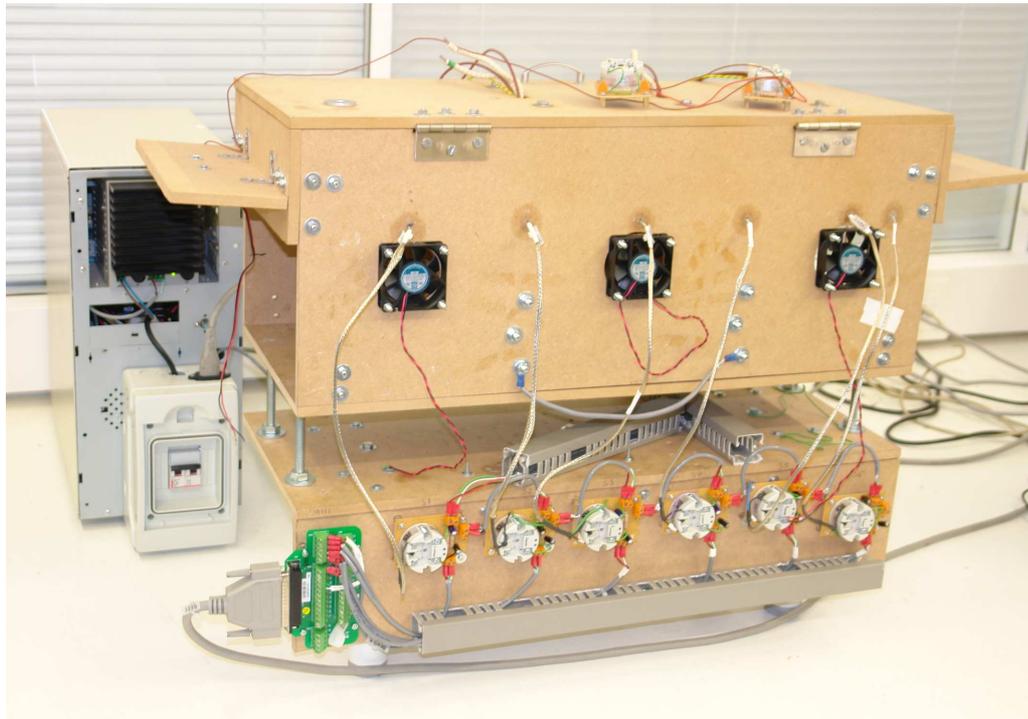


Figura 5.22: Distribución del Horno después de las modificaciones

- No se podrá acceder directamente a la tensión de red.
- El técnico solo tendrá que quitar la tapa de la carcasa para acceder a los actuadores.
- Se alejan lo campos magnético que puedan acoplarse al termopares

## 5.8. Medición en calentamiento

Para ver el resultado se ha calentado el horno y se ha medido directamente con la tarjeta. A continuación se representa los resultados de los sensores 3 y 4 (ver figura 5.23), que están situados en el centro. En las figura se puede ver que las fluctuaciones en la señal a desaparecido.

## 5.9. Reparación del sistema de ventilación

Después de analizar los picos de los transitorios de intensidad, se llego a la conclusión de que cambios sincronizados en más de 5 ventiladores superaban la potencia máxima suministrada por

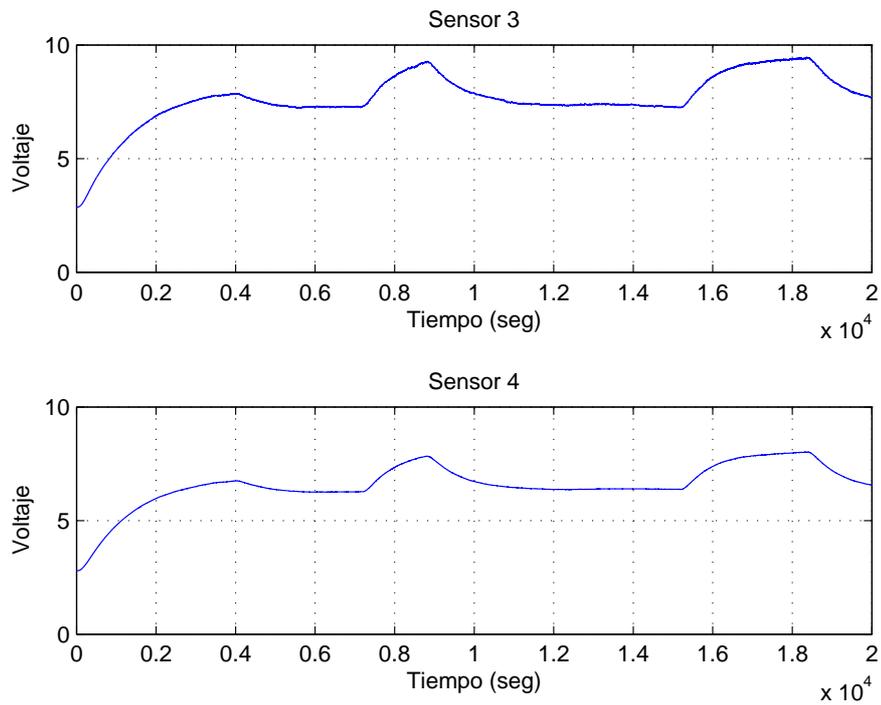


Figura 5.23: Resultados de la adquisición a periodo un segundo después del rediseño

la fuente. Para solucionar este problema de diseño se ha optado por añadir otra fuente, de forma que cuatro ventiladores están conectados a una fuente y los otros cuatro a la otra. Esta opción es la que minimiza las tareas de instalación en el proceso.

## **Parte III**

# **Identificación, Control y Resultados experimentales**



## Capítulo 6

# Identificación después del Rediseño

### 6.1. Introducción

Después de rediseñar la planta se va realizar de nuevo la identificación, en este caso y tras estudiar el toolbox de identificación de `Matlab`, se recompondrán los scripts de identificación. En la figura 6.1 se representan los pasos de identificación que se van a seguir en esta fase. Se obtendrá un primer modelo,  $G1(s)$  (ver figura 6.1), que partirá de la temperatura ambiente y tendrá los radiadores apagados. El modelo  $G1(s)$  nos servirá para diseñar un sistema de control que consiga un perfil de temperatura deseado y que además nos sirva para decidir el punto de funcionamiento ( $T_{deseada}, W_c$ ) (ver figura 6.1). En este modelo,  $G1(s)$ , los ventiladores se fijarán en un valor, siendo una perturbación que no identificaremos, sí lo haremos al obtener  $G2(s)$ .

### 6.2. Modelo de Puesta en Marcha $G1(s)$

#### 6.2.1. Condiciones del modelo

Como se ha dicho en 6.1,  $G1(s)$  parte del reposo, excepto los ventiladores que se pondrán desde el inicio en el punto de funcionamiento ( $V_{0vent}$ );  $V_{0vent}$  será tal que permita a los radiadores llevar al sistema al punto de funcionamiento deseado ( $T_{deseada}, W_c$  y  $V_{0vent}$ ), que permita un descenso de temperatura similar a la velocidad de aumento de temperatura, y que además este situado en un valor de tensión ( $V_{vent}$ ) que no limite su rango de actuación,  $0 \dots 10V$ .

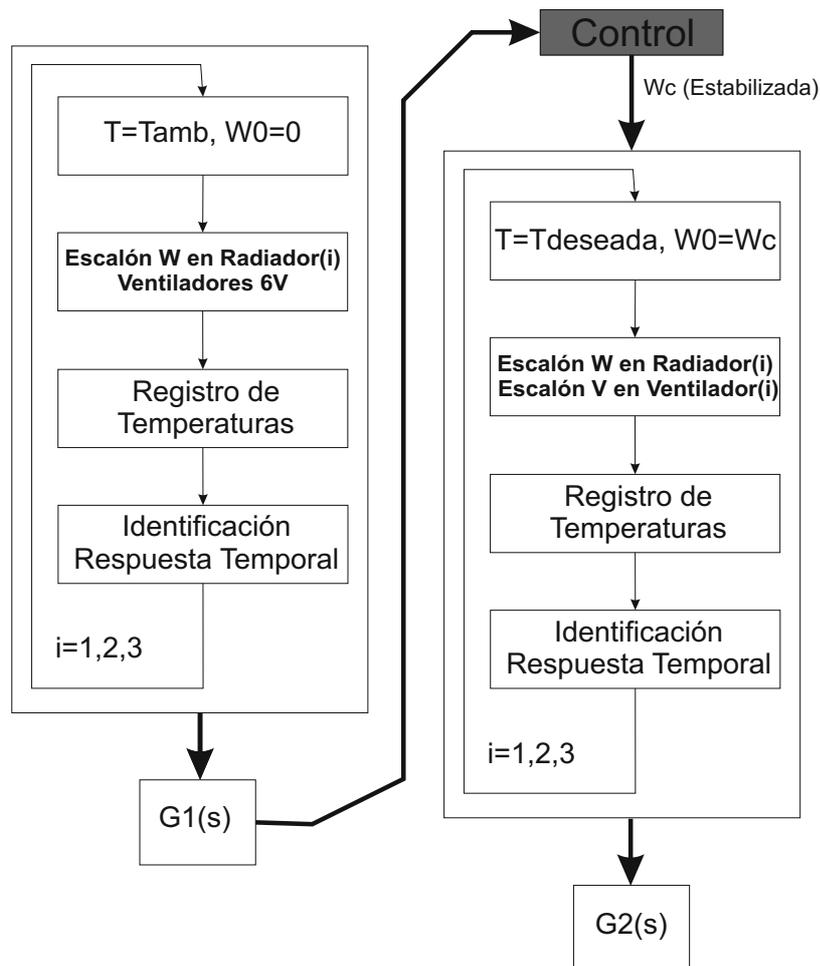


Figura 6.1: Etapas de Identificación.

Así, después de llevar a cabo varios experimentos de identificación y control se llegó a la conclusión que  $V_{0_{vent}} = 6V$  en todos los ventiladores era un buen punto para cumplir las tres condiciones.

### 6.2.2. Obtención del modelo

La toma de datos será idéntica a la realizada en el capítulo 4. Lo que variará en este caso es la forma de realizar el ajuste ya que la función `pem` de Matlab nos permite una mejor identificación del modelo.

`pem` es una función de toolbox de identificación de Matlab que permite la identificación de modelos continuos a partir de datos de entrada, datos de salida y una estructura de mode-

lo, así si tenemos los datos de la entrada  $i$  y los datos de la salida  $j$  podremos obtener el modelo  $G_{ji}$ . Esto en Matlab con la función pem sería:  $G_{ji} = \text{pem}(\text{Datos\_}j_i, 'P2')$ , donde  $\text{Datos\_}j_i = \text{iddata}(\text{Incremento\_}T(j), \text{Escalon}(i))$  son los datos:  $[\text{Salida}, \text{Entrada}]$  y 'P2' es la estructura seleccionada, expresión 6.2.1. El código completo se puede ver en la sección C.3.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (6.2.1)$$

En la tabla 6.1 se muestra los modelos de los 6 sensores frente al Radiador 1. Mientras que en la figura 6.2 se comparan los modelos con la respuesta real del sistema.

$$G_{11} = \frac{0,0263}{(1,6678s+1)(621,118s+1)} \frac{W}{^\circ C} \quad G_{21} = \frac{0,0449}{(136,8179s+1)(767,3997s+1)} \frac{W}{^\circ C}$$

$$G_{31} = \frac{0,1007}{(2,713s+1)(829,1874s+1)} \frac{W}{^\circ C} \quad G_{41} = \frac{0,1505}{(12,4035s+1)(918,2494s+1)} \frac{W}{^\circ C}$$

$$G_{51} = \frac{0,2040}{(14,3802s+1)(865,8009s+1)} \frac{W}{^\circ C} \quad G_{61} = \frac{0,1416}{(4,3459s+1)(862,0690s+1)} \frac{W}{^\circ C}$$

Tabla 6.1: Modelos de sensores  $j = 1 \dots 6$  frente a Radiador  $i = 1$ .

Observando el sistema de 3 entradas y 6 salidas se deduce que con tres radiadores podemos controlar 3 de los sensores, lo cual nos servirá para llevar al sistema a un punto de funcionamiento, al menos 3 sensores seguros y los otros 3 se quedarán estabilizados en otras temperaturas.

Los sensores seleccionados son 2, 4 y 6. A continuación, mediante el método de ganancia relativa, se comprueba el emparejamiento que utilizaremos para desacoplar el sistema.

```
1 Gest=dcgain (modelo (2:2:6 , :) ;
Test=pinv (Gest ' )
3 rga0=Gest.* Test
```

$$rga0 = \begin{bmatrix} -0,0005 & -0,5750 & \mathbf{1.5755} \\ -2,1736 & \mathbf{4.5903} & -1,4167 \\ \mathbf{3.1741} & -3,0153 & 0,8412 \end{bmatrix} \quad (6.2.2)$$

El emparejamiento realizado será  $y_2-u_3$ ,  $y_4-u_2$  y  $y_6-u_3$ , por tanto  $G1(s)$  sera de  $3 \times 3$ , los tres radiadores y las tres salidas 2, 4, y 6.

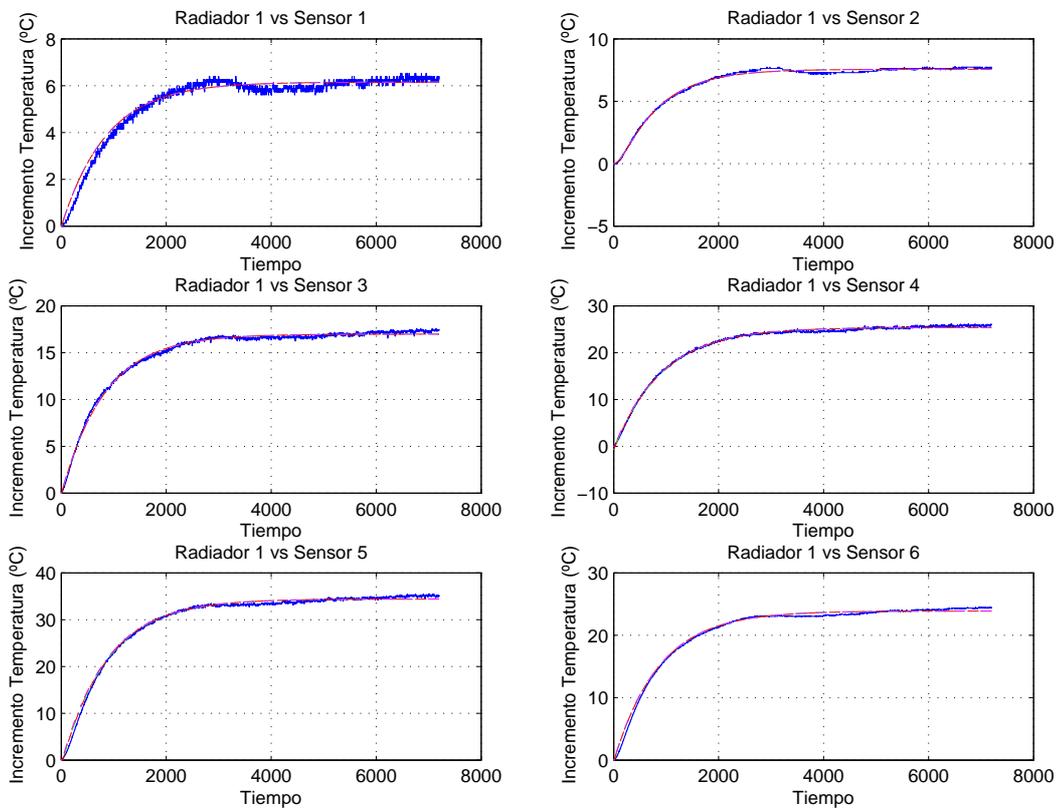


Figura 6.2: Modelo  $G_1(s)$ : Ejemplo de identificación de sensores frente a escalón en Radiador 1.

### 6.3. Modelo Nominal $G_2(s)$

En esta identificación se han utilizado todos los actuadores. Considerando que la pareja Extractor-Ventilador enfrentados tiene la misma tensión de entrada, actuando así como una única entrada y reduciendo el número de entradas de control a 8 (3 Radiadores+ 2 Sopladores+ 3 Ventiladores/Extractores).

La identificación del **Modelo nominal**,  $G_2(s)$ , se ha hecho alrededor de  $T_{deseado}$ ,  $W_c$  y  $V_{0,vent}$  al cual llegamos con el sistema de control diseñado en la sección 7.7. Los valores de dicho punto son:

- Controlados: Sensor 2  $\rightarrow T = 55^\circ\text{C}$ , Sensor 4  $\rightarrow T = 65^\circ\text{C}$  y Sensor 6  $\rightarrow T = 55^\circ\text{C}$ .
- No Controlados: Sensor 1  $\rightarrow T = 43^\circ\text{C}$ , Sensor 3  $\rightarrow T = 68^\circ\text{C}$  y Sensor 5  $\rightarrow T = 70^\circ\text{C}$ .
- Radiador 1  $\rightarrow W = 460,0\text{W}$ , Radiador 2  $\rightarrow W = 296,0\text{W}$ , Radiador 3  $\rightarrow W = 491,0\text{W}$ .

- Ventiladores/Extractores  $\rightarrow V = 6V$  y Sopladores  $\rightarrow V = 3,5V$ .

En la tabla 6.2 se muestra los modelos de los 6 sensores frente al Radiador 1. Mientras que en la figura 6.3 se comparan los modelos con la respuesta real del sistema.

$$G_{11} = \frac{0,0298}{(6,1387s+1)(1513,8s+1)} \frac{W}{^\circ C} \quad G_{21} = \frac{0,0380}{(15,2952s+1)(1318,9s+1)} \frac{W}{^\circ C}$$

$$G_{31} = \frac{0,0568}{(0,001s+1)(950,5703s+1)} \frac{W}{^\circ C} \quad G_{41} = \frac{0,0847}{(15,3704s+1)(800,6405s+1)} \frac{W}{^\circ C}$$

$$G_{51} = \frac{0,1096}{(15,3657s+1)(817,6615s+1)} \frac{W}{^\circ C} \quad G_{61} = \frac{0,0789}{(15,3468s+1)(906,6183s+1)} \frac{W}{^\circ C}$$

Tabla 6.2: Modelos de sensores  $j = 1 \dots 6$  frente a Radiador  $= i = 1$ .

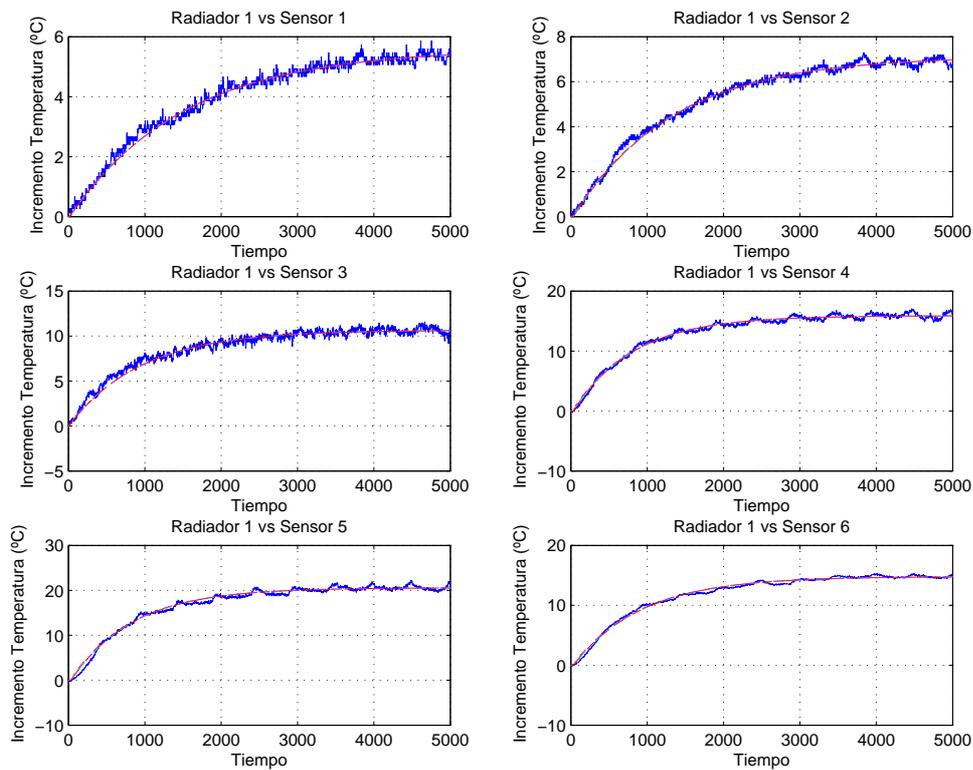


Figura 6.3: Modelo Nominal  $G^2(s)$ : Ejemplo de identificación frente a escalón en Radiador 1.

De la misma forma, en la figura 6.4 se muestra la respuesta frente a cambios en la pareja *Ventilador – Extractor* 1, lo que provoca introducción y extracción de aire por la misma zona.

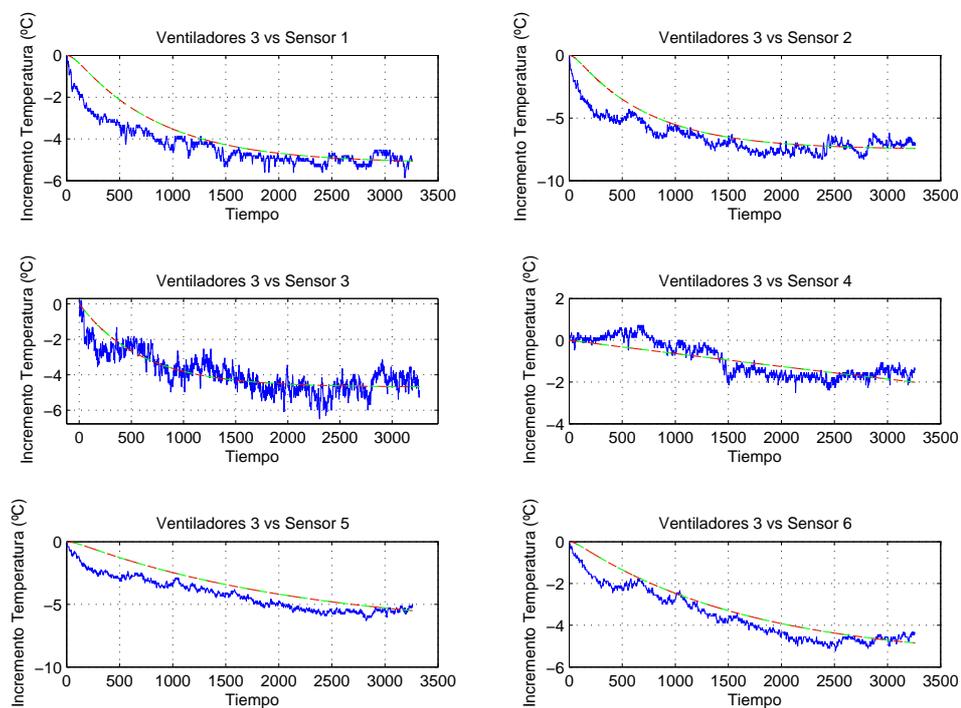


Figura 6.4: Modelo  $G^2(s)$ : Ejemplo de identificación frente a escalón positivo en en la pareja *Ventilador – Extractor 3*.

Se puede ver, figura 6.4, que las variaciones de los ventiladores en el punto de operación tienen poca influencia en la temperatura, esto es debido a la diferencia entre la temperatura interna del horno y la exterior.

También se debe a la dificultad de enfriar la estructura metálica de la cinta transportadora. Este problema se ha intentado solucionar aumentando el punto de funcionamiento para incrementar así la influencia de los ventiladores, pero las limitaciones de temperatura impuestas por los materiales de la plataforma dificultan este incremento.

Para la identificación de este modelo se ha implementado un Scada de identificación el cual se puede ver en la sección 8.2.

# Capítulo 7

## Control

### 7.1. Introducción

Como se explico en la sección 6.1 para situar el sistema en un punto de operación se ha identificado un modelo,  $G1(s)$ , de tres entradas (Radiadores) y tres salidas (Termopares 2, 4 y 6), linealizando alrededor de la temperatura ambiente de la sala que nos servirá para el diseño de un bucle de control el cual realiza la puesta en operación del horno. Por otro lado también intentaremos diseñar un sistema de control parar el modelo  $G2(s)$ , donde se utilizan todos los actuadores y sensores.

### 7.2. Arquitectura de Control

Se ha realizado un control por realimentación del estado con dos grados de libertad, Figura 7.1, [4], [9], [3], [12].

Las acciones de control se calculan en dos partes:

- El control por realimentación evalúa el error y aplica las acciones de control necesarias para llevar el sistema a error nulo.
- Por otro lado el control en bucle abierto calculará la acción basada en las entradas de referencia externas.

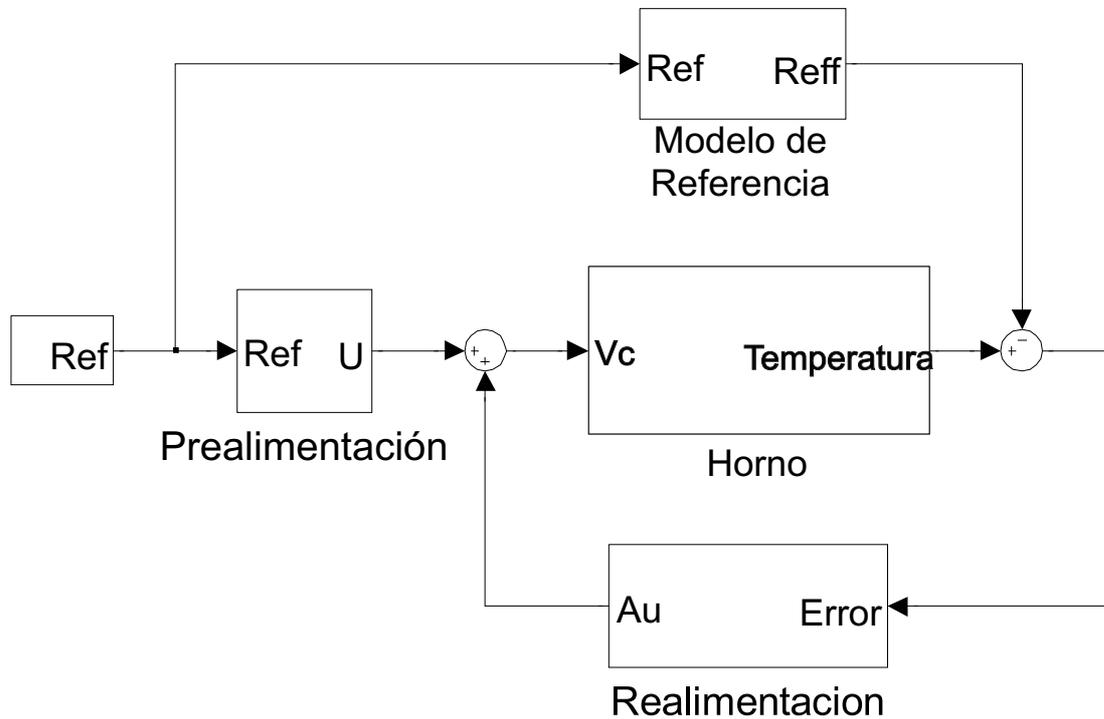


Figura 7.1: Arquitectura de Control con 2 grados de Libertad (2-Dof).

Siendo el modelo del Horno con tres entradas y tres salidas en simulink el que se muestra en la figura 7.2. Por otro lado el modelo en simulink del Horno con 8 entradas (3 Radiadores, 3 Ventilador-Extractor y 2 Sopladores Laterales) y 6 salidas (Todos los Sensores) se muestra en la figura 7.3.

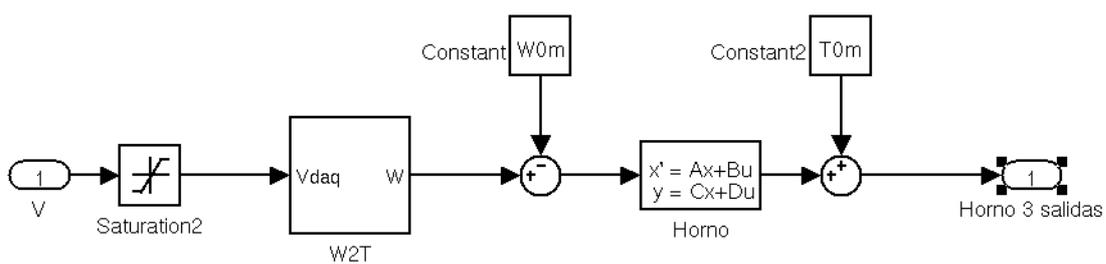


Figura 7.2: Modelo en Simulink del Horno con tres salidas (sensores 2, 4 y 6) y tres entradas (Radiadores):  $G1(s)$ .

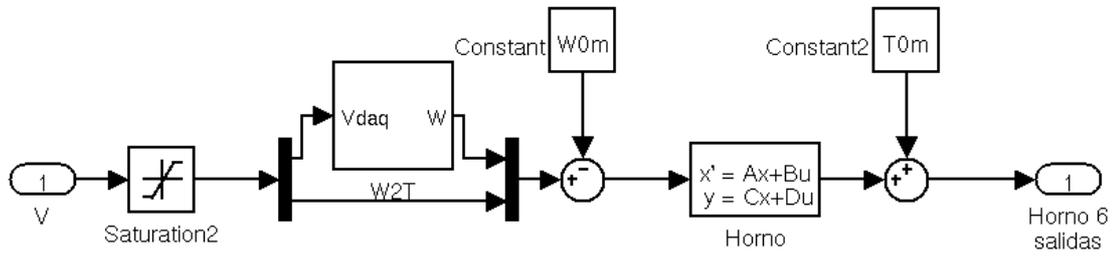


Figura 7.3: Modelo en simulink del Horno con 8 entradas (3 Radiadores, 3 Ventilador-Extractor y 2 Sopladores Laterales) y 6 salidas (Todos los Sensores):  $G2(s)$ .

### 7.3. Control por Realimentación del Estado

El control por re-alimentación del estado estará basado en el diseño de un regulador cuadrático lineal o LQR (Linear Quadratic Regulator). El cual a partir del valor del estado obtendrá la acción de control necesaria para eliminar el error que pueda haber entre la salida y el modelo de referencia.

Para el diseño del controlador será necesario representar el modelo en espacio de estado. Esto se consigue a partir del **modelo entrada/salida** obtenido, donde cada función de transferencia se transformará a espacio de estados, esto se consigue con la función `ss` de Matlab:  $[A, B, C, D] = \text{ss}(\text{modelo})$ .

El resultado obtenido para  $G1(s)$  tendrá 36 estados, con lo que el cálculo del controlador por re-alimentación se complicará considerablemente. Para solucionar esto se ha llevado a cabo un reducción del orden del sistema.

#### Reducción del número de estados

El objetivo que se persigue en este apartado es reducir el número de estados del modelo representado en forma implícita mediante su realización balanceada para poder calcular el controlador óptimo ( $K$ ). Esto se lleva a cabo mediante una serie de transformaciones que se describen a continuación. En primer lugar transformaremos el sistema lineal re-ordenándolo de tal forma que se tenga la variable más controlable y observable como primera en el vector  $x$  y gradualmente se van ordenando todas las variables hasta que quede la menos observable y controlable al final del vector  $x$ .

Si describimos el sistema lineal con las siguientes ecuaciones:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\quad (7.3.1)$$

Y aplicamos una transformación de la forma  $x \cdot T \cdot x$  obtendremos un sistema equivalente:

$$\begin{aligned}\dot{\bar{x}} &= TAT^{-1}\bar{x} + TBu & \rightarrow & \dot{\bar{x}} = A_b\bar{x} + B_bu \\ y &= CT^{-1}\bar{x} + Du & & y = C_b\bar{x} + Du\end{aligned}\quad (7.3.2)$$

A este nuevo sistema se le denominará sistema balanceado donde las variables de estado están ordenadas en función de su observabilidad y controlabilidad. La transformación altera el significado físico de las variables de estado originales, sin embargo si nosotros introducimos una señal de entrada el sistema responderá con la misma señal de salida ( $y$ ) que la del sistema original.

### Reducción utilizando Matlab

Para obtener la realización balanceada se ha utilizado la función `balred` de Matlab: `[SYSbal, g, T, Ti] = balred(SYS, n)` el argumento `SYS` representa el sistema lineal en espacio de estados. El argumento `n` representa el número de estados deseados en el modelo reducido, `SYSbal` es el nuevo sistema transformado, `g` es el peso que el programa ha asignado a cada una de las variables de estado en función de su observabilidad y su controlabilidad, [30], [3].

La Elección del número de estados ( $n$ ) se realizará midiendo la contribución de cada estado al comportamiento de la relación entrada/salida y revisando el transitorio.

Como ejemplo se va a mostrar la reducción para el modelo 3 entradas/3 Salidas ( $G1(s)$ ). Para obtener la contribución de cada estado a la salida utilizaremos los valores singulares de Hankel que equivalen al parámetro de salida  $g$  de la función `balred` de Matlab. Otra forma de obtener estos valores es mediante la función `hsvd` de Matlab, que además devuelve un gráfico que sirve de apoyo para la elección:

```
h=hsvd(SYS)
perc=(100/sum(hsv)).*hsv;
```

1.  $n=3$  contribucion= $\text{sum}(\text{perc}(1:3))=86.8567\%$

2.  $n=6$  contribucion= $\text{sum}(\text{perc}(1:6))=96.0178\%$

3.  $n=9$  contribucion= $\text{sum}(\text{perc}(1:9))=98.6104\%$

Observando la respuesta transitoria del los sistemas reducidos y la contribución de los  $n$  primeros estados se ha escogido el sistema reducido con 9 estados ( $n=9$ ), 6 salidas ( $p=6$ ) y 3 entradas ( $m=3$ ), expresión 7.3.3, obtenida con  $[\text{SYSba1}, g, T, Ti] = \text{balred}(\text{SYS}, 9)$ :

$$\begin{aligned}\dot{x}_{nx1} &= A_{n \times n} \cdot x_{nx1} + B_{n \times m} \cdot u_{mx1} \\ y_{px1} &= C_{p \times n} \cdot x_{nx1} + D_{p \times m} \cdot u_{mx1}\end{aligned}$$

Para mejorar la respuesta en estacionario se calculará el controlador con el sistema balanceado añadiendo tres integradores, uno por cada actuador, que harán que el error tienda a 0 en régimen permanente, (ver expresión 7.3), [3], [39].

$$\begin{aligned}\begin{bmatrix} \dot{x}_{nx1} \\ \dot{x}i_{mx1} \end{bmatrix} &= \begin{bmatrix} A_{n \times n} & B_{n \times m} \\ 0_{m \times n} & 0_{m \times m} \end{bmatrix} \begin{bmatrix} x_{nx1} \\ x i_{mx1} \end{bmatrix} + \begin{bmatrix} B_{n \times m} \\ 0_{m \times m} \end{bmatrix} \\ u_{mx1} y_{px1} &= \begin{bmatrix} C_{p \times n} & 0_{p \times m} \end{bmatrix} \begin{bmatrix} x_{nx1} \\ x i_{mx1} \end{bmatrix} + D_{p \times m} u_{mx1}\end{aligned}\quad (7.3.3)$$

Que expresándolo de forma compacta sería:

$$\begin{aligned}\dot{\bar{x}}_{(n+m)x1} &= \bar{A}_{(n+m) \times n} \cdot \bar{x}_{(n+m)x1} + \bar{B}_{(n+m) \times m} \cdot u_{mx1} \\ y_{px1} &= \bar{C}_{p \times (n+m)} \cdot \bar{x}_{(n+m)x1} + D_{p \times m} \cdot u_{mx1}\end{aligned}\quad (7.3.4)$$

Puesto que ahora el valor de los estados es desconocido será necesario el uso de un observador que con el error calculado a partir de las salidas medidas puede transmitir el valor del estado al controlador óptimo.

### Modelo discreto en espacio de estados reducido

Como el objetivo final es implementar los controladores en un sistema digital se a discretizado el modelo balanceado. Aunque se podría discretizar en los pasos anteriores como paso intermedio o al final, una vez diseñados los controladores .

El periodo de muestreo al que realizaremos el control es a un segundo, ( $t_{md}=1$ ), que podría ser incluso mayor si nos fijamos en las constantes de tiempo de modelo obtenido, ver expresión 6.1. El comando de Matlab utilizado para este fin es  $\text{SYSd}=\text{c2d}(\text{SYSba1}, t_{md}, 'zoh')$ ; donde el argumento  $\text{SYSba1}$  representa es el sistema ya balanceado (expresión 7.3.3),  $t_{md}$  es el tiempo de

muestreo, 'zoh' método de discretización utilizado. El argumento de salida SYSd será nuestro modelo discretizado.

## 7.4. Observador del estado

### 7.4.1. Observabilidad del sistema

Para poder diseñar un observador de nuestro sistema en primer lugar tiene que cumplir la propiedad de observabilidad. Esta propiedad lo que indicara sobre el modelo lineal que se ha reducido, ecuación 7.3.3 es la posibilidad de determinar el vector de estado  $x(t)$  a partir de la salida  $y(t)$ . Este sistema será completamente observable si existe un tiempo finito  $t_1 > 0$  tal que conociendo  $u(t)$  y  $y(t)$ , con  $0 \leq t \leq t_1$ , es suficiente para determinar el estado  $x(0)$ . Una condición necesaria y suficiente para que el sistema sea observable es que el orden del sistema  $n$ , sea igual al rango de la matriz de observabilidad. La matriz de observabilidad del sistema de pila será:

$$O_b = (C \quad C \cdot A \dots C \cdot A_{n-1}) \quad (7.4.1)$$

En el caso del modelo reducido el rango de la matriz  $O_b$  que se ha calculado es 9, por tanto el sistema es observable.

### 7.4.2. Diseño de un observador óptimo (Kalman Filter)

Para poder actuar sobre el sistema real y de esta forma controlarlo, se diseñará un observador basado en el filtro de Kalman que pueda estimar todas las variables de estado del sistema de orden reducido. El estimador de Kalman está formado por un modelo en espacio de estados y obtiene los datos de la covarianza del ruido en la medida y en la entrada. La planta se puede describir linealmente con las ecuaciones:

$$\begin{aligned} x &= Ax + Bu + Gw \\ y &= Cx + Du + Hw + v \end{aligned} \quad (7.4.2)$$

Las matrices  $A$ ,  $B$ ,  $C$  y  $D$  representan la planta linealizada, las matrices  $G$  y  $H$  representan el peso que le damos a la afectación del ruido de entrada en las variables de estado  $x$  y en la salida  $y$  del sistema y respectivamente.

Conociendo la entrada  $u$ , el ruido de entrada  $w$ , ruido en la medición  $v$  y la covarianza de ruido en la medida de la entrada, de la salida y entre salida y entrada:

$$Ewt \cdot wtT = QN \quad Evt \cdot vtT = RN \quad Ewt \cdot vtT = NN \quad (7.4.3)$$

Se puede construir un estimador del estado  $L$  que minimizará el error de la covarianza en el estado estacionario. Siendo el valor del estado estimado:

$$x = Ax + Bu + L(y - Cx - Du) \quad (7.4.4)$$

La ganancia del observador  $L$  se puede calcular, resolviendo la ecuación de Riccati o bien usando directamente la función `kalman` incluida en el toolbox de control de Matlab, [3], [37]. Para utilizar la función `kalman` es necesario entregar a la rutina un sistema dinámico como el de la ecuación 7.4.2:

$$sys = A, [B \ G], C, [DH] \quad (7.4.5)$$

el prototipo de la función es: `[KEST, L, P]=kalman(sys, QN, RN)` donde `KEST` es un sistema dinámico con el observador ya construido, `L` es la ganancia del observador y `P` la solución de la ecuación de Riccati. `SYS` es el sistema descrito en la expresión 7.4.2, `QN` matriz de covarianza del "ruido en los estados" y `RN` matriz de covarianza del ruido en la salida.

Para nuestra aplicación las matrices de pesos utilizadas tendrán la siguiente forma:

$$QN = Ewt \cdot wtT = \begin{pmatrix} q_{obs_1} & 0 & 0 & \dots & 0 \\ 0 & q_{obs_2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & q_{obs_n} \end{pmatrix} \quad (7.4.6)$$

(7.4.7)

$$RN = Evt \cdot vtT = \begin{pmatrix} r_{obs_1} & 0 & 0 & \dots & 0 \\ 0 & r_{obs_2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & r_{obs_n} \end{pmatrix} \quad (7.4.8)$$

Se ha considerado que el ruido de entrada no tiene relación con la salida con lo que la matriz  $H$  será nula. En el caso de la afectación del ruido a las variables de estado ( $G$ ) se ha indicado que todas las variables de estado serán afectadas por igual, quedando  $G$  como matriz diagonal:

$I_{(n+m) \times (n+m)}$ 

$$G = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}_{(n+m) \times (n+m)} \quad (7.4.9)$$

## CÓDIGO

```

1 QN=qobs*eye(n+m);
  RN=robs*eye(p);
3 BG=[modelo.b eye(n+m)]; %B G]
  DH=[modelo.d zeros(p,n+m)]; %D H]
5 SYS=ss(modelo.a,BG,modelo.c,DH);
  [{KEST,L,P}]=kalman(SYS,QN,RN)

```

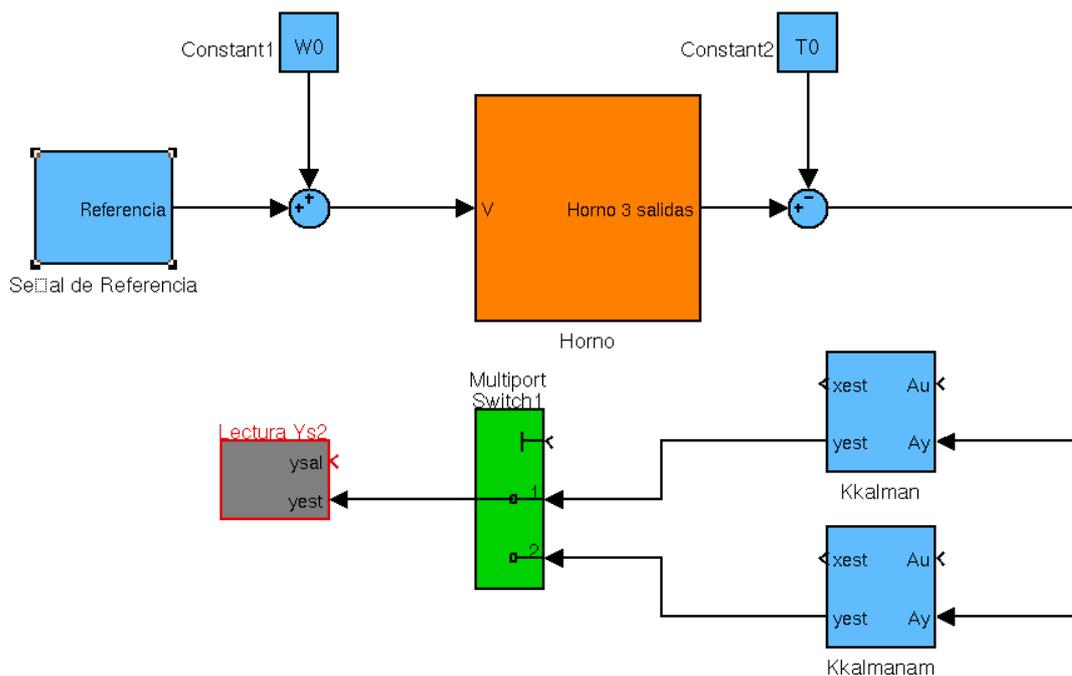


Figura 7.4: Implementación filtro de Kalman en Simulink para  $G1(s)$

Para simular la salida real se ha introducido ruido en la medida mediante el bloque de Simulink *Band – Limited – White – Noise*. Que introduce un ruido blanco.  $Noisepower = 3V$ .

### Sintonización de parámetros QN ( $q_{obs}$ ) y RN ( $r_{obs}$ )

- $q_{obs} \ll r_{obs}$ . Elimina más las perturbaciones en la salida, pero tarda más en converger.
- $q_{obs} \gg r_{obs}$ . Elimina menos las perturbaciones en la salida, pero converge más rápido.

Finalmente se ha optado por  $q_{obs} = r_{obs} = 1$  cuyo resultado se puede ver en la figura 7.5. En la sección C.4.1 se muestra el código utilizado para el diseño de los observadores para el sistema de 3x3.

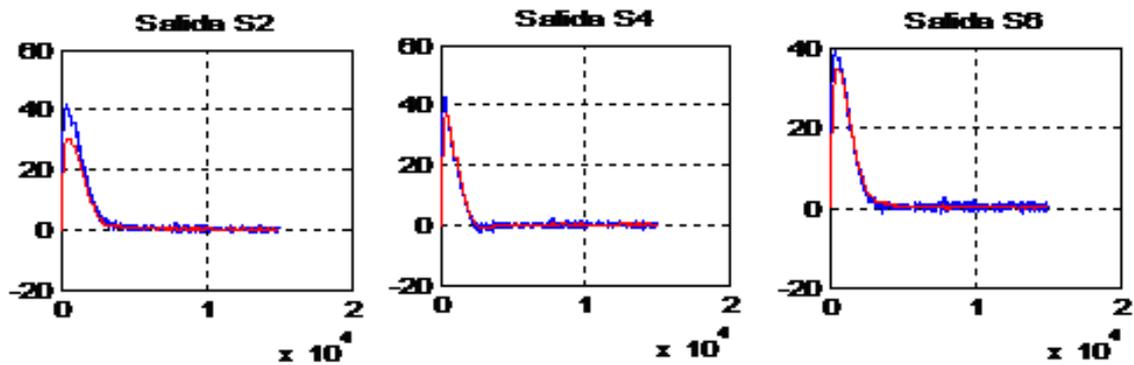


Figura 7.5: Resultados de la simulación

## 7.5. Diseño del Controlador

### 7.5.1. Controlabilidad del sistema

Una condición necesaria y suficiente para que en el sistema reducido de orden  $n = 9$  sea completamente controlable es que la matriz de controlabilidad sea de rango completo.

La matriz de controlabilidad se calculara con:

$$C_o = (B \cdot A \cdot B \dots A_{n-1} \cdot B) \quad (7.5.1)$$

En el caso del Horno la matriz de controlabilidad se calcula con la instrucción de Matlab  $C_o = \text{ctrb}(\text{SYS})$  el rango de la matriz  $C_o$  obtenido es de rango completo. Por tanto el sistema es completamente controlable.

### 7.5.2. Controlador LQR

Para realizar el control óptimo cuadrático por realimentación de estados del sistema:

$$\dot{x} = Ax + Buy = Cx + Du \quad (7.5.2)$$

se determinará el vector  $K$  según la ley de control  $u(t) = -K\bar{x}(t)$  con el objetivo de minimizar la función de coste:

$$J = \int_0^{\infty} (y^T \hat{Q}y + u^T \hat{R}u + 2y^T \hat{N}u)dt \quad (7.5.3)$$

Que se puede expresar:

$$J = \int_0^{\infty} (\bar{x}^T Q\bar{x} + u^T Ru + 2\bar{x}^T Nu)dt \quad (7.5.4)$$

donde  $Q$  es una matriz hermética definida positiva (o semidefinida positiva) o simétrica real y  $R$  es una matriz hermética definida positiva o simétrica real. La matriz  $Q$  determina la importancia relativa del error y  $R$  determina el coste energético de la acción de control.

La ley de control obtenida mediante la ecuación  $u(t) = -Kx(t)$  es la ley de control óptimo por realimentación de estados. Por tanto si se determina los elementos desconocidos de la matriz  $K$  utilizando el problema de optimización de la función de coste  $J$  sujeta a las restricciones de 7.5.2, entonces  $u(t) = -Kx(t)$  es óptima para cualquier estado inicial  $x(0)$ .

A partir de la ecuación de Riccati (expresión 7.5.5) se calcula  $S$  y posteriormente se obtiene  $K_{opt}$  con la expresión 7.5.6.

$$ATS + SA(SB + N)R^{-1}(BTS + NT) + Q = 0 \quad (7.5.5)$$

$$K = R^{-1}(BTS + NT) \quad (7.5.6)$$

Si la solución  $S$  que se obtiene de la ecuación de Riccati devuelve una matriz definida positiva indicará que el sistema es estable o la matriz  $A - BK$  es estable.

### 7.5.3. Diseño LQR

El diseño LQR será sobre la planta linealizada y las matrices del sistema lineal se han escogido del sistema ampliado 7.3.

La ganancia del observador  $K$  se puede calcular utilizando la función `lqr` y `lqry` incluida en el toolbox de control de Matlab. Para utilizar la función `lqr` o `lqry` es necesario entregar a la rutina un sistema dinámico con la siguiente descripción:

$$sys = A, B, C, D$$

Los parámetros de las funciones son  $[K_{opt}, S_o, E_o] = \text{lqr}(sys, Q_r, R_r)$  y  $[K_{opt}, S_o, E_o] = \text{lqry}(modelo, Q, R)$  donde  $K_{opt}$  es el controlador óptimo,  $S_o$  la solución de la ecuación de Riccati y  $E_o$  los valores propios de sistema en bucle cerrado  $(\bar{A} - \bar{B}K)$ ,  $Q$  matriz de ponderación de las salidas,  $R$  matriz de ponderación de las acciones de control.

`lqry` funciona igual que `lqr`, pero el diseño de  $Q$  es más intuitivo puesto que se ponderan directamente las salidas y no los estados, de los cuales al ser un sistema reducido no se conoce su significado físico.

$$J = \int_0^{\infty} (y^T \hat{Q} y + u^T \hat{R} u + 2y^T \hat{N} u) dt \quad (7.5.7)$$

La equivalencia entre los pesos de `lqry` y `lqr` viene dada por:

$$\begin{pmatrix} \hat{Q} & \hat{N} \\ \hat{N}^T & \hat{R} \end{pmatrix} = \begin{pmatrix} \bar{C} & 0 \\ D & I \end{pmatrix} \begin{pmatrix} Q & N \\ N^T & R \end{pmatrix} \begin{pmatrix} \bar{C} & D \\ 0 & I \end{pmatrix} \quad (7.5.8)$$

\* $Q$  debe ser definida positiva.

#### Sintonización de parámetros. Modelo 3x3, $G1(s)$

Se ha creado un archivo simulink como el que se muestra en la figura 7.6 y a partir de las simulaciones se han obtenido los parámetros de  $\hat{Q}_y$  y  $\hat{R}$ . En la sección C.4.1 se muestra el código

utilizado para el diseño de los controladores.

$$\hat{Q}_y = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \hat{R} = \begin{pmatrix} 0,0001 & 0 & 0 \\ 0 & 0,0100 & 0 \\ 0 & 0 & 0,0001 \end{pmatrix} \quad (7.5.9)$$

En la figura 7.7 se muestra un simulación del bucle cerrado donde  $x(0) \neq 0$ . En la figura 7.8 se

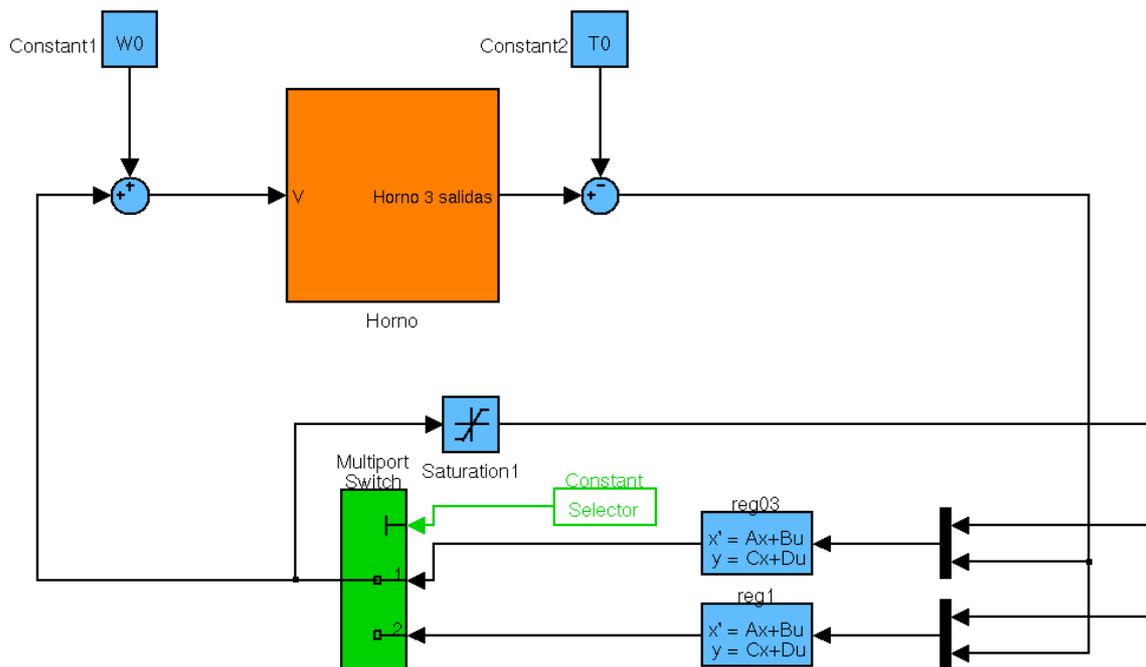


Figura 7.6: Bucle de Realimentación y Controlador + Observador.

muestras las acciones de control.

## 7.6. Control en Bucle Abierto. Modelo 3x3, $G1(s)$

**Desacoplador completo:** Considerando que nuestro modelo del proceso es de tres entradas por tres salidas ( $G(s)_{3 \times 3}$ ) y que se puede expresar como  $G(s)_{3 \times 3} = R(s) \cdot N(s)$ , siendo  $R(s)$  el común divisor por filas, se podría obtener una matriz  $A(s) = N^{-1}(s)$  de forma que se consiga

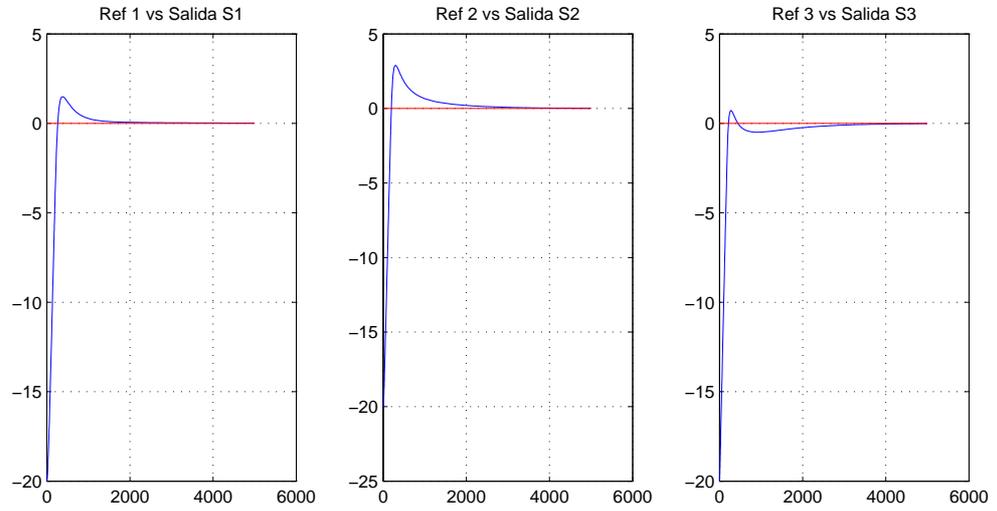


Figura 7.7: Salida del modelo

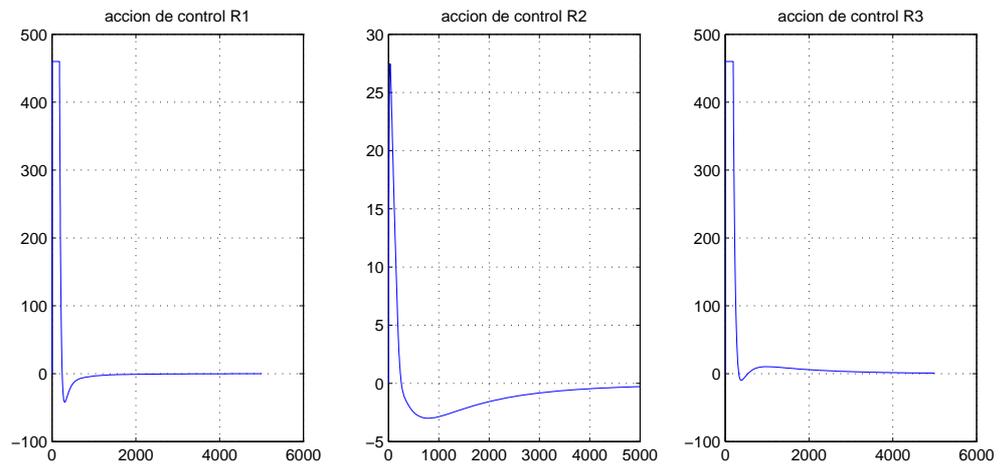


Figura 7.8: Acción de control en  $W$ .

que el bucle abierto,  $BA(s)_{3 \times 3} = \frac{y(s)}{ref(s)}$ , [3], [44], sea:

$$\begin{aligned} BA(s)_{3 \times 3} &= G(s)_{3 \times 3} \cdot A(s) = R(s) \cdot N(s) \cdot A(s) \\ \text{donde } N(s) &= A^{-1}(s) \text{ y por tanto} \\ G(s)_{3 \times 3} \cdot A(s) &= R(s) \cdot A^{-1}(s) \cdot A(s) = R(s) \end{aligned} \quad (7.6.1)$$

Y si  $R(s)$  es estable e invertible, podríamos obtener  $R^{-1}(s)$  tal que:

$$BA(s)_{3 \times 3} = \frac{y(s)}{ref(s)} = R^{-1}(s) \cdot R(s) = I_{3 \times 3} \quad (7.6.2)$$

Teniendo en cuenta las limitaciones de las acciones de control ( $[0 \dots 500W]$ ) utilizaremos un modelo de referencia,  $M_{ref}(s)$ , que alcance la consigna de referencia en un tiempo y una sobreoscilación realizables. Siendo el regulador de prealimentación resultante es  $Gr(s) = A(s) \cdot R^{-1}(s) \cdot M_{ref}(s)$ .

#### CÓDIGO

```
T_establecimiento=500;
2 wn=inv(T_establecimiento/10);
  Ej=1;
4 Q=tf(wn^2,1 2*wn*Ej wn^2)]; %Modelo de Referencia
  Mref=zpk(Q)*eye(num_salidas)^3;
6 N=minreal(inv(R)*G);
  A=minreal(inv(N));
8 Gr=A*inv(R)*Mref;
```

Se ha creado un archivo simulink como el que se muestra en la figura 7.9 considerando las limitaciones de las acciones de control. Los resultados se muestran en las figuras 7.10 y 7.11.

### 7.7. Implementación Control 2-DoF para Modelo $G1(s)$ (3x3)

Se ha introducido una medida de antiwindup en la realimentación para que el observador del bucle cerrado pueda calcular el estado a partir de la acción real de control aportada a la planta por el controlador óptimo, véase figura 7.12. De esta forma si  $U + \Delta u \geq V_{sat}$  o  $U + \Delta u \leq V_{sat}$  la

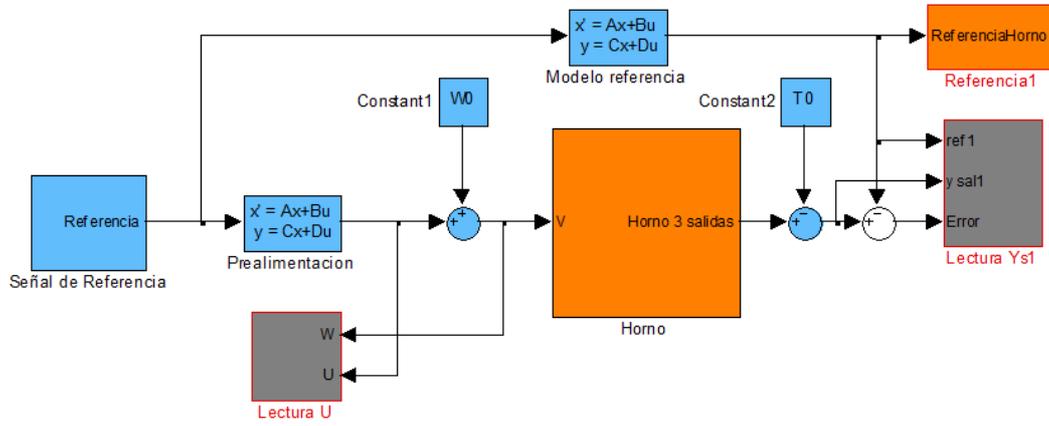


Figura 7.9: Bucle de Abierto: Prealimentación + Horno.

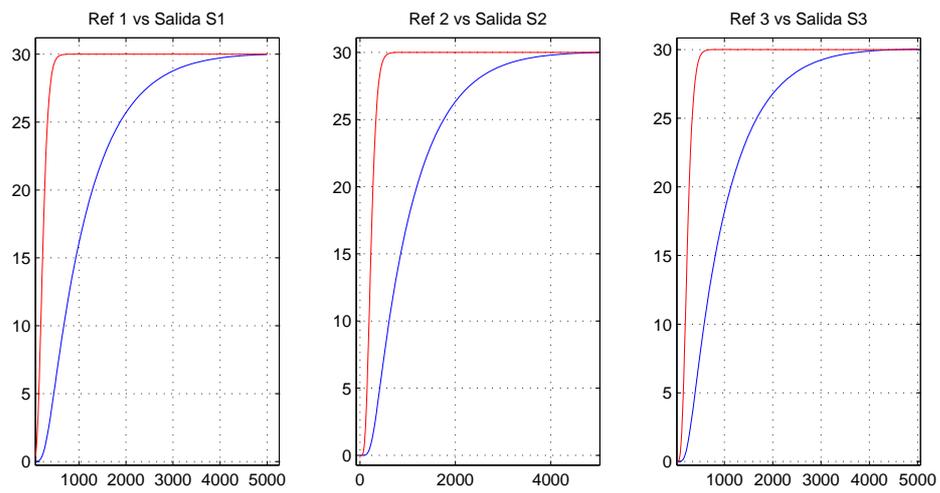


Figura 7.10: Salida del modelo

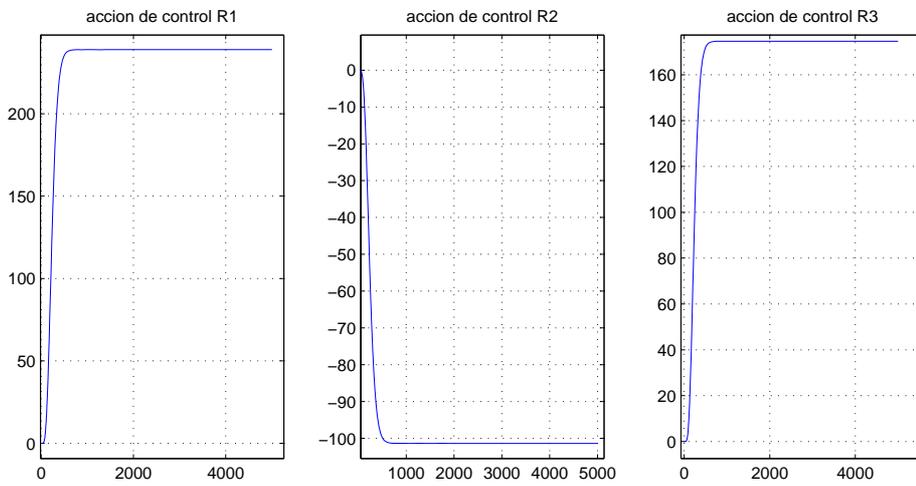


Figura 7.11: Acción de control en  $W$ .

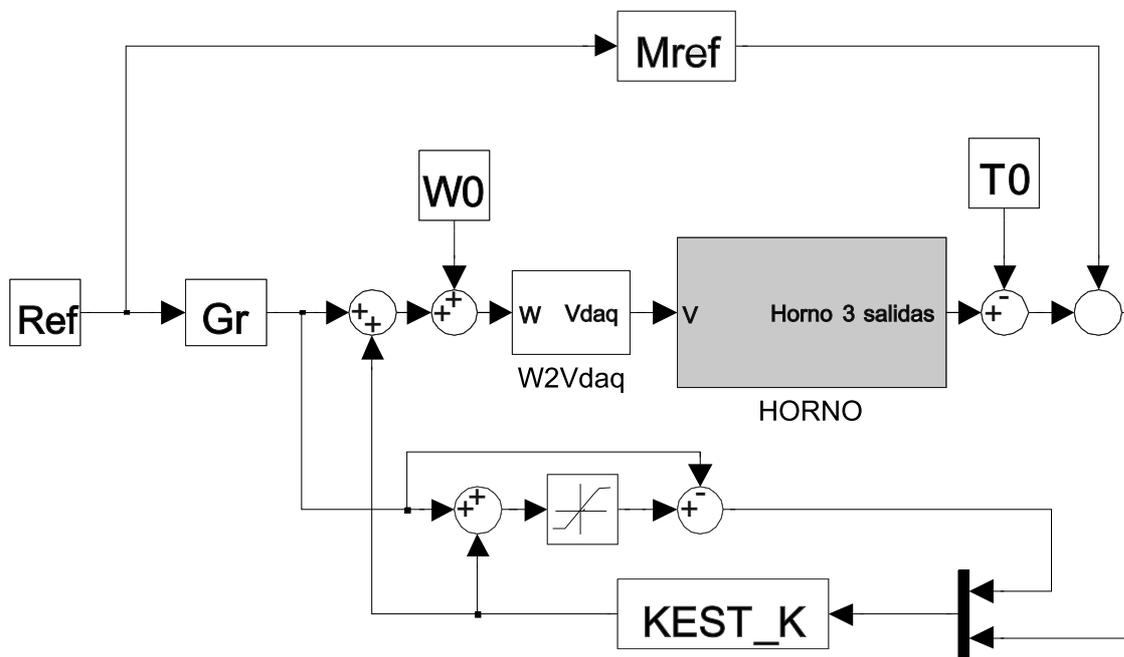


Figura 7.12: Arquitectura de Control 2-Dof Implementada.

acción del control re-alimentado  $\Delta u$  será interpretada como nula por el observador a pesar de la salida del controlador óptimo.

La implementación se ha hecho mediante Matlab y posteriormente se ha introducido en un Math-Script de Labview para tener una etapa de puesta a punto en el Scada de identificación (ver sección 8.2) y Scada de Control (sección 8.3). Para ver el código ver sección C.5.

### 7.7.1. Simulación

Los parámetros de la simulación son los que se definen en la siguiente rutina, los resultados se muestran en la figura 7.13 y 7.14. Se puede ver que no existe error en régimen permanente y que se intenta seguir el modelo de referencia, pero las saturaciones limitan dicho seguimiento.

```

load ( 'reg16 ', 'reg16 ' )
2  %PARAMETROS PARA SIMULINK
   Tsim=5000;
4  %SATURACIONES
   Wmax=[460;480;490];
6  Wmin=[0;0;0];
   Vventmax=[10;10;10;10;10];
8  Vventmin=[0;0;0;0;0];
   %PUNTOS DE FUNCIONAMIENTO
10 W0m=[250;80;320;6;6;6;6;6];
   T0m=[44;55;70;65;70;55];
12 W0=W0m;
   T0=T0m;
14 refer=[45;55;65;65;65;55];
   Vdaqmax=10;
16 Vdaqmin=0;

```

### 7.7.2. Resultados Reales

Los siguientes resultados muestran un experimento de control para el modelo  $G(1)$ . La implementación se ha hecho en Labview, mediante MATHLAB scripts el código se puede ver en el anexo C.5. (En Labview corresponderá a la fase de puesta a punto en los dos Scadas, ver secciones 8.2 y 8.3).

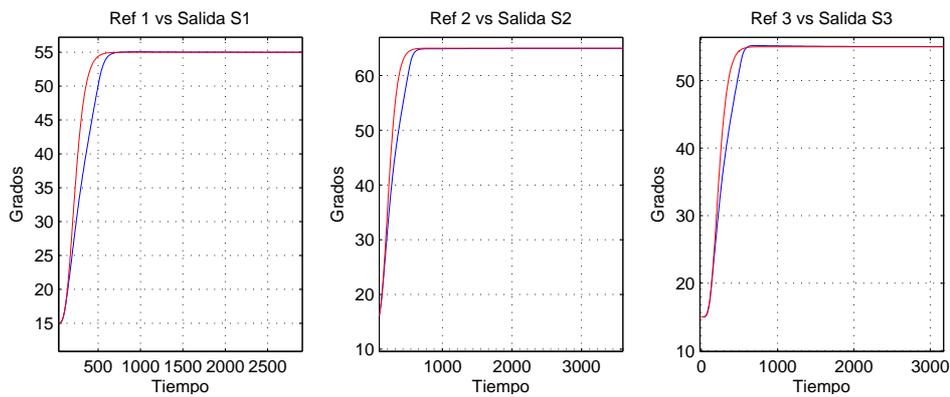


Figura 7.13: Seguimiento de referencias: sensor 2, sensor 4 y sensor 6.

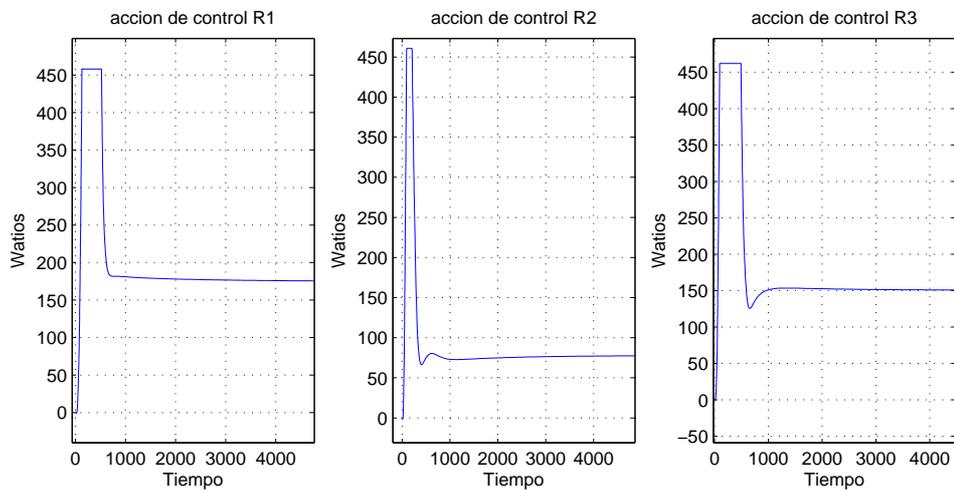


Figura 7.14: Acciones de control en los 3 Radiadores.

**Experimento:** Se pide un aumento de temperatura de  $5\text{ }^{\circ}\text{C}$  en todos los sensores. Más tarde se cambia la referencia para que sensor 2 y 6 vuelvan al punto de operación y el sensor 4 aumente  $5\text{ }^{\circ}\text{C}$ , ver figura 7.15.

En la gráfica inferior de la figura 7.16, se muestran las señales reales que se transmiten a la tarjeta de adquisición de datos, estas se obtienen del mismo modo que se hace en la sección 2.2.1 pero con el ajuste opuesto.

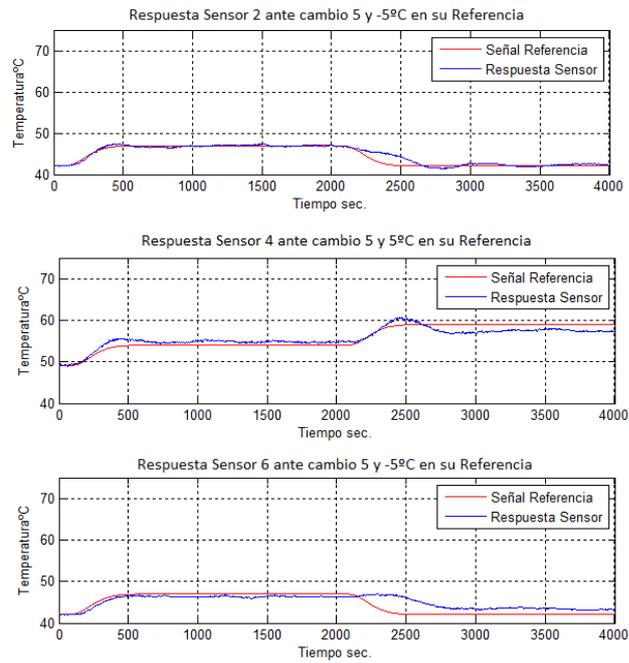


Figura 7.15: Respuestas de los sensores frente a las Referencias indicadas.

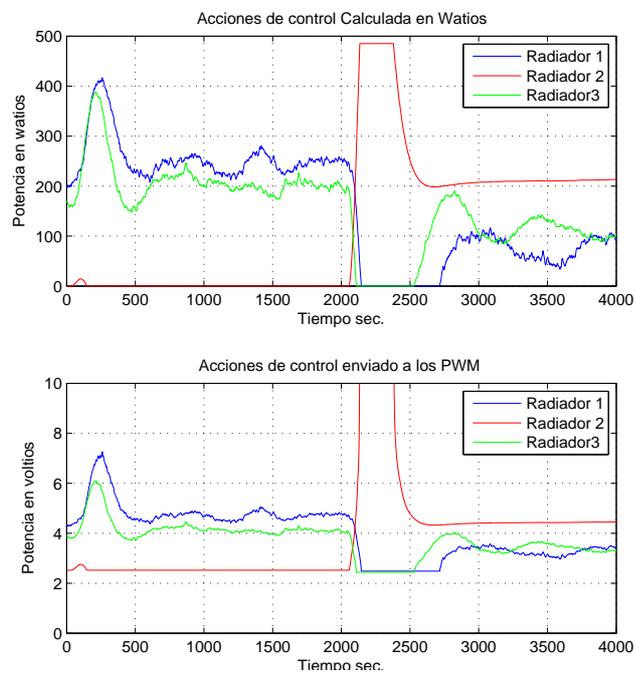


Figura 7.16: Acciones de control.

## 7.8. Implementación Control 2-DoF para Modelo Nominal $G^2(s)$ (8x6)

En este caso el diseño de la realimentación es idéntico al realizado para el modelo de  $3 \times 3$ . La reducción del modelo en este caso tendrá 15 estado más 8 integradores uno por cada entrada. En la sección C.4.2 se puede ver como se genera el sistema ampliado y se muestra el código utilizado para obtener los observadores y controladores.

Respecto a la prealimentación no es posible el diseño de un desacoplador completo, como se comprueba en la expresión 7.8.1 se muestra la matriz de ganancia relativa, donde se aprecia que no es posible el emparejamiento.

```
Gest=dcgain (Forn)
2 Test=pinv (Gest ')
rga0=Gest.* Test
```

$$rga0 = \begin{bmatrix} 0,054 & -0,038 & -0,311 & 0 & -0,266 & -0,325 & 1,306 & 0,581 \\ -0,083 & 0,075 & 0,700 & -0,631 & 0,536 & 0,726 & -0,322 & 0 \\ -0,110 & 0,130 & 0,588 & 0,046 & 0,324 & 0,023 & 0 & 0 \\ -0,170 & 0,227 & 0,378 & 0,594 & -0,005 & 0 & 0 & -0,026 \\ 0,700 & -0,471 & -0,925 & -0,7210 & 1,660 & 0 & 0,757 & \\ -0,320 & 0,176 & 0,422 & 1,706 & 0,412 & -1,084 & 0 & -0,312 \end{bmatrix} \quad (7.8.1)$$

En esta caso no se podrá tener una prealimentación con dinámica, que compense la dinámica del sistema y siga al modelo de referencia ( $M_r$ ). Así que se va utilizar la función inversa de la matriz de ganancia estática (Gest), aunque en realidad utilizaremos la pseudo-inversa por que el sistema no es cuadrada. Para calcular la pseudo-inversa utilizaremos la función `pinv` de Matlab.

En el siguiente fragmento se muestra el código empleado para obtener la Prealimentación:

```
1 %Modelo de referencia
t_establecimiento=1000;
3 wn=inv (t_establecimiento/10);
Ej=0.5;
5 Mr=zpk (tf (wn^2,[1 2*wn*Ej wn^2])*eye (salidas));
%Calculo de Prealimentación
7 invProcess0=pinv (dcgain (Forn));
Gr0=series (Mr,invProcess0);
```

7.8.1. Simulaciones

En la figura 7.17 muestra la implementación llevada a cabo. Esta misma implementación se ha llevado a realidad tal como se muestra en la sección 8.3. Los resultados de la simulación se muestran en la figura 7.18. Se puede ver que existen error en régimen permanente, a pesar de que las referencias no difieren mucho del valor de la temperatura en el punto de funcionamiento.

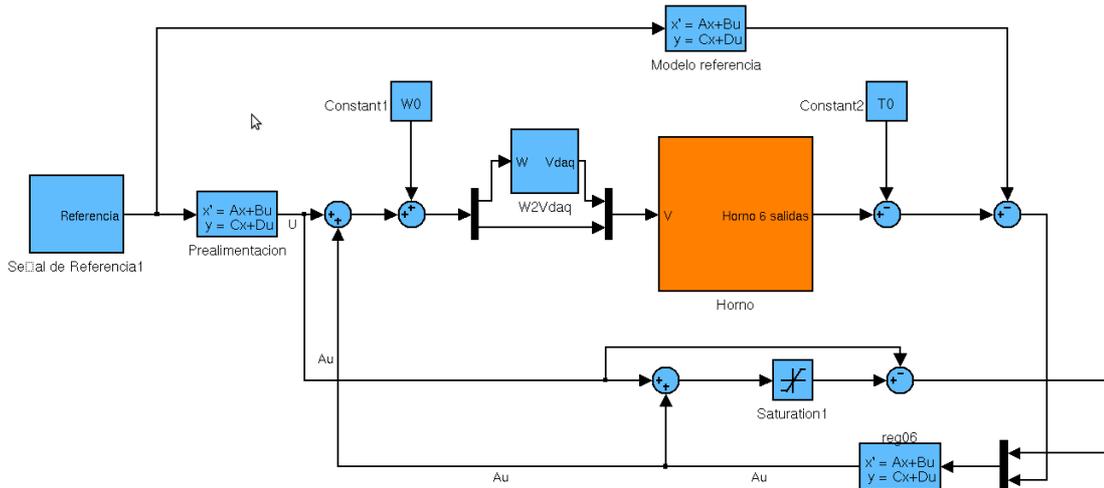


Figura 7.17: Seguimiento de referencias

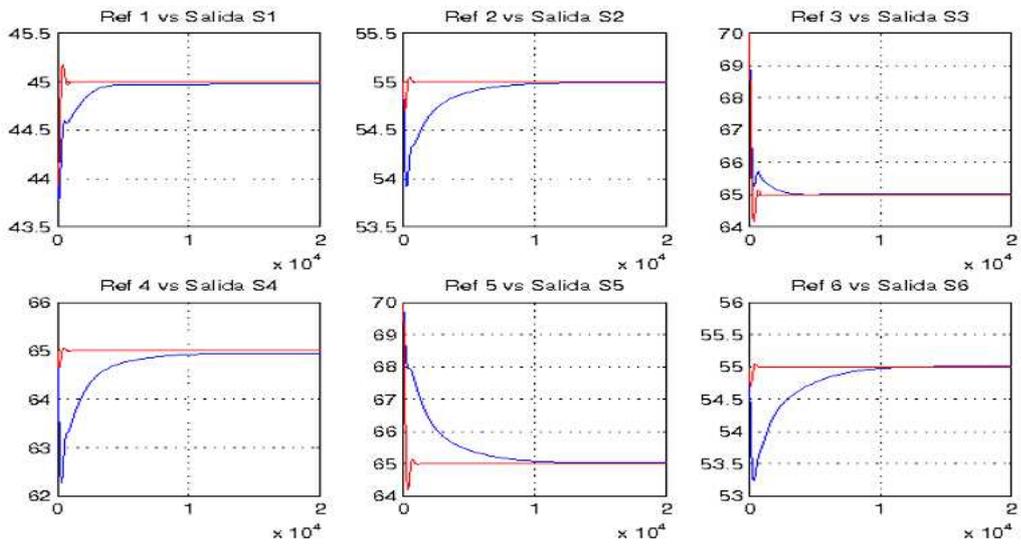


Figura 7.18: Seguimiento de referencias

### 7.8.2. Resultados Reales

Para implementar este control, primero se lleva el sistema al punto de funcionamiento con el controlador diseñado para el sistema  $G^1(s)$ , y posteriormente aplicamos el control diseñado para  $G^2(s)$ . Esto se ha implementado con un Scada hecho en Labview. En la sección 8.3 se puede encontrar las capturas, mientras que el código utilizado se puede encontrar en la sección C.6.5.

**Experimento:** Durante 1200 segundos se lleva el sistema al punto de funcionamiento, una vez alcanzado aplicamos 3 escalos de referencia en los sensores 1, 3, y 5  $\rightarrow$   $ref1=+3$ ,  $ref3=-2.5$  y  $ref5=-5$ , durante 2200 segundos, ver figura 7.19 donde se distinguen las dos etapas de control. Además se puede observar que el acoplamiento entre salidas y entradas no permite que se controlen los seis sensores.

La figura 7.20 y 7.21 representan las acciones de control de los Radiadores y Ventiladores/Extrac-tores&Sopladores, respectivamente.

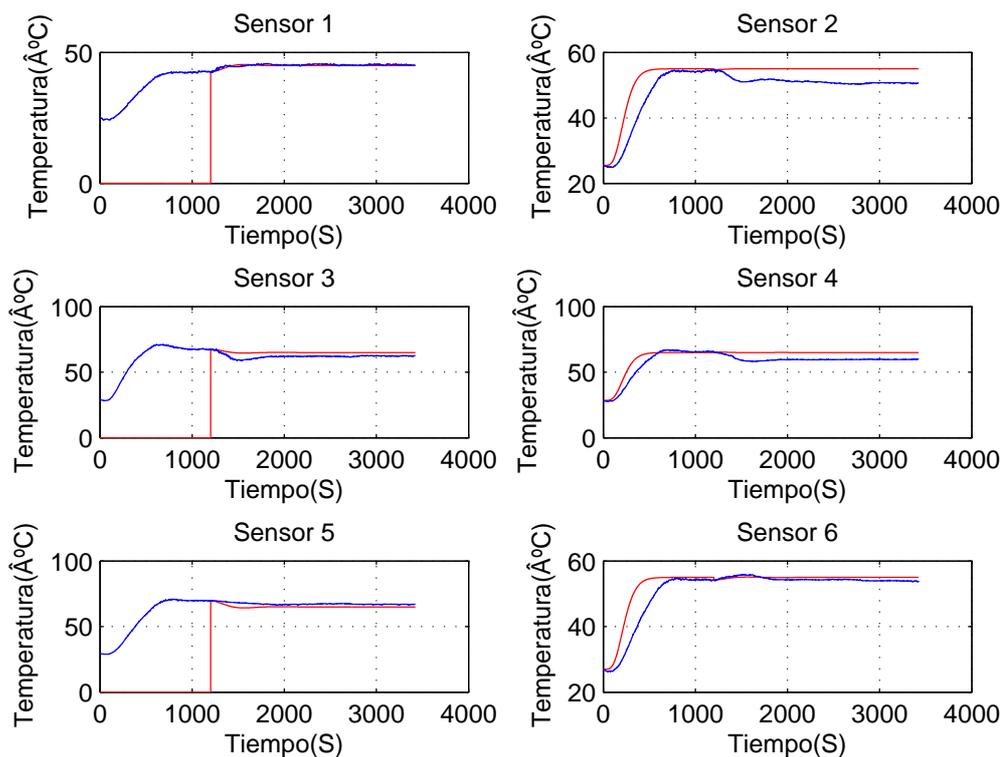


Figura 7.19: Seguimiento de referencias

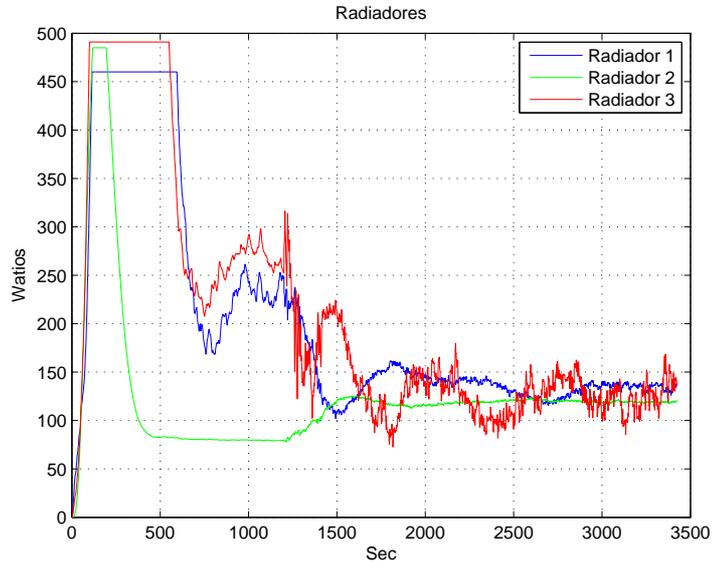


Figura 7.20: Acciones de Control de los Radiadores

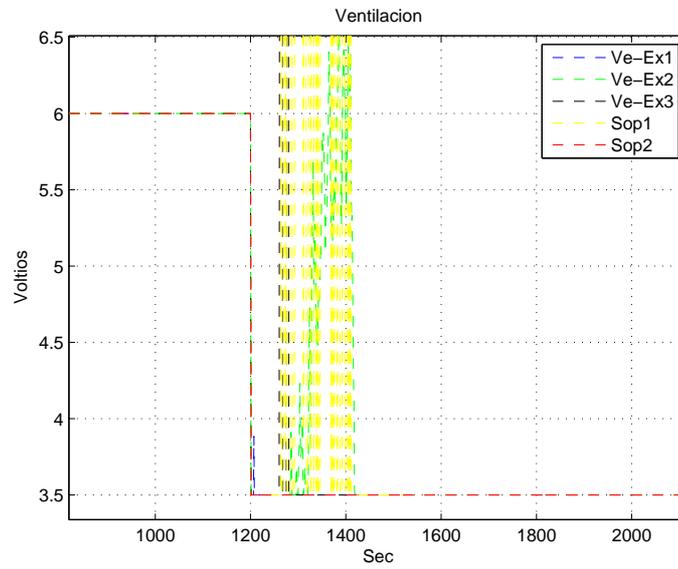


Figura 7.21: Acciones de Control de Ventiladores/Extractores&Sopladores

Debido a que las ganancias de los ventiladores son muy pequeñas comparadas con los radiadores (ver figura 6.4 el experimento de la figura 7.19 se ha realizado con referencias pequeñas (+3, 0, -2.5, 0,-5, 0). Ya que si los ventiladores no son capaces de enfriar a la velocidad necesaria el error aumenta rápidamente, ver figura 7.22, y los actuadores saturan, ver figuras 7.23 y 7.24.

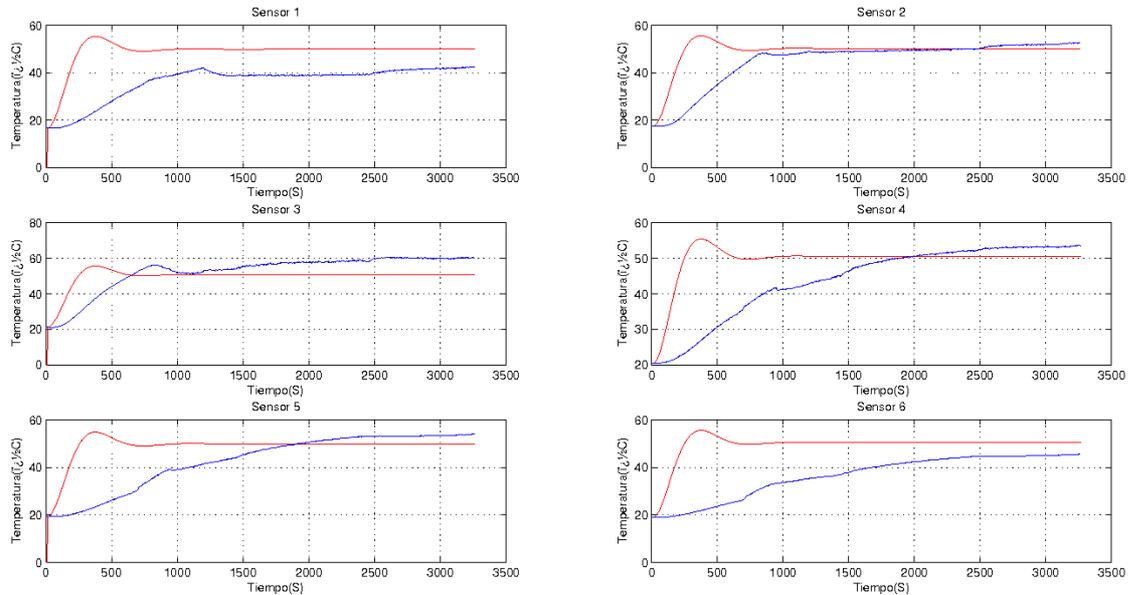


Figura 7.22: Seguimiento de referencias

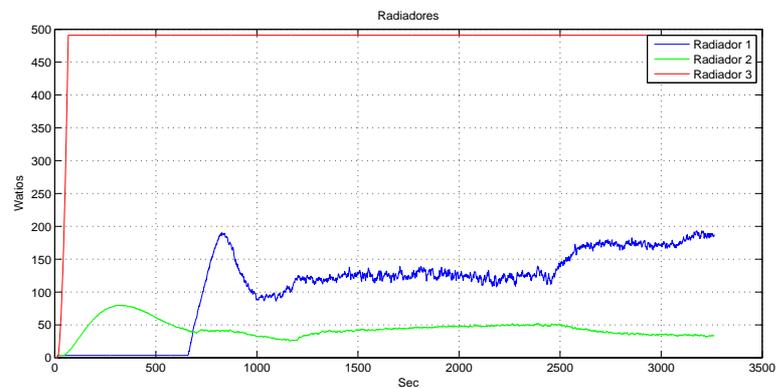


Figura 7.23: Acciones de Control de los Radiadores

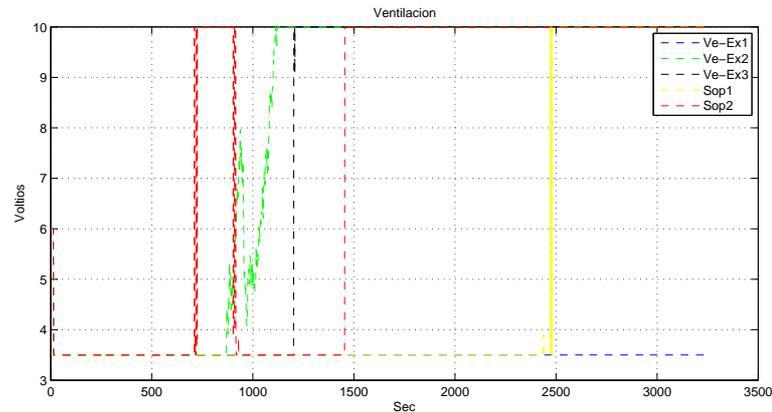


Figura 7.24: Acciones de Control de Ventiladores/Extractores&Sopladores

### 7.8.3. Posibles soluciones a realizar

El siguiente paso es encontrar que punto de funcionamiento consigue que las ganancias de todos los actuadores sean de la misma magnitud.

En la figura 7.25 se muestra la respuesta de los sensores ante los ventiladores al máximo y sin la cinta transportadora en el horno, la cual dificultaba la extracción de calor. Mientras que en la figura 7.26 se muestra la respuesta de los sensores ante los radiadores al máximo y sin cinta. Se puede ver que a temperaturas más bajas la pendiente es más pronunciada que en el descenso, para igualar esta pendiente tendríamos que subir el punto de funcionamiento de  $\approx 60^{\circ}\text{C}$  a  $\approx 80^{\circ}\text{C}$  que es la temperatura límite de la cinta y de los sensores inalámbricos.

Otra medida, la cual podría permitir utilizar la cinta y los sensores inalámbricos es la instalación de un sistema de refrigerado que posibilite la inyección de aire frío, ya que trabajando alrededor de  $60^{\circ}\text{C}$  el aire a temperatura ambiente no es adecuado.

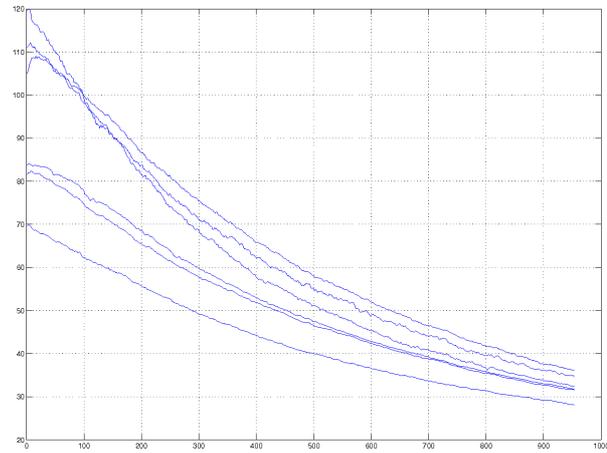


Figura 7.25: Descenso de temperatura provocado por los ventiladores a potencia máxima

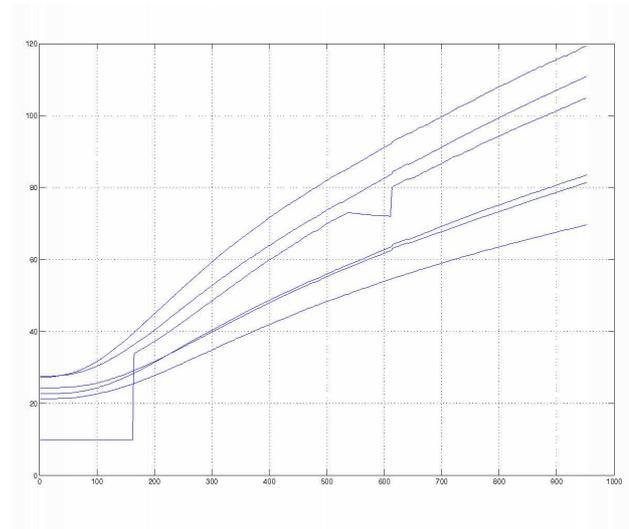


Figura 7.26: Aumento de temperatura provocado por los radiadores a potencia máxima

## Capítulo 8

# Anexo de Interfaces con el usuario (HMI)

### 8.1. Introducción

- HMI de Identificación Modelo Nominal: Aprovechando los scripts creados para el control [C.5](#) se ha creado una interface para identificar el proceso en un punto de funcionamiento deseado y frente a un escalón determinado. Se lleva el sistema al punto de funcionamiento deseado mediante el sistema de control de puesta a punto [C.5](#) y seguidamente se aplica el escalón indicado por el usuario.
- HMI de Control Horno: Para implementar los lazos de control y facilitar la interacción con el usuario se ha diseñado una aplicación que permite de forma rápida la variación de parámetros tales como tiempo de experimento, referencias, etc. Además de permitir cargar controladores diseñados con Matlab, almacena datos en `.m`.

## 8.2. HMI de Identificación

### Labview Control Panel

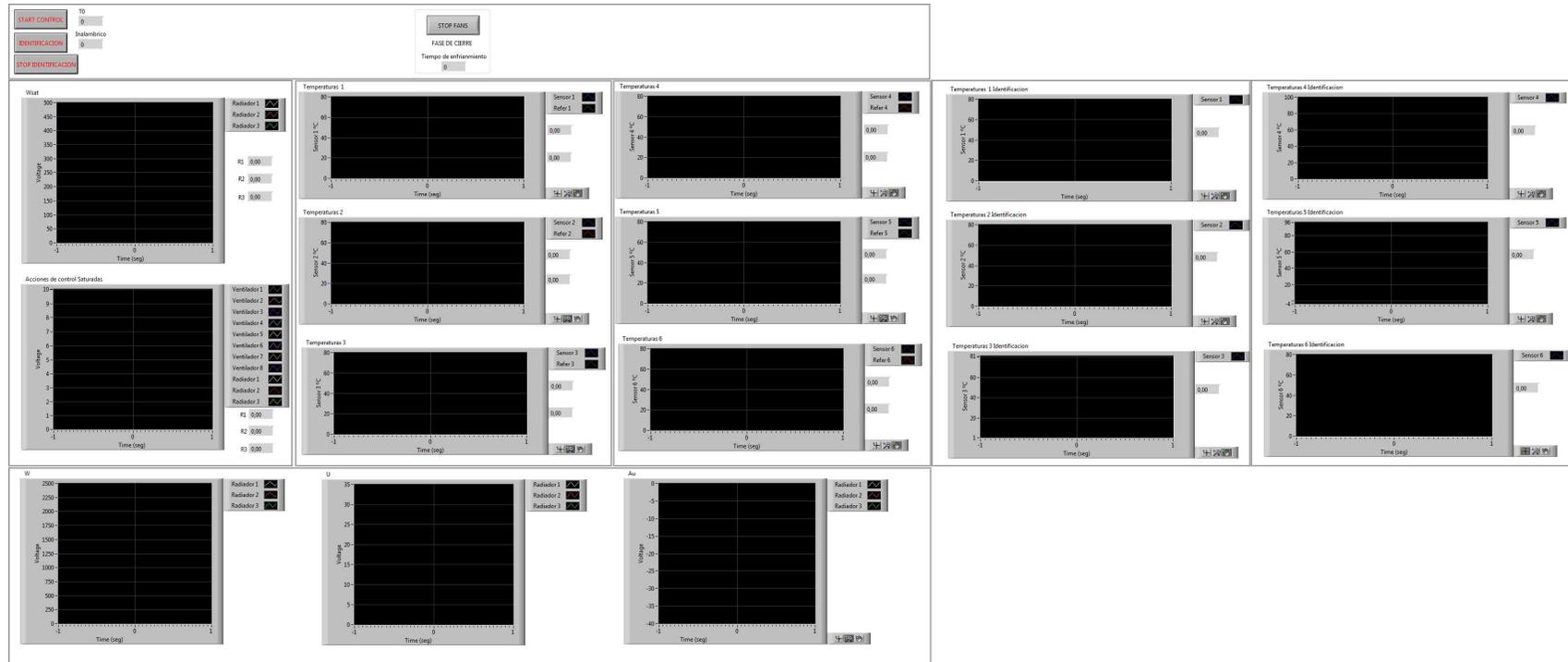


Figura 8.1: En la parte izquierda del frontal se ve el control realizado para llevar al sistema al punto de funcionamiento (con las acciones de control) la respuesta del sistema frente al cambio en escalón de uno de los actuadores.

## Labview Block Diagram

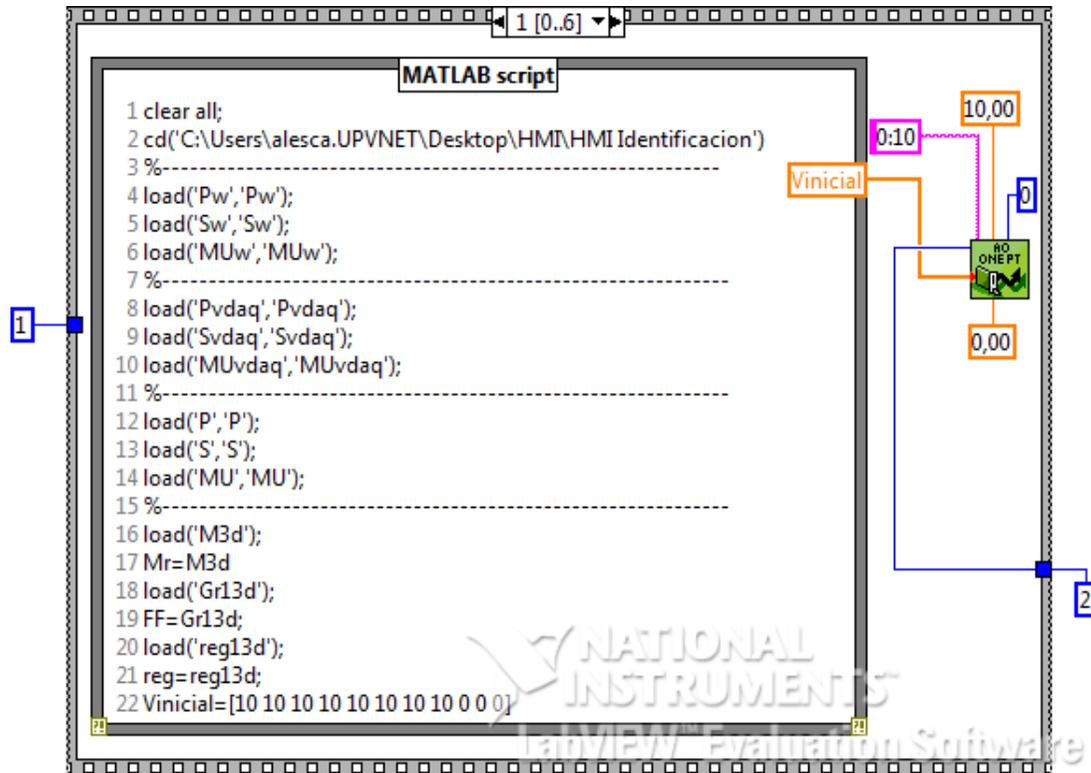


Figura 8.2: Inicialización del programa: Se indica el path donde están las variables y se cargan los parámetros para transformar los datos de V a °C y de V a W. También se cargan los datos de los controladores para la puesta a punto.

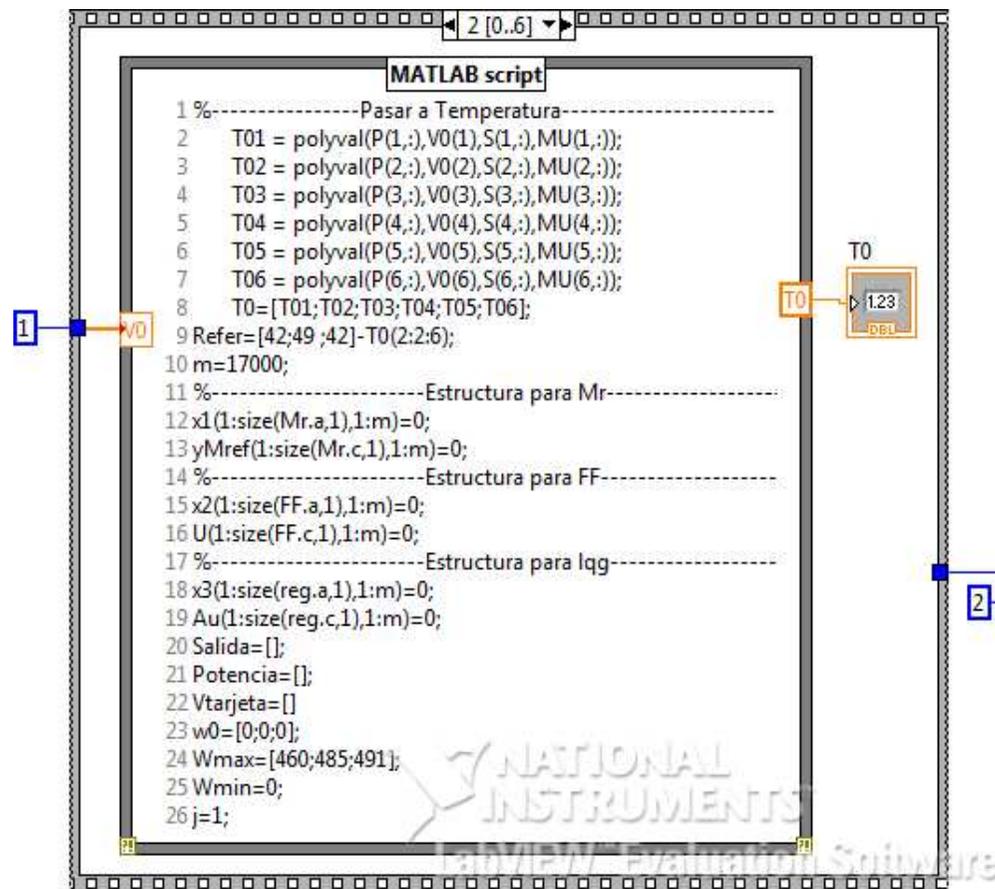


Figura 8.3: Se obtiene la temperatura inicial y se crea las variables de control a utilizar.

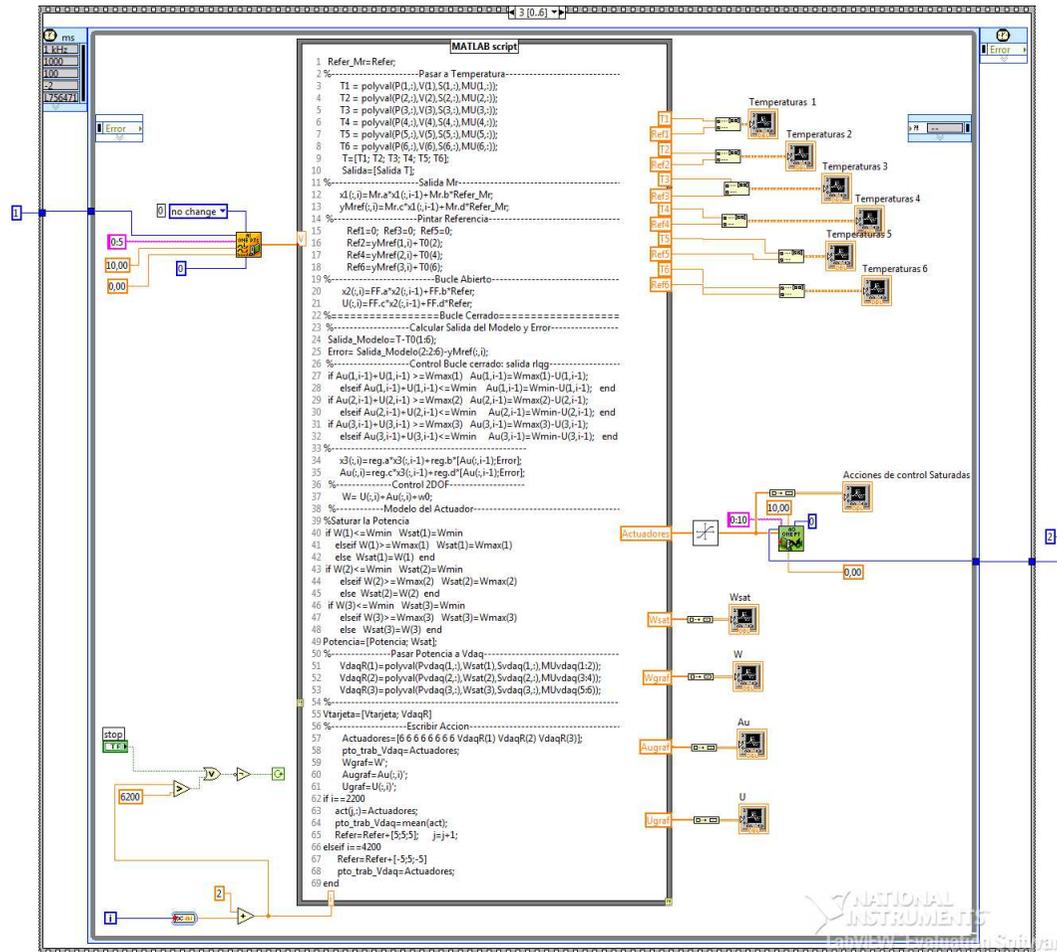


Figura 8.4: Se implementa el código de control para llevar al sistema al punto de funcionamiento, ver código en anexo C.5.

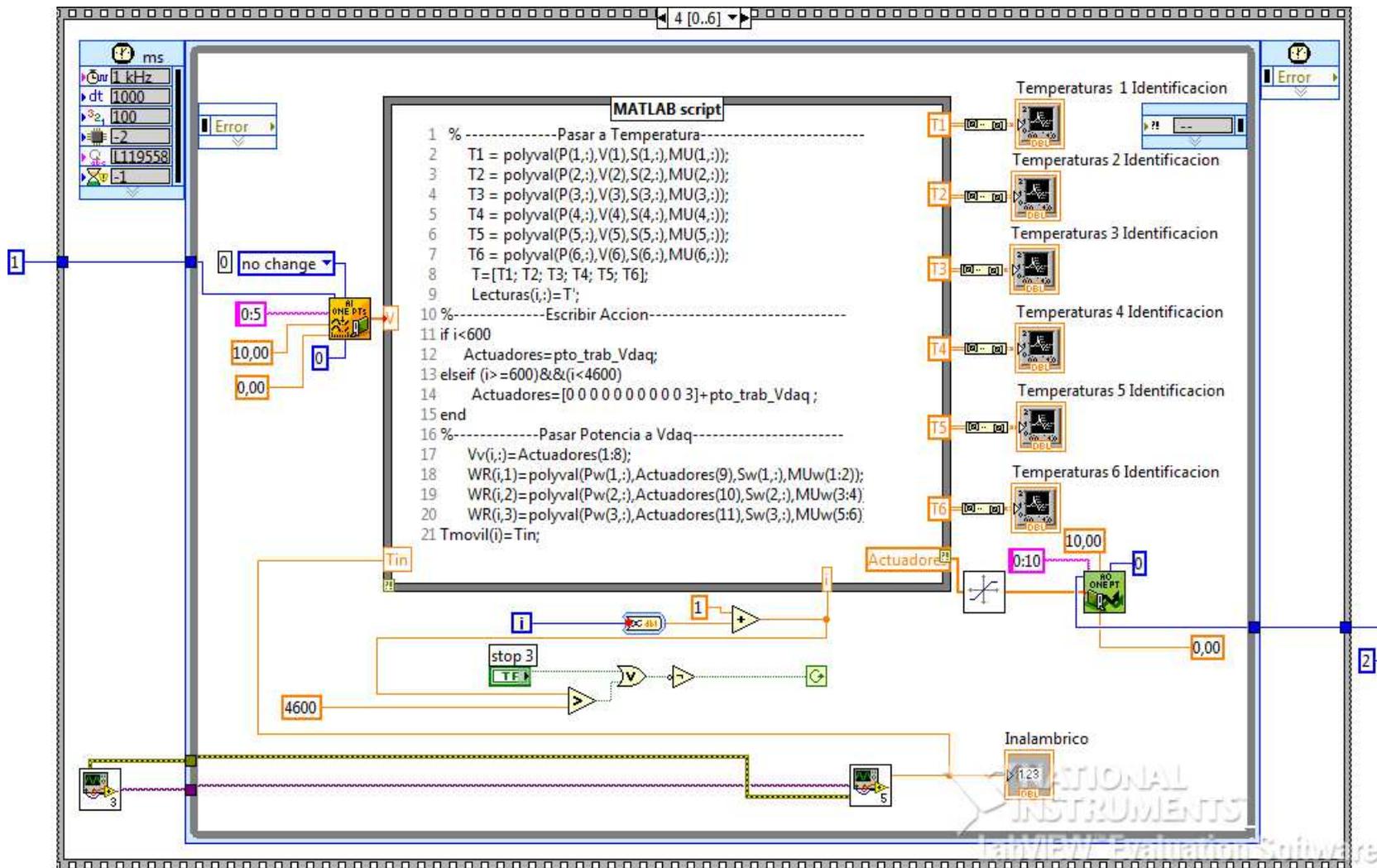


Figura 8.5: Se captura el valor de la temperatura en el punto de funcionamiento espero 10 minutos a que estabilice y se aplica el escalón de 3V en uno de los actuadores y dicha actuación se transforma en Watts para posteriormente salvarla.

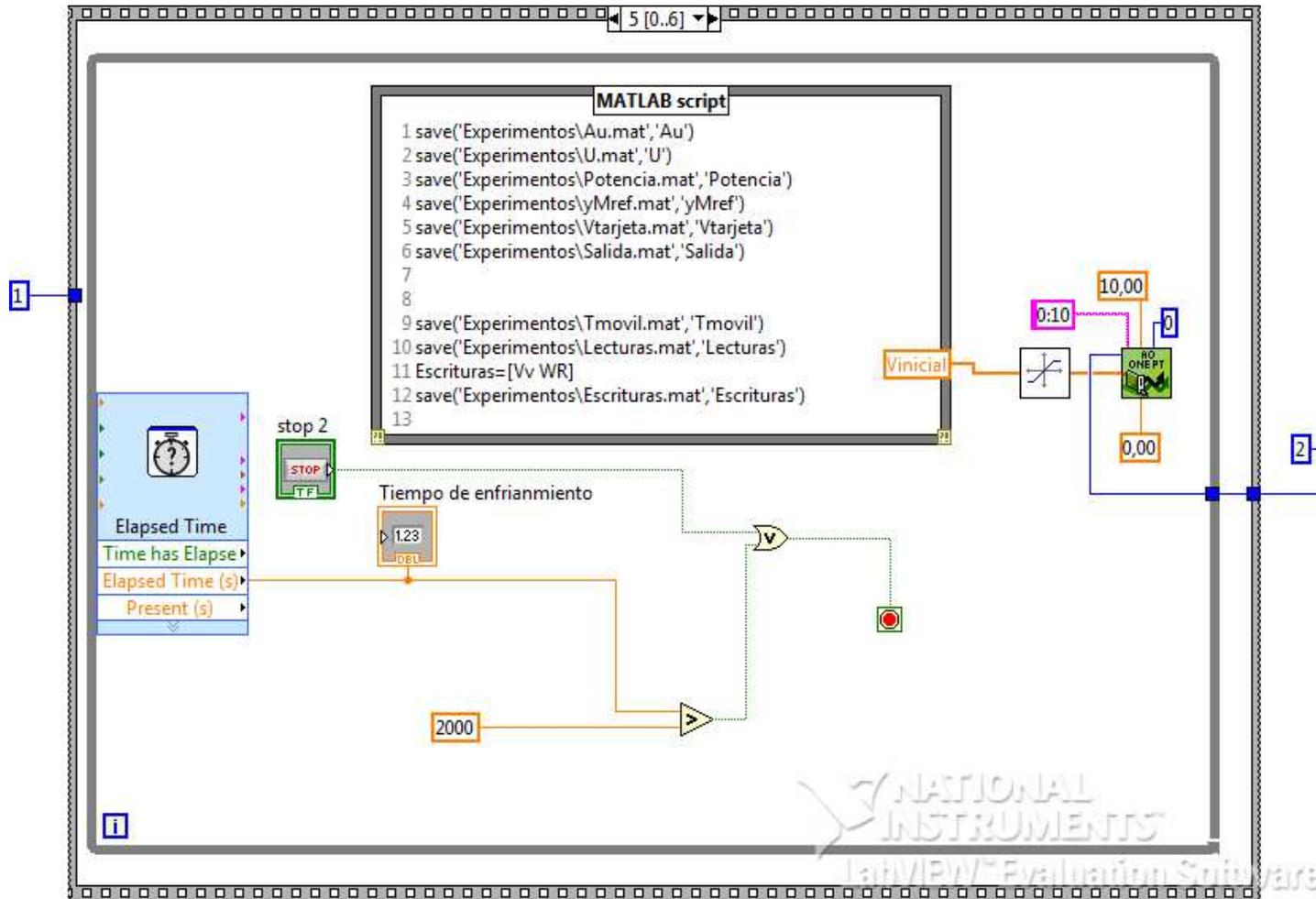


Figura 8.6: Se salvan los datos en .mat para que puedan ser leídos por matlab y refrigeramos el horno apagando los calentadores y encendiendo al máximo los ventiladores.

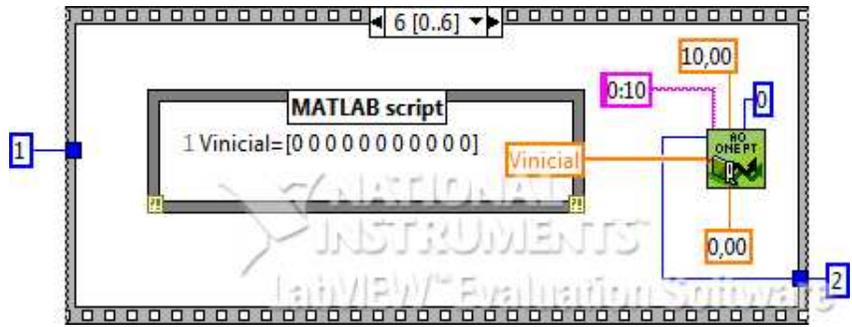


Figura 8.7: Se para todo.

## 8.3. HMI de Control Horno

## Labview Panel Control

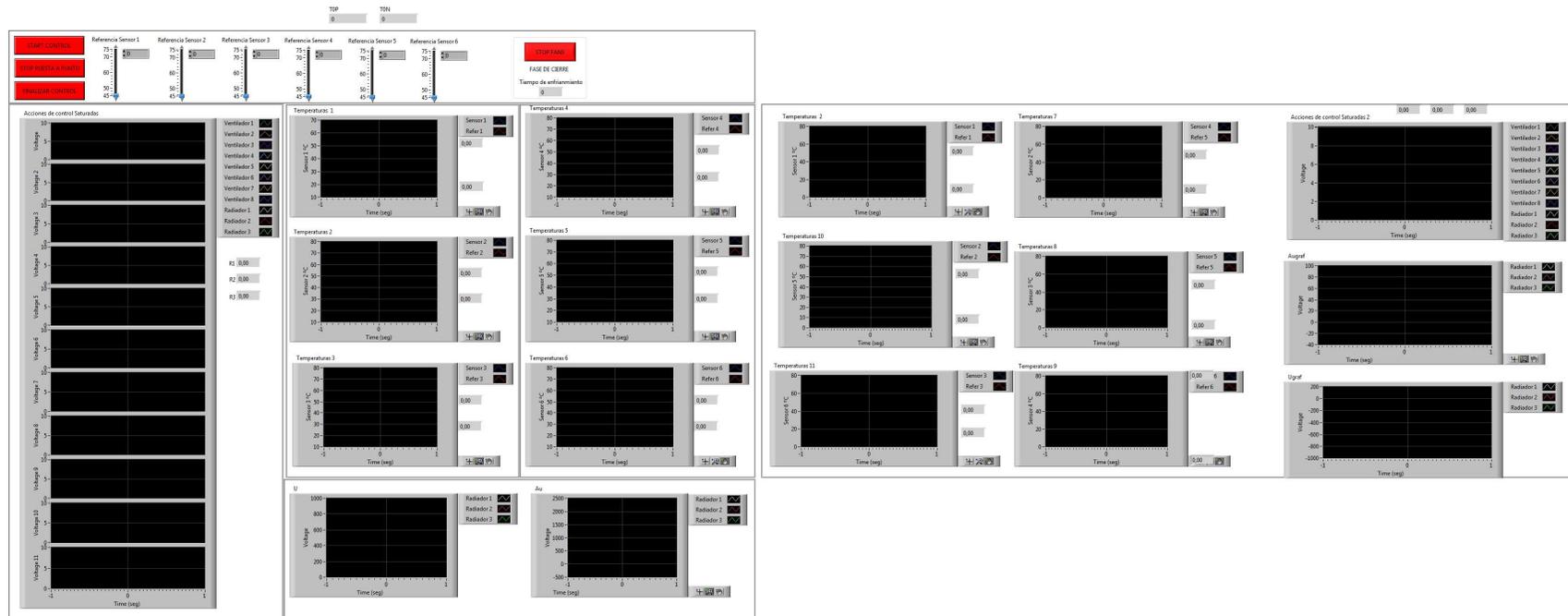


Figura 8.8: En la parte izquierda del frontal se ve el control realizado para llevar al sistema al punto de funcionamiento (con las acciones de control) y en la parte derecha el control del horno completo (con las acciones de control).

Labview Block Diagram

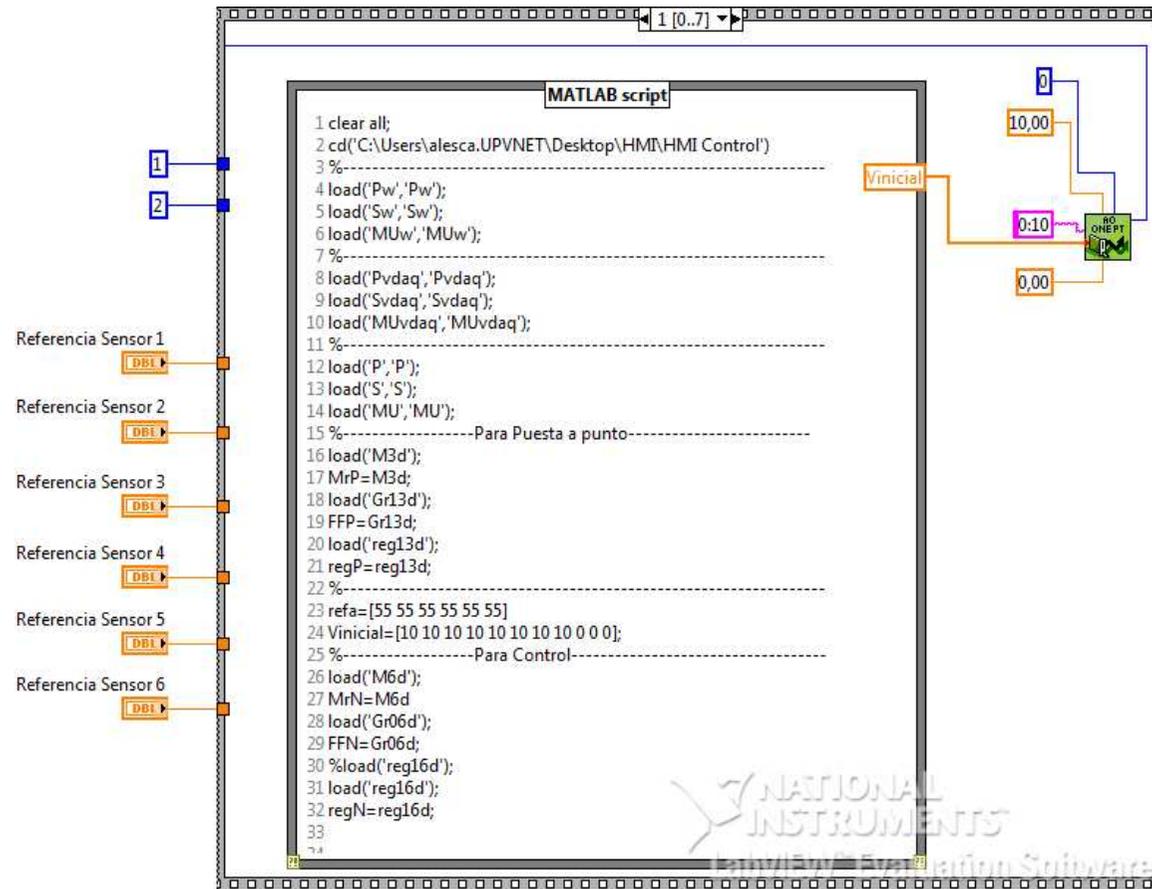


Figura 8.9: Inicialización del programa: Se indica el path donde están las variables y se cargan los parámetros para transformar los datos de V a °C y de V a W. También se cargan los datos de los controladores para la puesta a punto.

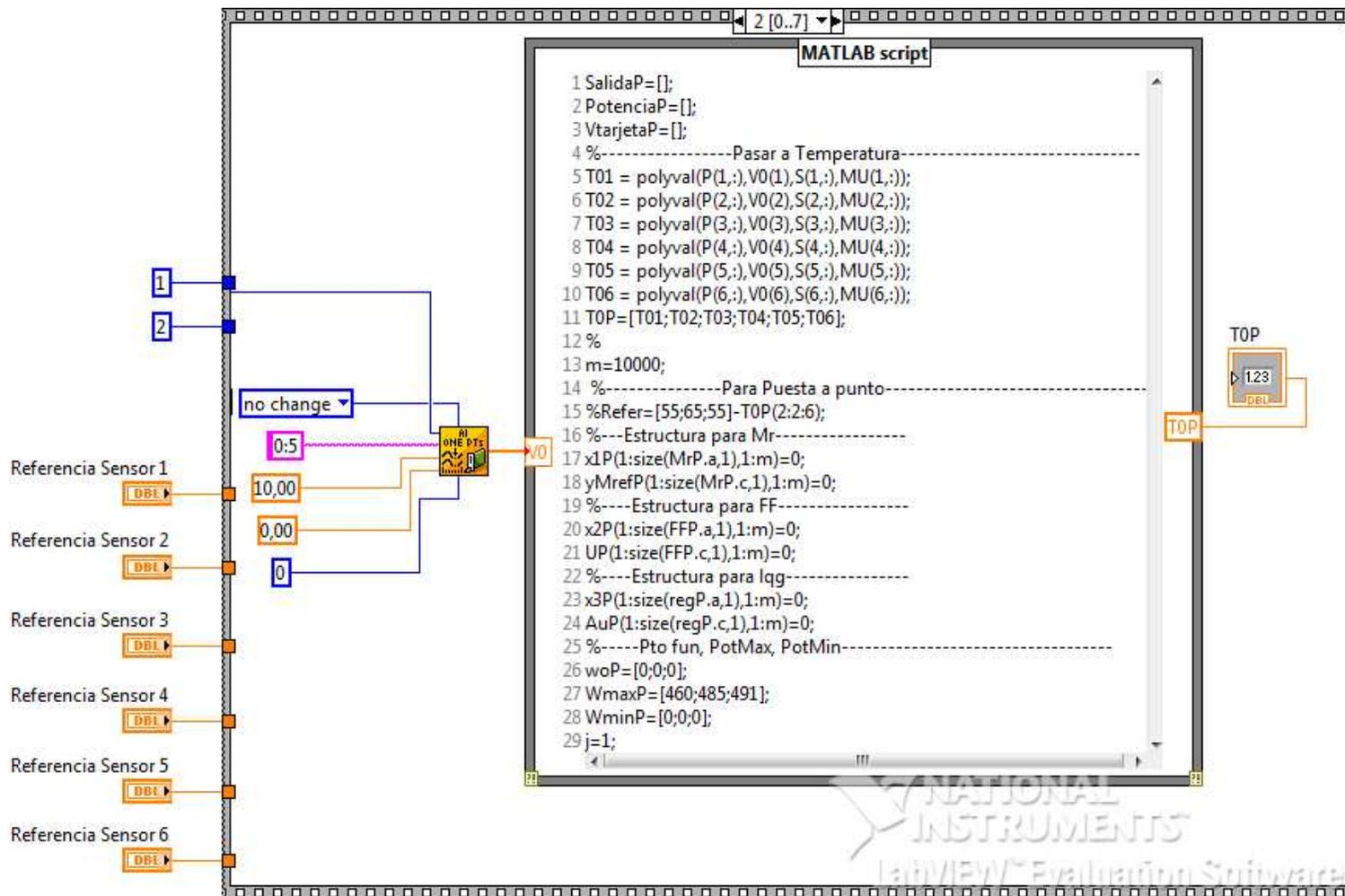


Figura 8.10: Se obtiene la temperatura inicial y se crean las variables de control a utilizar.

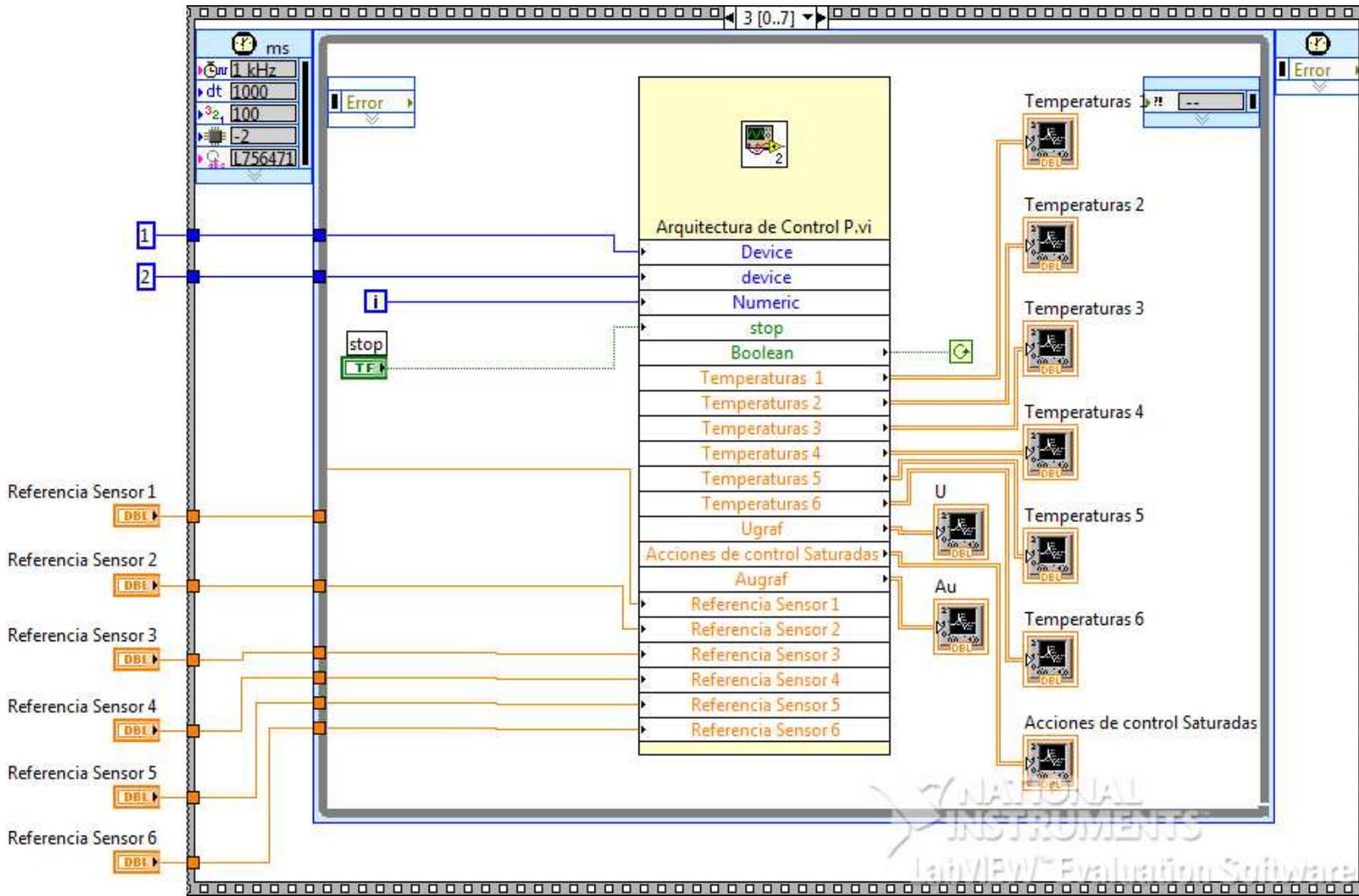


Figura 8.11: Implementación del código de control para llevar al sistema al punto de funcionamiento, ver código en anexo C.5.

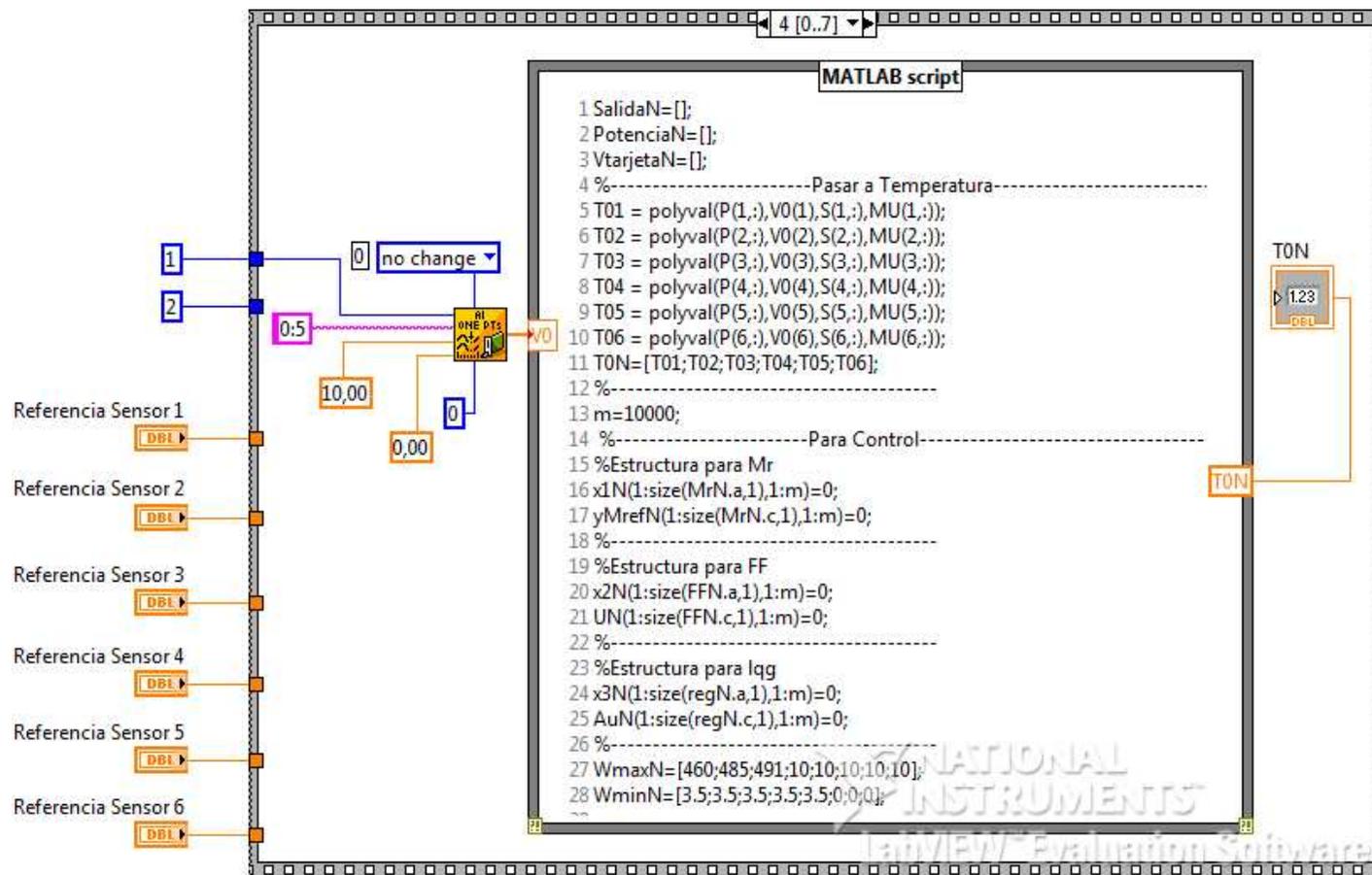


Figura 8.12: Se captura el valor de temperatura en el punto de funcionamiento y cargamos los reguladores para control todos los actuadores y sensores.

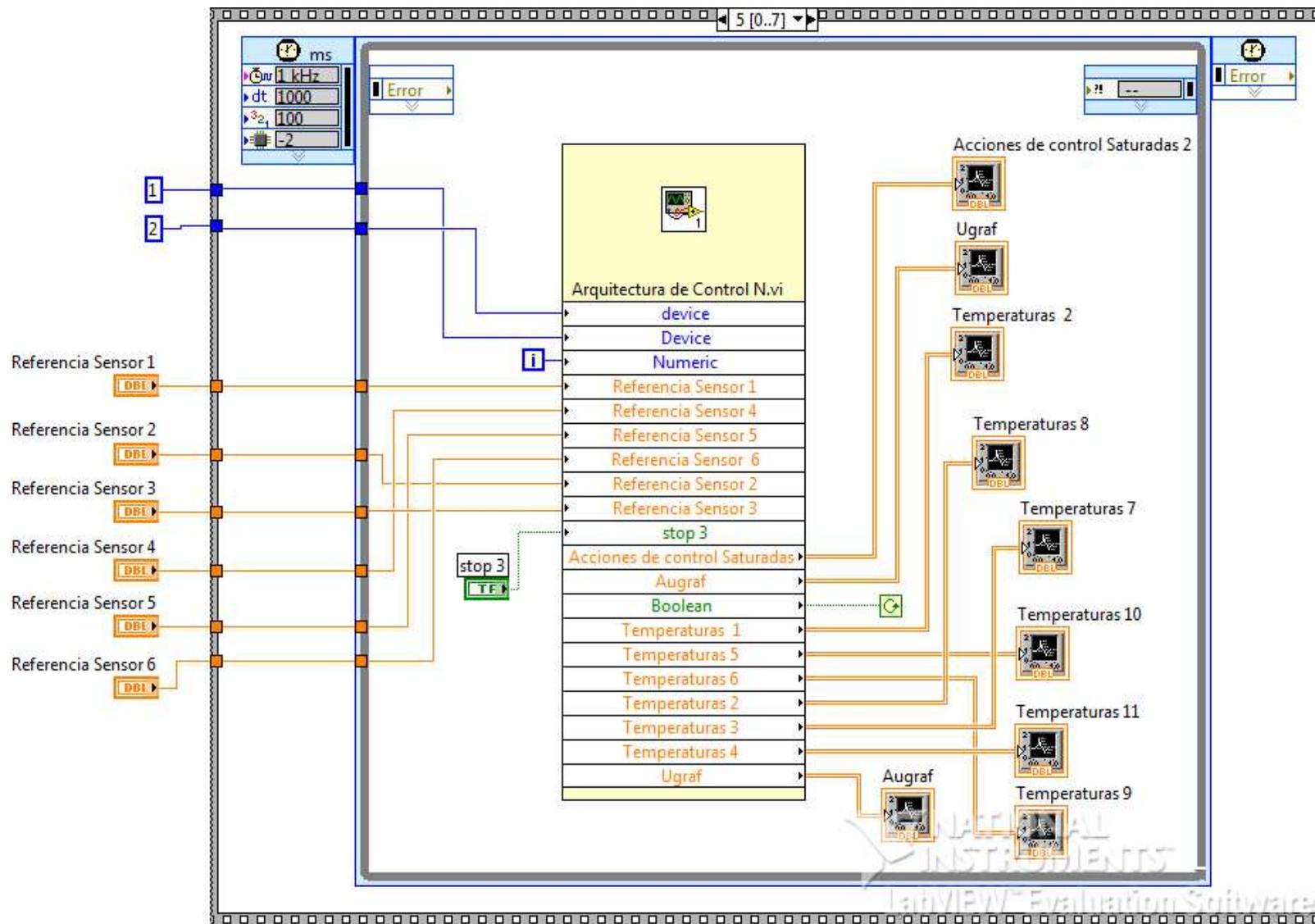


Figura 8.13: Implementación del código de control para controlar el Modelo nominal, ver código en anexo C.6.5.

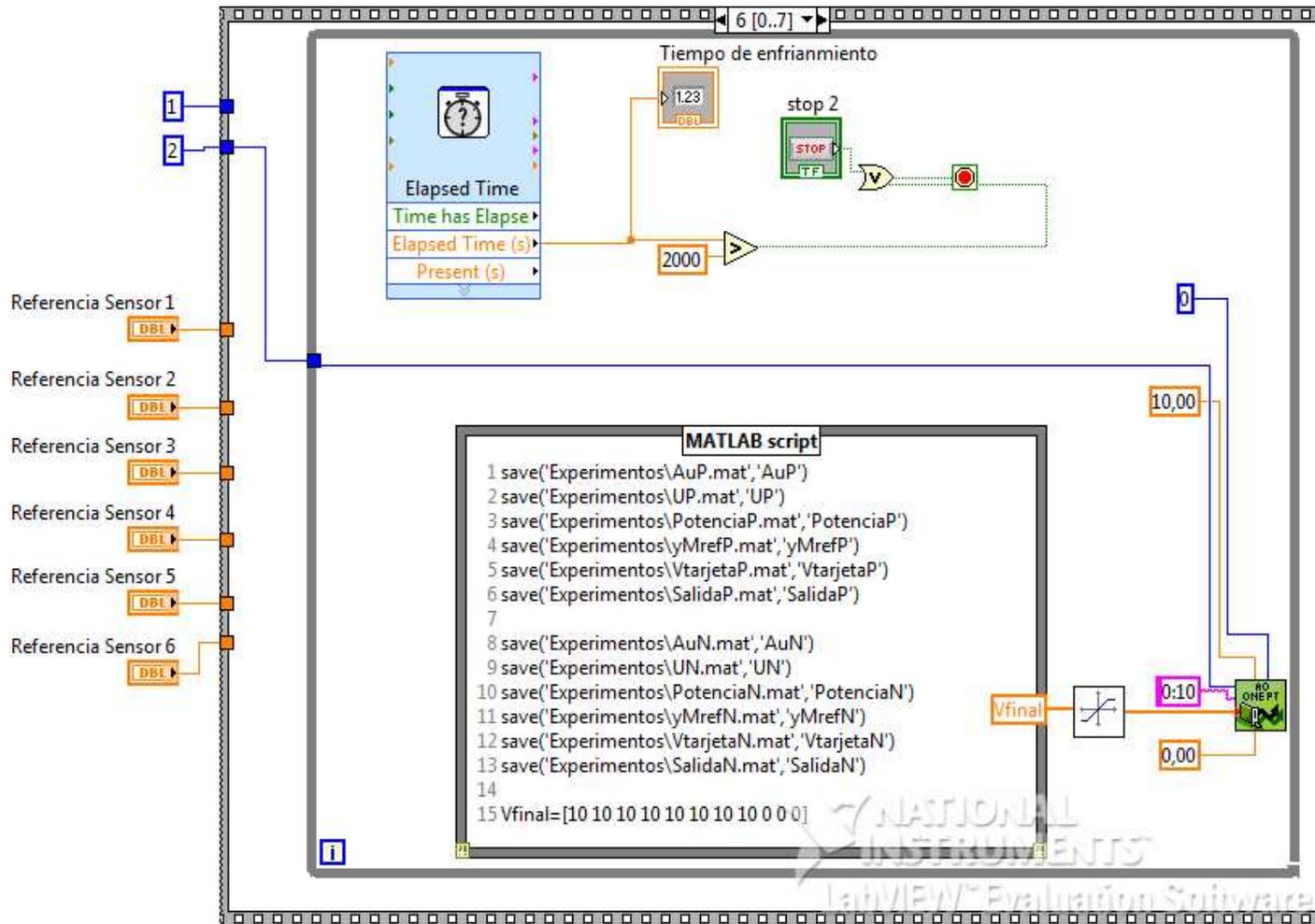


Figura 8.14: Salvamos los datos en .mat para que puedan ser leídos por matlab y refrescamos el horno apagando los calentadores y encendiendo al máximo los ventiladores.

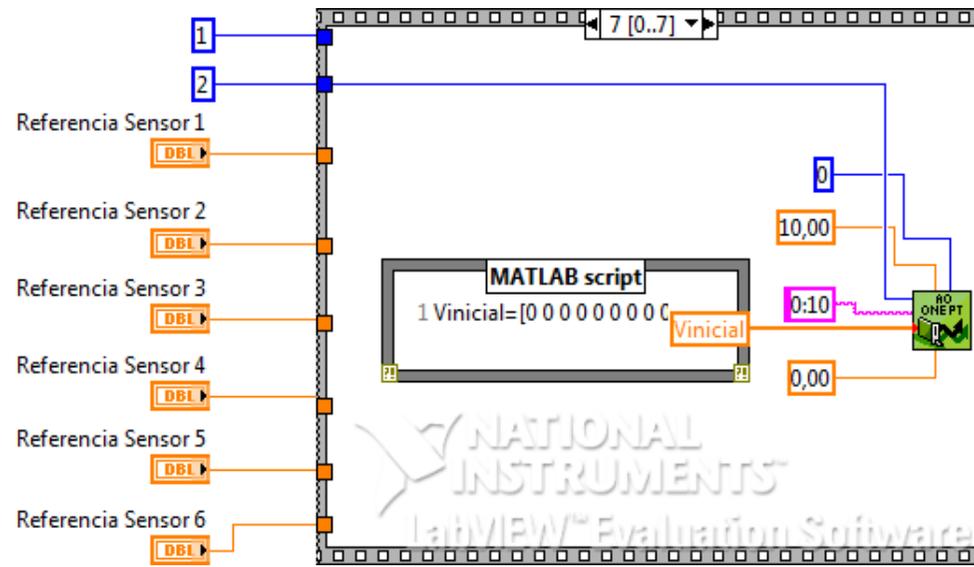


Figura 8.15: Se para todo

## Capítulo 9

# Conclusiones

El presente trabajo muestra un prototipo en desarrollo de horno continuo para experimentos de control multivariable. A pesar de todos los problemas de diseño, se han realizado ensayos de identificación y control en bucle cerrado (dos grados de libertad, LQG, antiwindup). Además se ha generado las aplicaciones necesarias para generación de modelos e implementación de técnicas de control multivariable. También se dispone de un sistema de Scadas para realización de experimentos.

Destacar que al ser una planta en desarrollo, las características óptimas de la misma aún no están definitivamente prefijadas, pero creemos que puede ser un equipo razonable para uso educativo. Los siguientes pasos están orientados a mejorar las configuraciones estudiadas.

Por otro lado subrayar que las limitaciones impuestas por la cinta transportadora y por los sensores inalámbricos han hecho que no se utilicen, de momento, aunque se ha comprobado que ambos sistemas funcionan correctamente. De hecho los sensores inalámbricos, basados en Zigbee, han sido programados.



## **Parte IV**

# **Apéndices**



# Apéndice A

## Conceptos útiles

### A.1. Definiciones

- **Precisión:** Es el intervalo donde es admisible que se sitúe la magnitud medida, y define los límites de los errores cometidos cuando el instrumento se emplea en condiciones normales de servicio durante un período de tiempo determinado, [11].
- **Exactitud:** Es la cualidad de un instrumento de medida por la que tiende a dar lecturas próximas al verdadero valor de la magnitud medida.
- **Precisión:** Es la tolerancia de medida o de transmisión del instrumento (intervalo donde es admisible que se sitúe la magnitud de la medida), y define los límites de los errores cometido cuando el instrumento se emplea en condiciones normales de servicio durante un periodo de tiempo determinado. La precisión varía en cada punto del campo de medida. Por ejemplo un instrumento puede tener una precisión de  $\pm 1\%$  en toda la escala y de  $\pm 0,5\%$  en la zona central.
- **Sensibilidad:** Es la razón entre el incremento de la lectura y el incremento de la variable que lo ocasiona, después de haberse alcanzado el estado de reposo.
- **Repetitividad:** Es la capacidad de reproducción de la señal de salida del instrumento al medir repetidamente valores idénticos de la variable a medir en las mismas condiciones de servicio y en el mismo sentido de variación, recorriendo todo el campo. Para determinarla se comprueba la diferencia entre el valor verdadero y la señal del instrumento, recorriendo todo el campo y partiendo para cada determinación desde el valor mínimo del campo de medida. Se considera en general su valor máximo y se expresa en tanto por ciento del alcance.

Nótese que este término no incluye histéresis. Expresión:

$$\sqrt{\frac{\sum (x_i - x)^2}{N}} \quad (\text{A.1.1})$$

- Histéresis: Es la diferencia máxima que se observa en los valores indicado por el instrumento para el mismo valor cualquiera del campo de medida, cuando la variable se recorre en los dos sentidos. Se expresa en tanto por ciento del alcance de la mediada:

$$\text{Histeresis\_en\_la\_Medida}(\%) = \frac{|Medida_{Subida} - Medida_{Bajada}|}{ValorMaximo_{FE} - ValorMinimo_{FE}} \cdot 100 \quad (\text{A.1.2})$$

- Alcance (SPAN): Es la diferencia algebraica entre los valores superior e inferior del campo de medida del instrumento.
- Deriva: Es una variación en la señal de salida que se presenta en un período de tiempo determinado mientras se mantiene constante la variable medid y todas las condiciones ambientales.
- Resolución: Magnitud de los cambios en escalón de la señal de salida (expresado en tanto por ciento de la salida de toda la escala) al ir variando contiunamente la medida en todo el campo. Es también el grado con que el instrumento puede discriminar valores equivalentes de una cantidad, o la menor diferencia de valor que el aparato puede distinguir.
- Ruido: Cualquier perturbación eléctrica o señal accidental no deseada que modifica la transmisión, indicación o registro de los datos deseados.
- Termopar: Un termopar es un circuito formado por dos metales distintos que produce un voltaje que es función de la diferencia de temperatura entre uno de los extremos denominado “unión caliente” y el otro denominado “unión fría”. Lás características principales se muestran en la tabla A.1.

PROS	CONTRA
Bajo costo.	La sensibilidad es baja, generalmente $50\mu\text{V}/^\circ\text{C}$ o menos
No hay piezas móviles, menos probabilidad de romperse.	Generalmente la exactitud no es menor a $0.5^\circ\text{C}/1^\circ\text{C}$
Amplia gama de temperaturas	Requiere una temperatura de referencia
Tiempo de reacción razonablemente corto	Muy sensible a ruido
Capacidad de repetición y exactitud razonables	

Tabla A.1: Ventajas y desventajas principales de los termopares

- Transmisor: Elemento que capta la variable del proceso a través del elemento primario y la transmite en forma de señal.
- Bit de resolución: Número de bits que el convertidor analógico a digital (ADC) utiliza para representar una señal.
- Teorema de Nyquist: Al muestrear una señal, la frecuencia de muestreo debe ser mayor que dos veces el ancho de banda de la señal de entrada, para poder reconstruir la señal original de forma exacta a partir de sus muestras.

## A.2. Acrónimos

- CJ: Cold Junction. Unión fría del termopar que considera la temperatura ambiente.
- PWM: La modulación por ancho de pulsos (MAP o PWM, siglas del inglés Pulse-Width Modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica.
- DAQ: Data acquisition. Sistema de adquisición de datos que toma muestras del mundo real (sistema analógico) para generar datos que puedan ser manipulados por un ordenador u otras electrónicas (sistema digital).
- LTI: (Linear Time-Invariant) o sistema lineal e invariante en el tiempo, es aquel que, como su propio nombre indica, cumple las propiedades de linealidad e invarianza en el tiempo.
- SISO: (Single Input Single Output). Sistemas de una sola entrada y una sola salida.
- MIMO (Múltiple-Input-Multiple-Output): Sistemas con múltiple entrada y múltiple salida.

## A.3. Tipos de ruido que podemos encontrar

- a) Ruido errático: Incluye el ruido atmosférico y el ruido espacial, que es consecuencia entre otras causas de la ionización y recombinación de moléculas gaseosas por acción de la radiación solar, cósmica, campos eléctricos intensos, etc. Afecta principalmente las transmisiones inalámbricas,[22].
- b) Ruido producido por el hombre: Comprende la radiación electromagnética emitida por artefactos empleados por el hombre. En general se origina en conmutaciones, chispas o emisión voluntaria o involuntaria de radiofrecuencia,[22]. También incluye las perturbaciones ocasiona-

das por la modificación de la carga en sistemas de alimentación y por filtrado insuficiente en las fuentes de corriente continua que rectifican una corriente alterna. Este último es el clásico zumbido a la frecuencia de línea en los amplificadores de audio,[22].

- c) Ruido circuital: Es el ruido introducido por los propios elementos del circuito y se debe a los fenómenos físicos que tienen lugar en ellos. Por ejemplo la agitación térmica de los electrones en las resistencias (que da origen al ruido térmico),[22].

## Apéndice B

# Instrucciones DAQ-MTLB

### Ejemplos de instrucciones

Obtener información:

```
hwinfo = daqhwinfo
adqpters = hwinfo.InstalledAdaptors
propinfo(ai, 'SampleRate')
```

Capturar una señal:

```
ai = analoginput('mwadlink', 0) %Abrir entrada analógica del dispositivo 0.
ch1= addchannel(ai, 0) %Añadir canal #0 al objeto ai.
ai_data = getsample(ai) %Toma de datos.
```

Capturar una señal continua:

```
set(ai, 'SampleRate', 1000) %Frecuencia de muestreo.
set(ai, 'SamplesPerTrigger', 1000); %Número de muestras.
start(ai) %Empieza la adquisición de datos.
wait(ai, Td)
ai_data = getdata(ai);
```

Escritura de señales:

```
ao = AnalogOutput( 'mwadlink', 1) %Abre salida analogical del dispositivo 1.
```

```
ch1= addchannel(ao, 0) %Añadir canal #0 a ao.  
putsample(ao_device, 3) %Enviar una muestra Output 3V en el canal #0.
```

**Manejo del Buffer:**

```
out=daqmem(ai); %Devuelve Memoria utilizada y Memoria Máxima.  
daqmem(ai,maxmem); %Con maxmem indicamos la memoria Máxima a utilizar.  
ai.BufferingMode %"Auto"/"Manual"  
ai.BufferingConfig [n m]
```

## Apéndice C

# Programas

### C.1. Experimentos de adquisición de datos

#### C.1.1. Bucle de Adquisición

```
% Inicializacion
2 clear all; close all ;clc
  hwinfo = daqhwinfo;
4 adqpters = hwinfo.InstalledAdaptors;
  ADLINK_INFO = daqhwinfo('mwadlink')
6
  inicializada_PCI9111=-1;
8 inicializada_PCI6208=-1;

10 if (inicializada_PCI9111== -1)
    [ai_device , error_PCI9111]=Inicializar_ADLINK_PCI9111 ();
12     if (error_PCI9111==0)
        inicializada_PCI9111=0;
14     else
        delete (ai_device);
16     clear ai_device;
    end
18 end

20 if (inicializada_PCI6208== -1)
    [ao_device , error_PCI6208]=Inicializar_ADLINK_PCI6208 ();
22     if (error_PCI6208==0)
        putsample(ao_device , [0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0]);
24     inicializada_PCI6208=0;
```

```

    else
26     delete (ao_device);
        clear ao_device;
28     end
end
30 %Experimento
Texp=5000.0; %Tiempo Experimento
32 tmuestreo=0.1; %Frecuencia de muestreo 10Hz
tini=clock;
34 for i=1:1:Texp/tmuestreo;
    t(i,:) = clock;
36     if (i<(0.5*Texp/tmuestreo))
        accion(i,:)=[0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 5.0 5.0 5.0];
38     else
        accion(i,:)=[0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 5.0 7.0 5.0];
40     end
    wait(ai_device,0.6*tmuestreo);
42     input_rad(i,:)=getsample(ai_device);
    putsample(ao_device, accion(i,:)); %Envio tension a los actuadores
44     while etime(clock,t(i,:))<tmuestreo end
end
46 %Finalizo experimento
tfin=clock;
48 tttotal=etime(tfin,tini); %Tiempo total del experimento
vectiempo=etime(t,ones(length(t),1)*t(1,:));
50 putsample(ao_device, [10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 0.0 0.0 0.0]);
putsample(ao_device, [0.0 0 0 0 0.0 0.0 0 0.0 0.0 0.0 0.0]);
52 while etime(clock,tfin)<10 end

54 %Borrar objetos y liberar memoria.
clear ai_channels;
56 delete (ai_device);
clear ai_device;
58 clear ao_channels;
delete (ao_device);
60 clear ao_device;
%Guardo el experimento
62 %Dibujo resultados

```

### C.1.2. Función para inicializar PCI9111

```

function [ai_device , error_PCI9111]=Inicializar_ADLINK_PCI9111 ();
2 PCI9111=0;

```

```
try
4  %Programo características tarjeta PCI9111 (sensores)
  ai_device = analoginput('mwdlink', PCI9111);
6  Set(ai_device, 'InputType', 'SingleEnded');
  Set(ai_device, 'SampleRate', 1000);
8  Set(ai_device, 'TriggerType', 'Immediate');
  Set(ai_device, 'TriggerDelay', 0);
10 Set(ai_device, 'SamplesPerTrigger', 100);
  %Canales de lectura
12   s1p = addchannel(ai_device, 0);
      set(s1p, 'InputRange', [-10 10])
14   s1n = addchannel(ai_device, 1);
      set(s1n, 'InputRange', [-10 10])
16   s2p = addchannel(ai_device, 2);
      set(s2p, 'InputRange', [-10 10])
18   s2n = addchannel(ai_device, 3);
      set(s2n, 'InputRange', [-10 10])
20   s3p = addchannel(ai_device, 4);
      set(s3p, 'InputRange', [-10 10])
22   s3n = addchannel(ai_device, 5);
      set(s3n, 'InputRange', [-10 10])
24   s4p = addchannel(ai_device, 6);
      set(s4p, 'InputRange', [-10 10])
26   s4n = addchannel(ai_device, 7);
      set(s4n, 'InputRange', [-10 10])
28   s5p = addchannel(ai_device, 8);
      set(s5p, 'InputRange', [-10 10])
30   s5n = addchannel(ai_device, 9);
      set(s5n, 'InputRange', [-10 10])
32   s6p = addchannel(ai_device, 10);
      set(s6p, 'InputRange', [-10 10])
34   s6n = addchannel(ai_device, 11);
      set(s6n, 'InputRange', [-10 10])
36 %
      error_PCI9111=0;
38 catch
  %Mensaje en caso de error al inicializar.
40   display('System error PCI9111!')
      error_PCI9111=-1;
42   return
end
```

### C.1.3. Función para inicializar PCI6208(6216)

```

1 function [ao_device , error_PCI6208 ]=Inicializar_ADLINK_PCI6208 ();
  PCI6208=1;
3 try
  %Programo características tarjeta PCI6208 (actuadores)
5 ao_device = analogoutput( 'mwadlink', PCI6208);
  set(ao_device, 'AD_DeviceSubType', '6216V')
7 %Para acutar sobre radiadores y ventiladores
  v1_tunel = addchannel(ao_device, 0);
9     set(v1_tunel, 'OutputRange',[-10 10])
  v2_tunel = addchannel(ao_device, 1);
11    set(v2_tunel, 'OutputRange',[-10 10])
  v3_tunel= addchannel(ao_device, 2);
13    set(v3_tunel, 'OutputRange',[-10 10])
  v1_techo = addchannel(ao_device, 3);
15    set(v1_techo, 'OutputRange',[-10 10])
  v4_tunel = addchannel(ao_device, 4);
17    set(v4_tunel, 'OutputRange',[-10 10])
  v5_tunel= addchannel(ao_device, 5);
19    set(v5_tunel, 'OutputRange',[-10 10])
  v6_tunel= addchannel(ao_device, 6);
21    set(v6_tunel, 'OutputRange',[-10 10])
  v2_techo= addchannel(ao_device, 7);
23    set(v2_techo, 'OutputRange',[-10 10])
  radiador1 = addchannel(ao_device, 8);
25    set(radiador1, 'OutputRange',[-10 10])
  radiador2 = addchannel(ao_device, 9);
27    set(radiador2, 'OutputRange',[-10 10])
  radiador3= addchannel(ao_device, 10);
29    set(radiador3, 'OutputRange',[-10 10])
  error_PCI6208=0;
31 catch
  %Mensaje en caso de error al inicializar.
33     display('System error PCI6208!')
  error_PCI6208=-1;
35     return
end

```

## C.2. Código de Identificación antes del rediseño

```

%Cargo los datos del experimento ....

```

```

2  %Cargar datos: Pasar voltios salida sensores a Temperatura
   load('P','P');
4  load('S','S');
   load('MU','MU');
6  %Cargar datos Pasar voltios entrada a Watios en los radiadores
   load('Pw','Pw');
8  load('Sw','Sw');
   load('MUw','MUw');
10 %Selector del Radiador 2 como actuador pin:10
    R=10;
12 %Numero de muestras
    t=vectiempo.vectiempo;
14 muestras=length(t);
    %Tiempo trascurrido
16 Temps=tmuestreo . tmuestreo : tmuestreo . tmuestreo : tmuestreo . tmuestreo *muestras;
    %Frecuencia de muestreo
18 Fs=inv(tttotal . tttotal /length(lectura_sensores . input_rad));
    %Paso de voltios a grados
20 T=V2T(V,P,S,MU);
    %Preprocesado de los datos
22 Fc=Fs/800; Orden=5;
    [b,a] = butter(Orden,Fc/Fs);
24 Tf = filtfilt(b, a, T); %Filtrado
    %Obtenemos Tension entrada a los Radiadores
26 u=referencia . accion(:,R);
    %Pasar a Watios
28 AW=Vdaq2W(Ufinal, Uinicial, Pw, Sw, MUw)
    %Identificacion
30 [K, tf632, y0, yinf, u0, uinf, v20, v2inf]=
    id_stepositivo(Tf, tmuestreo, muestras, tttotal, u, v2);
32
    [retardo, tau, handle0]=
34 calculo_retardo3(Tf, muestras, tmuestreo, tttotal, y0, yinf, tf632);
    s=tf('s');
36 G1=[K(1)/(tau(1)*s+1) K(2)/(tau(2)*s+1) K(3)/(tau(3)*s+1)
    K(4)/(tau(4)*s+1) K(5)/(tau(5)*s+1) K(6)/(tau(6)*s+1)];
38 G1.ioDelayMatrix=[retardo];
    %Simulacion ....
40 %Guardo grafica obtenidas y datos ...

```

### C.2.1. Función para identificar escalón positivo y tao ideal

```
function [K, tf632, y0, yinf, u0, uinf, v20, v2inf]=
```

```

2   id_steppositivo(T,tmuestreo ,muestras ,ttotal ,u,v2)
t632=[];
4   yinf=mean(T(muestras-1000:muestras,:));
y0=mean(T(0.5*muestras-1000:0.5*muestras,:));
6   %
uinf=mean(u(muestras-1000:muestras));
8   u0=mean(u(0.5*muestras-1000:0.5*muestras));
%
10  v2inf=mean(v2(muestras-1000:muestras));
v20=mean(v2(0.5*muestras-1000:0.5*muestras));
12  %Gananacia
K=(yinf-y0)/(v2inf-v20);
14  %Tao ideal
y632=y0+0.632*(yinf-y0);
16  %
i=0.5*muestras;
18  while mean(T(i-100:i+100,1))<y632(1)
t632(1)=i*tmuestreo .tmuestreo; i=i+1;
20  end
i=0.5*muestras;
22  while mean(T(i-100:i+100,2))<y632(2)
t632(2)=i*tmuestreo .tmuestreo; i=i+1;
24  end
i=0.5*muestras;
26  while mean(T(i-100:i+100,3))<y632(3)
t632(3)=i*tmuestreo .tmuestreo; i=i+1;
28  end
i=0.5*muestras
30 ;while mean(T(i-100:i+100,4))<y632(4)
t632(4)=i*tmuestreo .tmuestreo; i=i+1;
32  end
i=0.5*muestras;
34  while mean(T(i-100:i+100,5))<y632(5)
t632(5)=i*tmuestreo .tmuestreo; i=i+1;
36  end
i=0.5*muestras;
38  while mean(T(i-100:i+100,6))<y632(6)
t632(6)=i*tmuestreo .tmuestreo; i=i+1;
40  end
42  tf632=(t632-ttotal .ttotal/2);

```

## C.2.2. Función cálculo del retardo y de tao real

```

function [retardo ,tau ,handle0]=
2   calculo_retardo (T,muestras ,tmuestreo ,ttotal ,y0 ,yinf ,tf632)

4   t284=[];
   y284=y0+0.284*(yinf-y0);
6   tm=tmuestreo . tmuestreo ;
   i=0.5*muestras;
8   while mean(T(i-100:i+100,1))<y284 (1)
   t284 (1)=i*tm; i=i+1;
10  end
   i=0.5*muestras;
12  while mean(T(i-100:i+100,2))<y284 (2)
   t284 (2)=i*tm; i=i+1;
14  end
   i=0.5*muestras;
16  while mean(T(i-100:i+100,3))<y284 (3)
   t284 (3)=i*tm; i=i+1;
18  end
   i=0.5*muestras;
20  while mean(T(i-100:i+100,4))<y284 (4)
   t284 (4)=i*tm; i=i+1;
22  end
   i=0.5*muestras;
24  while mean(T(i-100:i+100,5))<y284 (5)
   t284 (5)=i*tm;
26  i=i+1;end
   i=0.5*muestras;
28  while mean(T(i-100:i+100,6))<y284 (6)
   t284 (6)=i*tm;
30  i=i+1;end

32  %Calculo de tau real
   t0=ttotal . tttotal /2;
34  tempfinal=yinf-y0;
   tau=1.5*(tf632 -(t284-t0));
36

   %Calculo del retardo
38  b=(y284-y0)-((tempfinal ./tau) .*(t284-t0));
   test=(tempfinal-b) .*(tau ./ tempfinal);
40  ret=test-tau;
   if ret (1)<0 ret (1)=0;end
42  if ret (2)<0 ret (2)=0;end
   if ret (3)<0 ret (3)=0;end
44  if ret (4)<0 ret (4)=0;end

```

```

46 if ret(5)<0 ret(5)=0;end
   if ret(6)<0 ret(6)=0;end
   retardo=ret;

```

### C.2.3. Función para pasar de Voltios a Grados

```

1 function T=V2T(V,P,S,MU)
  T1 = POLYVAL(P(1,:),V(:,1),S(1,:),MU(1,:));
3  T2 = POLYVAL(P(2,:),V(:,2),S(2,:),MU(2,:));
  T3 = POLYVAL(P(3,:),V(:,3),S(3,:),MU(3,:));
5  T4 = POLYVAL(P(4,:),V(:,4),S(4,:),MU(4,:));
  T5 = POLYVAL(P(5,:),V(:,5),S(5,:),MU(5,:));
7  T6 = POLYVAL(P(6,:),V(:,6),S(6,:),MU(6,:));
  T=[T1,T2,T3,T4,T5,T6];

```

### C.2.4. Función para pasar de Voltios a Watios

```

function AW=Vdaq2W(Ufinal,Uinicial,Pw,Sw,MUw)
2 Wfinal=POLYVAL(Pw,Ufinal,Sw,MUw);
  Winicial=POLYVAL(Pw,Uinicial,Sw,MUw);
4 AW=Wfinal-Winicial;

```

## C.3. Código de Identificación despues del rediseño

Ejemplo para identificación del modelo  $G1(s)$  (sería igual para el modelo  $G2(s)$ )

```

clear all
2 close all
  clc
4 %Cargar datos: Pasar voltios salida sensores a Temperatura
  load('P','P');
6  load('S','S');
  load('MU','MU');
8 %Cargar datos Pasar voltios entrada a Watios en los radiadores

```

```

load( 'Pw', 'Pw' );
10 load( 'Sw', 'Sw' );
load( 'MUw', 'MUw' );
12
%RADIADOR 1
14 % Salidas
Ys_R1=load( 'ExpMod1\Radiador1_2c4_to_5_Ventiladors_6.mat' );
16 salida=V2T(Ys_R1.input(:,1:6),P,S,MU);
puntofun1_sensores=salida(1,1:6);
18 inc_salida=[salida(:,1)-puntofun1_sensores(1) salida(:,2)-puntofun1_sensores(2)
salida(:,3)-puntofun1_sensores(3) salida(:,4)-puntofun1_sensores(4)
20 salida(:,5)-puntofun1_sensores(5) salida(:,6)-puntofun1_sensores(6)];
%Entrada
22 Ufinal=4*ones(max(length(salida)),1);
Uinicial=2.4*ones(max(length(salida)),1);
24 Escalon=Vdaq2W(Ufinal,Uinicial,Pw(1,:),Sw(1,:),MUw(1:2,:));
%
26 T=1:1:max(length(salida));
datas1=iddata(inc_salida(:,1),Escalon);
28 G11r= pem(datas1,'P1D');
datas12=iddata([inc_salida(:,2)],[Escalon]);
30 G12r= pem(datas12,'P2');
datas13=iddata([inc_salida(:,3)],[Escalon]);
32 G13r= pem(datas13,'P1D');
datas14=iddata([inc_salida(:,4)],[Escalon]);
34 G14r= pem(datas14,'P1Z');
datas15=iddata([inc_salida(:,5)],[Escalon]);
36 G15r= pem(datas15,'P2');
datas16=iddata([inc_salida(:,6)],[Escalon]);
38 G16r= pem(datas16,'P1D');

40 G1=[tf(G11r);tf(G12r);tf(G13r);tf(G14r);tf(G15r);tf(G16r)];
%RADIADOR 2
42 % Salidas
Ys_R2=load( 'ExpMod1\Radiador2_2c4_to_5_Ventiladors_6.mat' );
44 salida=V2T(Ys_R2.input(:,1:6),P,S,MU);
puntofun2_sensores=salida(1,1:6);
46 inc_salida=[salida(:,1)-puntofun2_sensores(1) salida(:,2)-puntofun2_sensores(2)
salida(:,3)-puntofun2_sensores(3) salida(:,4)-puntofun2_sensores(4)
48 salida(:,5)-puntofun2_sensores(5) salida(:,6)-puntofun2_sensores(6)];
%Entrada
50 Ufinal=4*ones(max(length(salida)),1);
Uinicial=2.4*ones(max(length(salida)),1);
52 Escalon=Vdaq2W(Ufinal,Uinicial,Pw(2,:),Sw(2,:),MUw(3:4,:));
%
54 T=1:1:max(length(salida));

```

```

datas21=iddata ([ inc_salida (: , 1) ], [ Escalon ] );
56 G21r= pem (datas21 , 'P1D' );
datas22=iddata ([ inc_salida (: , 2) ], [ Escalon ] );
58 G22r= pem (datas22 , 'P2' );
datas23=iddata ([ inc_salida (: , 3) ], [ Escalon ] );
60 G23r= pem (datas23 , 'P1D' );
datas24=iddata ([ inc_salida (: , 4) ], [ Escalon ] );
62 G24r= pem (datas24 , 'P2' );
datas25=iddata ([ inc_salida (: , 5) ], [ Escalon ] );
64 G25r= pem (datas25 , 'P2' );
datas26=iddata ([ inc_salida (: , 6) ], [ Escalon ] );
66 G26R= pem (datas26 , 'P2' );
G2=[ tf (G21r) ; tf (G22r) ; tf (G23r) ; tf (G24r) ; tf (G25r) ; tf (G26r) ];
68
%RADIADOR 3
70 % Salidas
Ys_R3=load ( 'ExpMod1\Radiador3_2c4_to_5_Ventiladors_6.mat' );
72 salida=V2T (Ys_R3 .input (: , 1:6) ,P,S,MU);
puntofun3_sensores=salida (1 , 1:6);
74 inc_salida=[salida (: , 1)-puntofun3_sensores (1) salida (: , 2)-puntofun3_sensores (2)
salida (: , 3)-puntofun3_sensores (3) salida (: , 4)-puntofun3_sensores (4)
76 salida (: , 5)-puntofun3_sensores (5) salida (: , 6)-puntofun3_sensores (6) ];
% Entrada
78 Ufinal=4*ones (max (length (salida) ) , 1);
Uinicial=2.4*ones (max (length (salida) ) , 1);
80 Escalon=Vdaq2W (Ufinal , Uinicial , Pw (3 , :) , Sw (3 , :) , MUw (5:6 , :) );
%
82 T=1:1:max (length (salida) );
datas31=iddata (inc_salida (: , 1) , Escalon);
84 G31r=pem (datas31 , 'P2' );
datas32=iddata ([ inc_salida (: , 2) ], [ Escalon ] );
86 G32r=pem (datas32 , 'P2' );
datas33=iddata ([ inc_salida (: , 3) ], [ Escalon ] );
88 G33r=pem (datas33 , 'P1D' );
datas34=iddata ([ inc_salida (: , 4) ], [ Escalon ] );
90 G34r=pem (datas34 , 'P2' );
datas35=iddata ([ inc_salida (: , 5) ], [ Escalon ] );
92 G35r=pem (datas35 , 'P1D' );
datas36=iddata ([ inc_salida (: , 6) ], [ Escalon ] );
94 G36=pem (datas36 , 'P1Z' );

96 G3=[ tf (G31r) ; tf (G32r) ; tf (G33r) ; tf (G34r) ; tf (G35r) ; tf (G36r) ];
Modelo_continuo1=[G1 G2 G3];
98 % Simulacion ....
%Guardo grafica obtenidas y datos ....
100 save ( 'ExpMod1 \Modelo_continuo1 ' , 'Modelo_continuo1' );

```

## C.4. Diseño de Realimentación

### C.4.1. Ejemplo para proceso 3x3

```

load('Modelo_continuo1');
2 %Modelo con solo tres salidas:2,4,6
Process=(Modelo_continuo1(2:2:6,1:2:5));
4 %Espacio de estados
ProcessSS=ss(Process);
6 %Reducción del modelo
modelo = balred(ProcessSS,9);
8
%NxPxM del sistema
10 actuadores=size(modelo.b,2);
estados=size(modelo.b,1);
12 salidas=size(modelo.c,1);

14 %Sistema ampliado con integradores en
%la entrada para eliminar el error en regimen permanente
16 Aam1=[modelo.a modelo.b; zeros(actuadores,estados)
zeros(actuadores)-1e-16*eye(actuadores)];
18 Bam1=[modelo.b; zeros(actuadores)];
Cam1=[modelo.c zeros(salidas,actuadores)];
20 Dam1=[modelo.d];
modelo_ampliado1=ss(Aam1,Bam1,Cam1,Dam1);
22
%OBSERVADORES
24 %COVARIANZAS
g=1;h=0;
26 q_obs=1;
r_obs=1;
28 %OBSERVADOR
BG=[modelo.b g*eye(estados)]; %B G]
30 DH=[modelo.d h*eye(salidas,estados)]; %D H]
SYS=ss(modelo.a,BG,modelo.c,DH);
32 [Kkalman3,Lkalman,Pkalman] =
kalman(SYS,q_obs*eye(estados),r_obs*eye(salidas));
34 %OBSERVADOR CON SISTEMA AMPLIADO 1
BGam1=[Bam1 g*eye(estados+actuadores)]; %B G]
36 DHam1=[Dam1 h*eye(salidas,estados+actuadores)]; %D H]

```

```

38     SYSam1=ss (Aam1,BGam1,Cam1,DHam1);
    [Kkalmanam13,Lkalmanam1,Pkalmanam1] =
        kalman(SYSam1,q_obs*eye(estados+actuadores),r_obs*eye(salidas));
40
41 %REGULADORES
42     %PESOS
    q=[1 1 1];r=[0.0001 0.01 0.0001];
44     Qy=diag(q);
    R=diag(r);
46 %REGULADOR
    [Kopt3,Sopt,Eopt]=lqry(modelo.a,modelo.b,modelo.c,modelo.d,Qy,R);
48 %REGULADOR CON SISTEMA AMPLIADO 1
    [Koptam13,Soptam1,Eoptam1] =
50     lqry(modelo_ampliado1.a,modelo_ampliado1.b,
        modelo_ampliado1.c,modelo_ampliado1.d,Qy,R);
52
53 %CONECTO EN SERIE OBSERVADOR Y REGULADOR
54     output01=salidas+1:salidas+estados;
    input02=1:estados;
56     reg03=series(Kkalman3,ss(-Kopt3),output01,input02);
57 %CONECTO EN SERIE OBSERVADOR Y REGULADOR AMPLIADOS
58     output11=salidas+1:salidas+estados+actuadores;
    input12=1:estados+actuadores;
60     reg13=series(Kkalmanam13,ss(-Koptam13),output11,input12);
62 % Simulo los bucles en simulink...
    % Dibujo los datos...
64 % Salvo los datos, reg03, reg13, ...

```

#### C.4.2. Ejemplo para proceso 6x8

```

2 %PROCESO CONTINUO REDUCIDO
    modelo = balred(ProcessSS,15);
4 %MxPxM del sistema
    actuadores=size(modelo.b,2);
6     estados=size(modelo.b,1);
    salidas=size(modelo.c,1);
8
9 %SISTEMA AMPLIADO 1 PARA ELIMINAR ERRORES EN REGIMEN ESTACIONARIO
10 Aam1=[modelo.a modelo.b; zeros(actuadores,estados) zeros(actuadores)-1e-16*eye(
    actuadores)];
    Bam1=[modelo.b; zeros(actuadores)];

```

```

12 Cam1=[modelo.c zeros(salidas,actuadores)];
   Dam1=[modelo.d];
14 modelo_ampliado1=ss(Aam1,Bam1,Cam1,Dam1);

16 %OBSERVADORES
   %COVARIANZAS
18 g=1;h=0;
   q_obs=0.8;
20 r_obs=2;
   %OBSERVADOR
22 BG=[modelo.b g*eye(estados)]; %B G]
   DH=[modelo.d h*eye(salidas,estados)]; %D H]
24 SYS=ss(modelo.a,BG,modelo.c,DH);
   [Kkalman6,Lkalman,Pkalman] = kalman(SYS,q_obs*eye(estados),r_obs*eye(salidas));
26 %OBSERVADOR SISTEMA AMPLIADO 1
   BGam1=[Bam1 g*eye(estados+actuadores)]; %B G]
28 DHam1=[Dam1 h*eye(salidas,estados+actuadores)]; %D H]
   SYSam1=ss(Aam1,BGam1,Cam1,DHam1);
30 [Kkalmanam16,Lkalmanam1,Pkalmanam1] = kalman(SYSam1,q_obs*eye(estados+actuadores),r_obs
   *eye(salidas));

32 %REGULADOR con PROCESO CONTINUO REDUCIDO
   [Kopt6,Sopt,Eopt]=lqry(modelo.a,modelo.b,modelo.c,modelo.d,Qy,R,N);
34
   %REGULADOR con SISTEMA AMPLIADO 1
36 [Koptam16,Soptam1,Eoptam1]=lqry(modelo_ampliado1.a,modelo_ampliado1.b,modelo_ampliado1.
   c,modelo_ampliado1.d,Qy,R,N);

38 %CONECTO EN SERIE OBSERVADOR Y REGULADOR
   output01=salidas+1:salidas+estados;
40 input02=1:estados;
   reg06=series(Kkalman6,ss(-Kopt6),output01,input02);
42
   %CONECTO EN SERIE OBSERVADOR Y REGULADOR AMPLIADO
44 output11=salidas+1:salidas+estados+actuadores;
   input12=1:estados+actuadores;
46 reg16=series(Kkalmanam16,ss(-Koptam16),output11,input12);

48 %Simulo los bucles en simulink...
   %Dibujo los datos...
50 %Salvo los datos, reg06, reg16, ...

```

## C.5. Implementación de Bucle de Control

```

% Cargar los datos necesarios...
2 % Toma de punto de funcionamiento inicial...
% Crear estructura de datos necesaria...
4 % -----
% Pasar a Temperatura
6 T1 = polyval(P(1,:),V(1),S(1,:),MU(1,:));
T2 = polyval(P(2,:),V(2),S(2,:),MU(2,:));
8 T3 = polyval(P(3,:),V(3),S(3,:),MU(3,:));
T4 = polyval(P(4,:),V(4),S(4,:),MU(4,:));
10 T5 = polyval(P(5,:),V(5),S(5,:),MU(5,:));
T6 = polyval(P(6,:),V(6),S(6,:),MU(6,:));
12 T=[T1; T2; T3; T4; T5; T6];

14 Refer_P=ref-TOP;
% Salida MrP (Modelo de Referencia)
16 x1P(:,i)=MrP.a*x1P(:,i-1)+MrP.b*Refer_P(2:2:6);
yMrefP(:,i)=MrP.c*x1P(:,i-1)+MrP.d*Refer_P(2:2:6);
18

% Bucle Abierto
20 x2P(:,i)=FFP.a*x2P(:,i-1)+FFP.b*Refer_P(2:2:6);
UP(:,i)=FFP.c*x2P(:,i-1)+FFP.d*Refer_P(2:2:6);
22

% Bucle Cerrado
24 % Calcular Salida del Modelo y Error
Salida_Modelo=T-TOP;
26 ErrorP= Salida_Modelo(2:2:6)-yMrefP(:,i);

28 % Control Bucle cerrado: salida rlqg
if AuP(1,i-1)+UP(1,i-1) >=WmaxP(1)-woP(1)
30     AuP(1,i-1)=WmaxP(1)-UP(1,i-1);
elseif AuP(1,i-1)+UP(1,i-1) <=WminP(1)-woP(1)
32     AuP(1,i-1)=WminP(1)-UP(1,i-1);
end
34 if AuP(2,i-1)+UP(2,i-1) >=WmaxP(2)-woP(2)
     AuP(2,i-1)=WmaxP(2)-UP(2,i-1);
36 elseif AuP(2,i-1)+UP(2,i-1) <=WminP(2)-woP(2)
     AuP(2,i-1)=WminP(2)-UP(2,i-1);
38 end
if AuP(3,i-1)+UP(3,i-1) >=WmaxP(3)-woP(3)
40     AuP(3,i-1)=WmaxP(3)-UP(3,i-1);
elseif AuP(3,i-1)+UP(3,i-1) <=WminP(3)-woP(3)
42     AuP(3,i-1)=WminP(3)-UP(3,i-1);

```

```

end
44
x3P(:, i)=regP.a*x3P(:, i-1)+regP.b*[AuP(:, i-1); ErrorP];
46 AuP(:, i)=regP.c*x3P(:, i-1)+regP.d*[AuP(:, i-1); ErrorP];

48 %Control 2DOF
    WP=UP(:, i)+AuP(:, i)+woP;
50
    %Modelo del Actuador
52 %Saturar la Potencia
    if WP(1)<=WminP(1)
54     WsatP(1)=WminP(1);
    elseif WP(1)>=WmaxP(1)
56     WsatP(1)=WmaxP(1);
    else
58     WsatP(1)=WP(1);
    end
60 if WP(2)<=WminP(2)
    WsatP(2)=WminP(2);
62 elseif WP(2)>=WmaxP(2)
    WsatP(2)=WmaxP(2);
64 else
    WsatP(2)=WP(2);
66 end
68 if WP(3)<=WminP(3)
    WsatP(3)=WminP(3);
    elseif WP(3)>=WmaxP(3)
70     WsatP(3)=WmaxP(3);
    else
72     WsatP(3)=WP(3);
74 end
    %Pasar Potencia a Vdaq
76 VdaqR(1)=polyval(Pvdaq(1,:), WsatP(1), Svdaq(1,:), MUvdaq(1:2));
    VdaqR(2)=polyval(Pvdaq(2,:), WsatP(2), Svdaq(2,:), MUvdaq(3:4));
78 VdaqR(3)=polyval(Pvdaq(3,:), WsatP(3), Svdaq(3,:), MUvdaq(5:6));
    Actuadores=[6 6 6 6 6 6 6 VdaqR(1) VdaqR(2) VdaqR(3)];
80
    %Pintar Referencia
82 Ref1=T(1);
    Ref2=yMrefP(1, i)+TOP(2);
84 Ref3=T(3);
    Ref4=yMrefP(2, i)+TOP(4);
86 Ref5=T(5);
    Ref6=yMrefP(3, i)+TOP(6);
88 %Pintar Acciones de control

```

```

Augraf=[0 0 0 0 0 AuP(:,i)'];
90 Ugraf=[0 0 0 0 0 UP(:,i)'];
    %Variables a guardar
92 SalidaP=[SalidaP T];
    PotenciaP=[PotenciaP; WsatP];
94 VtarjetaP=[VtarjetaP; Actuadores];
    %Calculo punto fun
96 if i>10
        act(:,j)=[6;6;6;6;6;6;WsatP'];
98     woN=mean(act,2);
        j=j+1;
100 end

```

## C.6. HMI Control Completo

En esta sección el código introducido en los MATLAB Scripts de Labview al completo (o referenciado).

### C.6.1. Cargar todos los datos

```

%Indico el workspace...
2 clear all;
  cd('C:\Users\alesca.UPVNET\Desktop\HMI\HMI Control')
4 %Cargo los parámetros para pasar de V a W...
  load('Pw','Pw');
6  load('Sw','Sw');
  load('MUw','MUw');
8  %Cargo los parámetros para pasar de W a V
  load('Pvdaq','Pvdaq');
10 load('Svdaq','Svdaq');
  load('MUvdaq','MUvdaq');
12 %Cargo los datos para pasar de V a °C
  load('P','P');
14 load('S','S');
  load('MU','MU');
16
  %Cargo los reguladores para Puesta punto
18 load('M3d');
  MrP=M3d;
20 load('Gr13d');
  FFP=Gr13d;

```

```

22 load('reg13d');
   regP=reg13d;
24
   refa=[55 55 55 55 55 55]
26 Vinicial=[10 10 10 10 10 10 10 10 0 0 0];

28 %Carga los reguladore para Control del Modelo Nominal
   load('M6d');
30 MrN=M6d
   load('Gr06d');
32 FFN=Gr06d;
   %load('reg16d');
34 load('reg16d');
   regN=reg16d;

```

### C.6.2. Genero estructura de datos para fase de puesta a punto

```

1 SalidaP=[];
   PotenciaP=[];
3 VtarjetaP=[];

5 %Pasar a Temperatura
   T01 = polyval(P(1,:),V0(1),S(1,:),MU(1,:));
7 T02 = polyval(P(2,:),V0(2),S(2,:),MU(2,:));
   T03 = polyval(P(3,:),V0(3),S(3,:),MU(3,:));
9 T04 = polyval(P(4,:),V0(4),S(4,:),MU(4,:));
   T05 = polyval(P(5,:),V0(5),S(5,:),MU(5,:));
11 T06 = polyval(P(6,:),V0(6),S(6,:),MU(6,:));
   TOP=[T01;T02;T03;T04;T05;T06];
13 m=10000;

15 %Estructura para Mr
   x1P(1:size(MrP.a,1),1:m)=0;
17 yMrefP(1:size(MrP.c,1),1:m)=0;

19 %Estructura para FF
   x2P(1:size(FFP.a,1),1:m)=0;
21 UP(1:size(FFP.c,1),1:m)=0;

23 %Estructura para Iqg
   x3P(1:size(regP.a,1),1:m)=0;
25 AuP(1:size(regP.c,1),1:m)=0;

```

```

27 woP=[0;0;0];
    WmaxP=[460;485;491];
29 WminP=[0;0;0];
    j=1;

```

### C.6.3. Implemtación del Bucle de Puesta a Punto

Coincide con el código de la sección [C.5](#).

### C.6.4. Genero estructura de datos para Fase de Control

```

%Para N
2 SalidaN=[];
  PotenciaN=[];
4 VtarjetaN=[];

6 %Pasar a Temperatura
  T01 = polyval(P(1,:),V0(1),S(1,:),MU(1,:));
8  T02 = polyval(P(2,:),V0(2),S(2,:),MU(2,:));
  T03 = polyval(P(3,:),V0(3),S(3,:),MU(3,:));
10 T04 = polyval(P(4,:),V0(4),S(4,:),MU(4,:));
  T05 = polyval(P(5,:),V0(5),S(5,:),MU(5,:));
12 T06 = polyval(P(6,:),V0(6),S(6,:),MU(6,:));
  TON=[T01;T02;T03;T04;T05;T06];
14 m=10000;

16 %Para Control
  %Estructura para Mr
18 x1N(1:size(MrN.a,1),1:m)=0;
  yMrefN(1:size(MrN.c,1),1:m)=0;
20
  %Estructura para FF
22 x2N(1:size(FFN.a,1),1:m)=0;
  UN(1:size(FFN.c,1),1:m)=0;
24
  %Estructura para Iqg
26 x3N(1:size(regN.a,1),1:m)=0;
  AuN(1:size(regN.c,1),1:m)=0;
28
30 WmaxN=[460;485;491;10;10;10;10;10];
  WminN=[3.5;3.5;3.5;3.5;3.5;0;0;0];

```

### C.6.5. Código Control Modelo Completo, $G^2(s)$

```

Tcal=1200
2  %Pasar a Temperatura
T1 = polyval(P(1,:),V(1),S(1,:),MU(1,:));
4  T2 = polyval(P(2,:),V(2),S(2,:),MU(2,:));
T3 = polyval(P(3,:),V(3),S(3,:),MU(3,:));
6  T4 = polyval(P(4,:),V(4),S(4,:),MU(4,:));
T5 = polyval(P(5,:),V(5),S(5,:),MU(5,:));
8  T6 = polyval(P(6,:),V(6),S(6,:),MU(6,:));
T=[T1; T2; T3; T4; T5; T6];
10
Refer_N=ref-T0N;
12  % Salida Mr
x1N(:,i)=MrN.a*x1N(:,i-1)+MrN.b*Refer_N;
14  yMrefN(:,i)=MrN.c*x1N(:,i-1)+MrN.d*Refer_N;

16  %Bucle Abierto
x2N(:,i)=FFN.a*x2N(:,i-1)+FFN.b*Refer_N;
18  UN(:,i)=FFN.c*x2N(:,i-1)+FFN.d*Refer_N;

20  %Bucle Cerrado
  %Calcular Salida del Modelo y Error
22  Salida_Modelo=T-T0N; %Transponer ref o TON
ErrorN= Salida_Modelo-yMrefN(:,i);
24
  %Control Bucle cerrado
26  if AuN(1,i-1)+UN(1,i-1) >=WmaxN(1)-woN(1)
      AuN(1,i-1)=WmaxN(1)-UN(1,i-1);
28  elseif AuN(1,i-1)+UN(1,i-1)<=WminN(1)-woN(1)
      AuN(1,i-1)=WminN(1)-UN(1,i-1);
30  end
  if AuN(2,i-1)+UN(2,i-1) >=WmaxN(2)-woN(2)
32      AuN(2,i-1)=WmaxN(2)-UN(2,i-1);
  elseif AuN(2,i-1)+UN(2,i-1)<=WminN(2)-woN(2)
34      AuN(2,i-1)=WminN(2)-UN(2,i-1);
  end
36  if AuN(3,i-1)+UN(3,i-1) >=WmaxN(3)-woN(3)
      AuN(3,i-1)=WmaxN(3)-UN(3,i-1);
38  elseif AuN(3,i-1)+UN(3,i-1)<=WminN(3)-woN(3)
      AuN(3,i-1)=WminN(3)-UN(3,i-1);

```

```

40 end
   if AuN(4,i-1)+UN(4,i-1) >=WmaxN(4)-woN(4)
42     AuN(4,i-1)=WmaxN(4)-UN(4,i-1);
   elseif AuN(4,i-1)+UN(4,i-1)<=WminN(4)-woN(4)
44     AuN(4,i-1)=WminN(4)-UN(4,i-1);
   end
46 if AuN(5,i-1)+UN(5,i-1) >=WmaxN(5)-woN(5)
     AuN(5,i-1)=WmaxN(5)-UN(5,i-1);
48 elseif AuN(5,i-1)+UN(5,i-1)<=WminN(5)-woN(5)
     AuN(5,i-1)=WminN(5)-UN(5,i-1);
50 end
   if AuN(6,i-1)+UN(6,i-1) >=WmaxN(6)-woN(6)
52     AuN(6,i-1)=WmaxN(6)-UN(6,i-1);
   elseif AuN(6,i-1)+UN(6,i-1)<=WminN(6)-woN(6)
54     AuN(6,i-1)=WminN(6)-UN(6,i-1);
   end
56 if AuN(7,i-1)+UN(7,i-1) >=WmaxN(7)-woN(7)
     AuN(7,i-1)=WmaxN(7)-UN(7,i-1);
58 elseif AuN(7,i-1)+UN(7,i-1)<=WminN(7)-woN(7)
     AuN(7,i-1)=WminN(7)-UN(7,i-1);
60 end
   if AuN(8,i-1)+UN(8,i-1) >=WmaxN(8)-woN(8)
62     AuN(8,i-1)=WmaxN(8)-UN(8,i-1);
   elseif AuN(8,i-1)+UN(8,i-1)<=WminN(8)-woN(8)
64     AuN(8,i-1)=WminN(8)-UN(8,i-1);
   end
66
     x3N(:,i)=regN.a*x3N(:,i-1)+regN.b*[AuN(:,i-1);ErrorN];
68     AuN(:,i)=regN.c*x3N(:,i-1)+regN.d*[AuN(:,i-1);ErrorN];

70 %Control 2DOF
WN= UN(6:8,i)+AuN(6:8,i)+woN(6:8);
72 VN= UN(1:5,i)+AuN(1:5,i)+woN(1:5);

74 %Modelo del Actuador para los radiadores
   %Saturar la Potencia
76 if VN(1)<=WminN(1)
     WsatN(1)=WminN(1);
78 elseif VN(1)>=WmaxN(1);
     WsatN(1)=WmaxN(1);
80 else
     WsatN(1)=VN(1);
82 end
   if VN(2)<=WminN(2)
84     WsatN(2)=WminN(2);
   elseif VN(2)>=WmaxN(2);

```

```

86     WsatN(2)=WmaxN(2);
      else
88     WsatN(2)=VN(2);
      end
90 if VN(3)<=WminN(3)
      WsatN(3)=WminN(3);
92 elseif VN(3)>=WmaxN(3);
      WsatN(3)=WmaxN(3);
94 else
      WsatN(3)=VN(3);
96 end

98 %Pasar Potencia a Vdaq
      VdaqR(1)=polyval(Pvdaq(1,:),WsatN(1),Svdaq(1,:),MUvdaq(1:2));
100 VdaqR(2)=polyval(Pvdaq(2,:),WsatN(2),Svdaq(2,:),MUvdaq(3:4));
      VdaqR(3)=polyval(Pvdaq(3,:),WsatN(3),Svdaq(3,:),MUvdaq(5:6));
102
      GRAFICOS
104 %Escribir Accion
      Actuadores=[VN(5) VN(3) VN(2) VN(1)
106                 VN(4) VN(1) VN(2) VN(3)
                  VdaqR(1) VdaqR(2) VdaqR(3)];
108
      %Pintar Referencia
110 Ref1=yMrefN(1,i)+TON(1);
      Ref2=yMrefN(2,i)+TON(2);
112 Ref3=yMrefN(3,i)+TON(3);
      Ref4=yMrefN(4,i)+TON(4);
114 Ref5=yMrefN(5,i)+TON(5);
      Ref6=yMrefN(6,i)+TON(6);
116 %Pintar Acciones de control
      %Wsat=WsatN;
118 %Wgraf=WN';
      Augraf=AuN(:,i)';
120 Ugraf=UN(:,i)';

122 %Variables a guardar
      SalidaN=[SalidaN T];
124 PotenciaN=[PotenciaN; WsatN];
      VtarjetaN=[VtarjetaN; Actuadores];

```

### C.6.6. Guardar datos

```
1 save('Experimentos\AuP.mat','AuP')
  save('Experimentos\UP.mat','UP')
3 save('Experimentos\PotenciaP.mat','PotenciaP')
  save('Experimentos\yMrefP.mat','yMrefP')
5 save('Experimentos\VtarjetaP.mat','VtarjetaP')
  save('Experimentos\SalidaP.mat','SalidaP')
7
  save('Experimentos\AuN.mat','AuN')
9 save('Experimentos\UN.mat','UN')
  save('Experimentos\PotenciaN.mat','PotenciaN')
11 save('Experimentos\yMrefN.mat','yMrefN')
  save('Experimentos\VtarjetaN.mat','VtarjetaN')
13 save('Experimentos\SalidaN.mat','SalidaN')
15 Vfinal=[10 10 10 10 10 10 10 10 0 0 0]
```

# Bibliografía

- [1] *Medición de tiempo de cocción en hornos continuos por Silge Electronica S.A. Nota de aplicación AP05*, 2001. [1.1.1](#)
- [2] A. B. Aguado. *Identificación y control adaptativo*. Pearson, 2003. [4.4](#), [3b](#), [4b](#)
- [3] P. Albertos and A. Sala. *Multivariable Control Systems*. Springer, 2004. ([document](#)), [7.2](#), [7.3](#), [7.3](#), [7.4.2](#), [7.6](#)
- [4] S. Alepuz Menéndez and J. Bordonau Farrerons. Aportación al control del convertidor CC/CA de tres niveles. Master's thesis, Universitat Politècnica de Catalunya, 2004. [7.2](#)
- [5] Alferieff. *Hornos Continuos*. website: [www.alferieff.com](http://www.alferieff.com) / contact: [hornos@alferieff.com](mailto:hornos@alferieff.com). [1.1](#), [1.1.1](#), [1.2.1](#)
- [6] M. Barreras Carracedo, S. Díaz, and M. García-Sanz. Modelado y simulación de un horno eléctrico industrial de 1MW de potencia. *XXIV Jornadas de Automática León*, 2003. ([document](#)), [2](#), [1.2](#), [2.2.1](#)
- [7] Caneparo. *Fábrica de Máquinas, Hornos Túnel Continuos y Moldes Rotativos para Galletitas y Afines*. website: [www.caneparo.com.ar](http://www.caneparo.com.ar) / contact: [info@caneparo.com.ar](mailto:info@caneparo.com.ar). [1.1](#), [1.1.1](#), [1.2.1](#)
- [8] S. Ciruelos. Diseño y evaluación de las prestaciones de un sistema de control de la pila de combustible. Master's thesis, Universitat Politècnica de Catalunya, 2006.
- [9] B. Claire Torrico and J. E. Normey-Rico. 2-DoF discrete dead-time compensators for stable and integrative processes with dead-time. *Process Control*, 2004. [7.2](#)
- [10] J. A. Cortes Osorio, J. A. Chavez Osorio, and H. B. Cano Garzón. Fundamentos y aplicación del muestreo en señales ubicadas en las bandas de alta del espectro. *Scientia et Technica*, 2008.

- [11] A. Creus Solé. *Instrumentación Industrial, 6ª Edición*. Marcombo, 1997. [A.1](#)
- [12] W. Daowei, W. Youyi, and G. Guoxiao. Short-span seeking control for dual-stage servomechanism in hard disk drives. *Asian Control Conference*, 2004. [7.2](#)
- [13] F. Dorado et al. Modelado y control de un horno de yeso rotativo y continuo. *XXI Jornadas de Automática*, 2000. ([document](#))
- [14] X. Elías. La fabricación de los materiales cerámicos, 3ª parte. Technical report, Universidad Pontificia Bolivariana, Centro Nacional Producción Mas Limpia y Fundación Universitaria Iberoamericana, 2001. [1.1](#), [1.1.1](#), [1.2](#), [1.2](#), [1.2.1](#)
- [15] Emison. *Tratamiento térmico Metales en hornos continuos*. website: [www.emison.com / contact: braso@emison.com](http://www.emison.com/contact:braso@emison.com). [1.1](#), [1.1.1](#), [1.2](#)
- [16] J. Espinosa. Curso de control iii. ejercicio ilustrativo. Technical report, Universidad Nacional de Colombia, 2006.
- [17] A. Flores. Control en cascada. Technical report, Universidad Iberoamericana-Santa Fe, 2009.
- [18] ADLink Technology Inc. *DAQ-MTLB Data Acquisition Toolbox Adapter User's Manual*. website: [www.adlinktech.com](http://www.adlinktech.com). [1](#)
- [19] ADLink Technology Inc. *DAQPilot Function Reference Manual LabVIEW™ Interfaces For ADLINK DAQ Devices*. website: [www.adlinktech.com](http://www.adlinktech.com). [2](#)
- [20] ADLINK Technology Inc. *NuDAQ 6208/6216 Series Multi-channel Analog Output Cards User's Guide*. website: [www.adlinktech.com](http://www.adlinktech.com). [3.1](#)
- [21] ADLink Technology Inc. *PCI-9111 Multi-Functions Data Acquisition Card User's Guide*. website: [www.adlinktech.com](http://www.adlinktech.com). [3.1](#)
- [22] L. A. Lahoz and F. Miyara. Introducción al análisis frecuencial y al ruido eléctrico. Technical report, Universidad Nacional de Rosario, 2003. [a](#), [b](#), [c](#)
- [23] J. R. Lajara Vizcaíno and J. Pelegrí Sebastiá. *LabVIEW Entorno gráfico de programación*. Marcombo, 2006. [3.2](#)
- [24] L. Ljung. *System Identification: Theory for the User, 2nd Edition*. Kindle Books, 1999. [4.2](#)
- [25] X. López and A. Portero. Desenvolupament d'una xarxa de sensors i actuadors sense fils de

- propòsit industrial. Master's thesis, Universitat Autònoma de Barcelona, 2008.
- [26] J. C. Maraña. Short-span seeking control for dual-stage servomechanism in hard disk drives. *Asian Control Conference*, 2004.
- [27] MathWorks. *Matlab Help*. website: [www.mathworks.com](http://www.mathworks.com). [2.2.1](#)
- [28] D. Miraut. Aliasing y como combatirlo. Técnicas avanzadas de gráficos 3D. Máster Oficial en Informática Gráfica, Juegos y Realidad Virtual. Universidad Rey Juan Carlos.
- [29] Nabertherm. *Tratamiento térmico Metales, plástico y técnicas de superficies. Hornos e instalaciones para: Revenir, Recocer/Templar, Proceso de revenido, Recocer por disolución, Soldadura, Endurecer, Envejecer, Precalentar, Secar, Forjar*. website: [www.nabertherm.es](http://www.nabertherm.es) / contact: [contact@nabertherm.de](mailto:contact@nabertherm.de). [1.1](#), [1.1.1](#), [1.2](#), [1.2.1](#)
- [30] K. Ogata. *Ingeniería de Control Moderna, Tercera Edición*. Pearson, 1998. [4.4](#), [7.3](#)
- [31] A. Pedro, A. Sala, and V. Marina. Cuestiones en la implementación del control multivariable: Relevancia de señales y priorización. *XXV Jornadas de Automática*, 2004.
- [32] S Piccacli and A. Visioli. An optimal feedforward control design for the set-point following of mimo processes. *Journal of Process Control*, 2009.
- [33] J. Quevedo Casin and S. Ciruelos Lao. Diseño y evaluación de las prestaciones de un sistema de control de la pila de combustible. Master's thesis, Universidad Politécnica de Barcelona, 2006.
- [34] RS. *Hoja de características del TX104TC In-Head Thermocouple*. website: [www.rs-components.com](http://www.rs-components.com). [2.2.2](#)
- [35] Hornos Eléctricos Lago S.A. Horno continuo de gas (Patente), 2001. [1.1](#), [1.1.1](#), [1.2](#)
- [36] M. E. Salgado Brocal. Inversión, un concepto unificador en la enseñanza de control automático. *Revista iberoamericana de automática e informática industrial (RIAI)*, 2006.
- [37] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design, 2nd Edition*. Wiley, 2005. [7.4.2](#)
- [38] A. Solera. Filtro de kalman. Technical report, Banco Central de Costa Rica, 2003.
- [39] R. N. Souza, J. R. B. Souza, and E. J. Matos. Uso de pré-compensador dinâmico no projeto

- de controlador robusto para sistema mimo instável. *Revista IEEE América Latina*, 2008. [7.3](#)
- [40] STMicroelectronics. *Hoja de características del L165*. website: STMicroelectronics. [2.2.1](#)
- [41] C. Valdivieso, I. Pazmiño, and A. Torres. Interfaz de receptor de bucle de 4 – 20mA en aplicación con DSPIC. 2004.
- [42] E. Vallejo. Identificación paramétrica de sistemas dinámicos. universidad del norte. *Ingeniería & Desarrollo*, 1997. [4.2](#), [4.5.1](#)
- [43] R. Velázquez Cuevas. Análisis de la respuesta transitoria de a sistemas de primer y segundo orden lti. 2008.
- [44] F. Vázquez, F. Morilla, and J. Garrido. Control pid 3x3 de un grupo caldera turbina. *XXIX Jornadas de Automática*, 2008. [7.6](#)