

METODOLOGÍA PARA EL DISEÑO JERÁRQUICO DE SISTEMAS DE CONTROL INDUSTRIAL VALIDADO MEDIANTE MODELADO BASADO EN COMPONENTES

N. Iriondo*, F. Artaza, S. Calvo, I. Cabanes, D. Orive, M. Marcos

Escuela Superior de Ingenieros de Bilbao (Universidad del País Vasco)
Alameda Urquijo s/n 48013 Bilbao
Teléfono: 34 94 601 4049; Fax: 34 94 601 4187
e-mail: jtpirurn@bi.ehu.es

Resumen

En este artículo se propone una metodología para el diseño jerárquico de sistemas de control y de modelos de proceso. Uno de los principales objetivos es la reutilización a todos los niveles de la jerarquía, tanto dentro de la misma aplicación como en otras aplicaciones. Este artículo aplica la metodología al diseño de una línea de tratamiento en caliente. El diseño del sistema de control ha sido desarrollado en ISaGRAF Enhanced, que es un entorno de programación para PLCs que cumple con el estándar IEC 61131. El diseño del modelo del proceso se ha desarrollado en Matlab/Simulink. Para validar el diseño del sistema de control se han hecho pruebas unitarias y de integración usando un marco de co-simulación desarrollado dentro del proyecto FLEXICON.

Palabras Clave: ISaGRAF, IEC 61131-3, diseño jerárquico, sistemas de control, simulación, modelado del proceso.

1 INTRODUCCIÓN

1.1 ESTÁNDAR IEC 61131-3 PARA PLC

La mayoría de los sectores industriales emplean PLCs (Programmable Logic Controller) para realizar el control de su sistema productivo. En los últimos años estamos asistiendo a importantes avances tecnológicos en estos controladores que son cada vez más demandados para mejorar la fabricación y optimizar procesos, al tiempo que se reducen los costes. Las aplicaciones presentes y futuras se caracterizan por la integración de los PLCs con otros sistemas y dispositivos, precisando además que dichos controladores sean lo suficientemente flexibles para ser capaces de adaptar rápidamente las estrategias de control a los cambios que requiera el proceso. Como consecuencia, se requieren sistemas abiertos que se puedan integrar tanto en células de

producción como en sistemas computacionales de un nivel superior en la pirámide de automatización.

Así mismo, la aplicación de estándares también tiene un gran impacto en el rápido crecimiento del mercado de la instrumentación y control de procesos industriales. En este sentido, la aparición del estándar de programación IEC1131-3 [13] proporciona lenguajes y métodos estandarizados que permiten resolver un amplio rango de problemas tecnológicos como elementos software no propietarios. Los fabricantes de PLCs están cumpliendo cada vez más con este estándar, y da la sensación de que dentro de unos años estará fuertemente introducido en la industria. Ya se han desarrollado estructuras de control jerárquico utilizando Step7 de Siemens [22]. En este artículo se utiliza ISaGRAF Enhanced como herramienta para programar el sistema de control. Se trata de una herramienta de programación de PLCs que cumple básicamente el estándar IEC 61131-3. A continuación se enumeran algunas de las características más importantes de este estándar [13]:

- Datos fuertemente tipados.
- Permite que diferentes partes del programa se puedan ejecutar con una frecuencia diferente.
- Soporta el diseño de comportamientos secuenciales complejos mediante el lenguaje *Sequential Function Chart*.
- Permite la definición de estructuras de datos.
- Posibilita la programación en diferentes lenguajes, concretamente ofrece tres lenguajes gráficos y dos textuales para expresar distintas partes del control de la aplicación.

El estándar IEC1131 describe los programas, funciones y bloques funcionales (Function Blocks, FBs) como diferentes tipos de POU (Program Organization Units), y pueden ser utilizados varias veces a lo largo de las distintas partes que componen una aplicación. Por ejemplo, los bloques funcionales se definen por medio de una declaración de tipo, y se utilizan copias de este bloque funcional denominadas instancias. La definición de POU favorece la

reutilización del software desde el más alto nivel, con los programas, hasta el más bajo nivel, por medio de las funciones y los bloques funcionales, como se muestra en la tabla 1.

Tipo POU	Implementación	comentarios
Tipo Programa	Programa instancia	Máximo nivel de reutilización
Tipo Bloque Funcional	Bloque Funcional instancia	Permiten la descomposición de un algoritmo de control complejo en algoritmos simples que pueden ser reutilizados
Tipo Función	Función	Para manipulación de datos

Tabla 1: Tipo de POUs

1.2 VISIÓN GENERAL DEL DISEÑO JERÁRQUICO DE SISTEMAS DE CONTROL

Cuanto más madura se vuelve la industria mayor es la necesidad de consolidar una metodología para el diseño de sistemas de control. Los sistemas de control jerárquico empezaron siendo usados en sistemas robóticos debido a su gran complejidad [8]. También se usaban comúnmente para sistemas de control robusto no lineal [12] y para sistemas de control óptimo, debido a que muchos de estos problemas de control son inabordables de forma analítica. El control inteligente [10] es otro campo donde los sistemas de control jerárquico son muy utilizados, sobre todo para sistemas expertos [23].

Hoy en día, los sistemas de control jerárquico se utilizan en todo tipo de sistemas de control, y no solo en los más complejos. Mesarovic [18] hizo una clasificación y descripción de los principales tipos de estructuras jerárquicas. Varaiya [24] expuso los principios más importantes que hay que tener en cuenta a la hora de dividir en capas un sistema de control. Estos diseños jerárquicos pueden también ser más complejos. Así, por ejemplo, Pappas et al [20] utilizan un algoritmo de controlabilidad para chequear la consistencia, Huang [9] aplica su diseño a un interfaz de operador con distintos niveles de autoridad, y Bradac [2] añade al diseño un modelo matemático de las comunicaciones desarrollado en Matlab.

1.3 VISIÓN GENERAL DEL MODELADO BASADO EN COMPONENTES

Hoy en día, la mayoría de los fabricantes industriales se enfrentan a grandes desafíos para conseguir reducir el tiempo de salida al mercado de sus diseños. Por ello se está usando cada día más el denominado “portotipado virtual”, donde un modelo que se

ejecuta en un ordenador sustituye al prototipo físico. Con esto se consiguen grandes beneficios, tales como un menor coste, un desarrollo más rápido, una evaluación de mayor número de alternativas de diseño y una obtención de diseños más óptimos.

Mackulak y Cochran [15] estiman que el 45% del esfuerzo de simulación de un proyecto está en su construcción, formulación y traducción. Así, implementar un modelo genérico que sea aplicable a muchos sistemas puede ahorrar una gran cantidad de tiempo en un estudio de simulación. Por tanto, la reutilización de modelos permite realizar análisis de simulaciones más baratos y rápidos [6]. Sin embargo, para ello debe existir una librería de módulos reutilizables.

Normalmente, las herramientas de diseño están relacionadas con librerías de componentes de modelado reutilizables que hacen más fácil la descripción del modelo y su posterior validación. Ruppel y Wysor [21] analiza y compara algunas de estas herramientas de modelado. Hay algunas herramientas que son muy específicas, como LabVIEW para instrumentación y VHDL para componentes electrónicos. Entre las herramientas más genéricas cabe destacar ACSL (Advanced Continuous Simulation Language) [17], que tiene la gran ventaja de su experiencia al ser la primera herramienta en ser introducida en el mercado, MATRIXx/RealSim [25], que ha sido adquirida por The MathWorks y tiende a desaparecer, SCILAB-SCICOS [11], que tiene la gran ventaja de pertenecer al denominado software libre, y 20-sim [3], que permite interactuar con Matlab/Simulink, que es la herramienta más extendida de todas. Esta ha sido además la razón fundamental para elegir Matlab/Simulink como herramienta de modelado. Además de esto, ACSL es más cara y SCILAB-SCICOS tiene poca experiencia en el mercado. Otra herramienta no muy extendida todavía pero que merece mención especial es Modelica [4], que es un nuevo lenguaje de modelado muy centrado en la reutilización de componentes. Modelica está pensado como un nuevo estándar para intercambio de modelos y librerías. Es por ello que es un lenguaje textual, sin entornos tan amigables como puede tener Matlab, ACSL, 20-sim o SCILAB-SCICOS.

1.4 DISEÑO JERÁRQUICO. VALIDACIÓN

En este artículo se presenta una metodología para el diseño jerárquico de sistemas de control industriales. El sistema de control se especifica como una jerarquía de N niveles. Cada nivel de la jerarquía está formado por un conjunto de Componentes Básicos Funcionales (Functional Basic Components, FBCs) con unas entradas y unas salidas. Todos los FBCs, independientemente de la capa a la que pertenezcan,

están definidos de la misma forma y tienen los mismos componentes y características. Un FBC de un determinado nivel se define a través de un conjunto de FBCs del nivel inferior. El concepto de FBC nos permite definir la jerarquía a nivel de funcionalidad, pero para definir la jerarquía a nivel de programación necesitamos usar el concepto de FB y de instancia de FB.

Los FBCs pueden reutilizarse dentro del mismo modelo, pero también en otros dominios de aplicación. Hoy día hay muchas aplicaciones industriales que tienen elementos comunes que tan solo difieren en el valor de algunos parámetros de configuración. Tal es el caso, por ejemplo, de una línea de tratamiento en caliente, una línea de laminación de acero o un sistema de acabado superficial. Así, los elementos comunes de estas aplicaciones pueden ser definidos una sola vez y después utilizados tantas veces como sea necesario.

Por otro lado, para poder desarrollar pruebas de validación incrementales del sistema de control es necesario tener también un modelado jerárquico del proceso. Así, es este artículo también se presenta una metodología para el modelado jerárquico basado en componentes. Estos componentes genéricos del modelo tienen que corresponderse con los FBCs del sistema de control. Así, podremos validar el sistema de control de forma incremental.

Dentro del proyecto FLEXICON se ha desarrollado un marco de co-simulación [16] que permite realizar pruebas de integración a través de CORBA. El prototipo desarrollado permite la integración entre el modelo del proceso definido en Simulink y el sistema de control diseñado en ISaGRAF Enhanced. Existe otro desarrollo interesante para validar sistemas de control basados en IEC 61131-3, pero utilizando Modelica en vez de Matlab y generando un ejecutable por medio de tecnologías XML [7].

La exposición del trabajo se organiza en 7 apartados: tras la introducción, en el apartado 2 se presenta la metodología utilizada. En el apartado 3 se describen las características principales de la aplicación utilizada como caso de estudio, una línea de tratamiento en caliente. En el resto de apartados se explican el diseño modular del modelo y el diseño jerárquico del sistema de control para la línea de tratamiento en caliente, así como las pruebas de validación del sistema de control. Finalmente, en el último apartado se exponen las conclusiones obtenidas.

2 METODOLOGÍA

El diseño del sistema de control se ha desarrollado siguiendo las pautas y recomendaciones del estándar IEC para desarrollo de software para PLCs. Esto permite implementar la jerarquía de control de arriba hacia abajo o de abajo hacia arriba. El diseño jerárquico permitirá la reutilización en cada nivel de la jerarquía. El diseño del sistema de control está compuesto de un número de instancias de bloques funcionales genéricos. El diseño de estos bloques genéricos se puede hacer en ISaGRAF usando cualquiera de los cinco lenguajes definidos en el estándar IEC 61131-3, y después serán almacenados como librerías de FBs.

La especificación del sistema de control consiste en una jerarquía de cuatro niveles:

- NIVEL 0: planta completa.
- NIVEL 1: subsistemas independientes de la planta.
- NIVEL 2: elementos funcionales asociados al subsistema de nivel 1.
- NIVEL 3: componentes funcionales elementales asociados a cada elemento de nivel 2.

Cada nivel contiene un conjunto de FBCs y un conjunto de conexiones de entrada y salida. La detección de fallos se realiza por medio de una serie de enclavamientos y sensores replicados (denominados seguridades). Así, el sistema de control incluye tanto la acción de control en funcionamiento normal como el procesado de los enclavamientos y las seguridades para tratar los fallos. Normalmente, la acción a realizar por parte del control para tratar un fallo consiste en enviar una alarma al operador y en realizar una acción sobre el proceso.

Las entradas y salidas de un FBC se agrupan en estructuras que dependen de su funcionalidad.

- Entradas a FBCs:
 - Seguridades: señales de alarma
 - Enclavamientos: condiciones que se deben cumplir para que se pueda ejecutar el bloque.
 - Comandos de Operador
 - Datos de Operador: datos introducidos por el operador
 - Datos de proceso: señales de campo
 - Datos externos: datos procedentes de otros sistemas independientes.
- Salidas de FBCs:
 - Señalización: lámparas o indicadores sonoros.
 - Datos de proceso: señales de campo.

Cada FBC se caracteriza por sus entradas, salidas, parámetros de configuración y datos internos, como puede observarse en la figura 1.

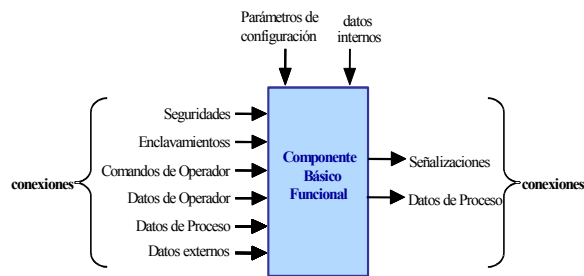


Figura 1: FBC genérico

La jerarquía definida para la especificación del sistema de control facilita la reutilización de los FBCs de cualquier nivel. Para conseguir esto, todos los FBCs pertenecientes a una aplicación concreta están descritos de la siguiente forma:

- TIPO: tipo de FBC que se está caracterizando.
- CATEGORIA: el path dentro de la jerarquía del FBC.
- ETIQUETA: identificador asignado por el programador.
- ENTRADAS: lista de conexiones de entrada (analógicas o digitales). Pueden venir de otros bloques o del exterior (señales de campo).
- SALIDAS: lista de conexiones de salida (analógicas o digitales). Pueden ir a otros bloques o al exterior (señales de campo).
- DATOS INTERNOS: lista de variables locales necesarias para poder implementar la funcionalidad del FBC.
- PARAMETROS DE CONFIGURACION: usados para adaptar la lógica del FBC.
- FUNCION: si el FBC es de nivel 3 se describe el código fuente que ejecuta su funcionalidad. Si no, se enumeran los FBCs de nivel inferior que contiene.

Por otro lado, el diseño modular del proceso se ha realizado siguiendo las pautas y recomendaciones de uno de los miembros del proyecto FLEXICON, y tiene que desarrollarse de acuerdo con el diseño jerárquico del sistema de control. El diseño jerárquico permitirá también la reutilización de componentes genéricos en cada nivel de la jerarquía y la implementación de la jerarquía del modelo de arriba hacia abajo o de abajo hacia arriba. El sistema debe representar tanto los aspectos mecánicos como los balances de propiedades que intervienen en la aplicación. Una vez que el subsistema ha sido modelado como un componente genérico y su contenido ha sido enmascarado, ya puede ser

utilizado configurando los distintos parámetros y conectándolo a otros subsistemas del mismo nivel. Todos los componentes genéricos se describen de la misma forma, y tienen el aspecto que se muestra en la figura 2.

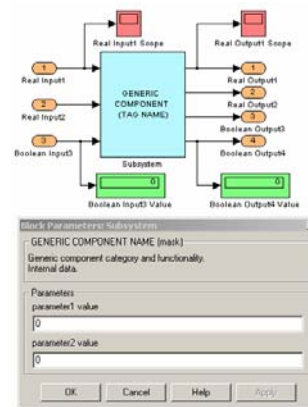


Figura 2: Componente genérico en Simulink

3 APLICACIÓN INDUSTRIAL: LÍNEA DE TRATAMIENTO EN CALIENTE

En este artículo se ha utilizado una línea de tratamiento en caliente (Heat Treatment Line, HTL) como caso de estudio. La HTL estará controlada por un sistema de control híbrido [19].

Se trata de una HTL que procesa piezas metálicas para proporcionarles las características necesarias de dureza y ductilidad. Las piezas que van a ser tratadas serán tornillos, tuercas y demás tipos de piezas pequeñas y ligeras. Así, la carga va a ser considerada como un único flujo en vez de considerar cada pieza individualmente. La línea de tratamiento completa consta de un sistema de carga, un horno de austenizado, un tanque de temple, un sistema de lavado y un horno de revenido, como puede observarse en la figura 3.

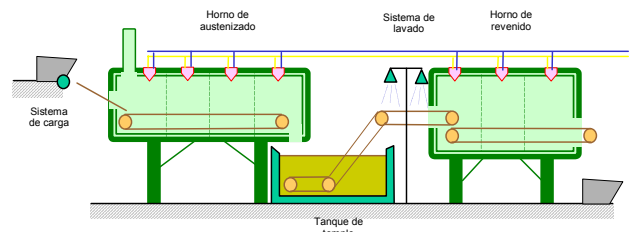


Figura 3: Línea de tratamiento en caliente

El sistema de carga introduce las piezas dentro del horno de austenizado de manera uniforme y continua, situándolas en la cinta del horno. Como resultado del

proceso de austenizado, el material adquiere las características necesarias de dureza. Después de dejar el horno, las piezas caen en el tanque de temple, donde son rápidamente enfriadas en un baño de aceite. Después van al sistema de lavado, que tiene unas duchas de agua para limpiar la superficie de las piezas de aceite. Finalmente las piezas son situadas en el horno de revenido, donde adquieren las características necesarias de ductilidad.

En este artículo se presenta el diseño jerárquico del sistema de carga y del horno de austenizado. Los otros subsistemas de la HTL se diseñarán usando los FBCs y componentes genéricos creados en estos dos subsistemas, o creando unos nuevos cuando sea necesario, siguiendo la misma estructura de diseño y mirando por la reutilización de los nuevos elementos.

Sistema de carga: El sistema de carga esta compuesto de diferentes partes: el elevador, los vibradores, el sistema de pesado y el economizador (ver figura 4). Todos estos subsistemas están compuestos únicamente de elementos mecánicos.

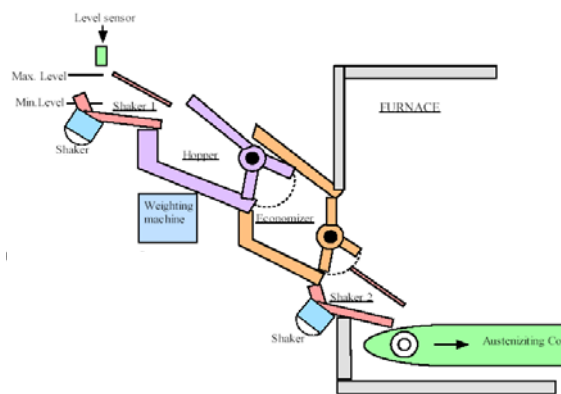


Figura 4: Sistema de carga

El sistema de carga opera de la siguiente forma: El elevador sube un cestón lleno de piezas y lo vacía dentro de una tolva. Esta tolva alimenta a través de un vibrador al sistema de pesado, el cual a su vez conduce la cantidad deseada de carga hacia el economizador. El economizador es un zona situado entre dos puertas cuyas aperturas y cierres están sincronizados de tal forma que ambas puertas nunca pueden estar abiertas al mismo tiempo. Esto es necesario para evitar pérdidas de calor a la entrada del horno. A continuación un segundo vibrador extiende la carga homogéneamente y la coloca en la cinta del horno de austenizado.

Horno de austenizado: Está compuesto por la cinta y el horno de austenizado en si, que a su vez se divide en cuatro zonas con dos quemadores cada una, como puede observarse en la figura 5.

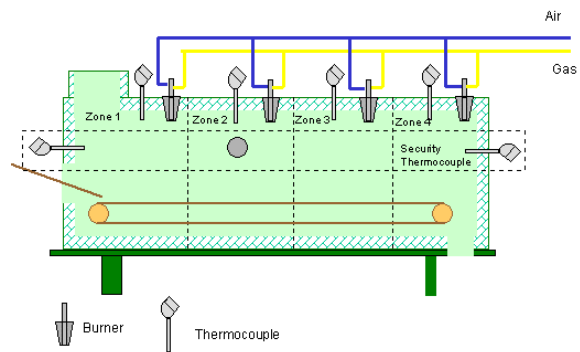


Figura 5: Horno de austenizado

La regulación de temperatura se realiza de forma independiente en cada una de las cuatro zonas, aunque el horno en si está formado realmente por un único volumen. La temperatura interior del horno tiene que estar alrededor de los 850 °C. Cada zona tiene un termopar para medir la temperatura, un termopar de seguridad y dos quemadores donde se realiza el proceso de combustión por medio de una mezcla de gas y aire. También hay una chimenea para la evacuación de los humos del generados en la combustión.

El otro subsistema del horno es un elemento mecánico, la cinta transportadora, que lleva las piezas a lo largo del horno. Regulando la velocidad de la cinta se consigue ajustar la el tiempo de permanencia de la carga del horno, en función del tratamiento térmico requerido.

El modelo tendrá que tener en cuenta todos los diferentes estados de operación que puede tener el horno: vacío con y sin calor intercambiado con las paredes, entrando carga, lleno de carga y saliendo carga. Habrá diferentes modelos para representar cada una de estas situaciones.

4 DISEÑO MODULAR DEL MODELO DE LA LÍNEA DE TRATAMIENTO EN CALIENTE

4.1 SISTEMA DE CARGA

El diseño jerárquico del sistema de carga con los nombres de los distintos componentes se muestra en la figura 6. Como ejemplo, el componente genérico utilizado para modelar el vibrador 1 se muestra en la figura 7. Los elementos utilizados para modelar los subsistemas mecánicos presentes en el sistema de carga son básicamente operadores lógicos (AND, OR, NOT, ...), conmutadores, integradores, saturadores, triggers, etc. La conexión entre los diferentes subsistemas se consigue por medio de variables de flujo de carga que hacen que la carga

vaya de un elemento mecánico al siguiente, hasta que llega a la cinta del horno de austenizado.

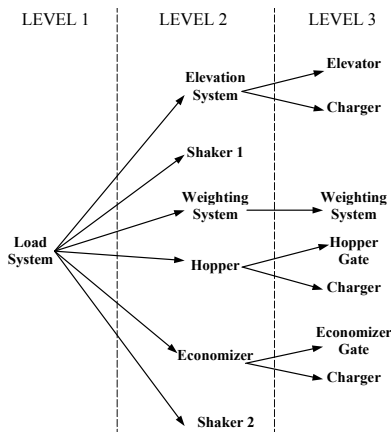


Figura 6: Diseño jerárquico del modelo de carga

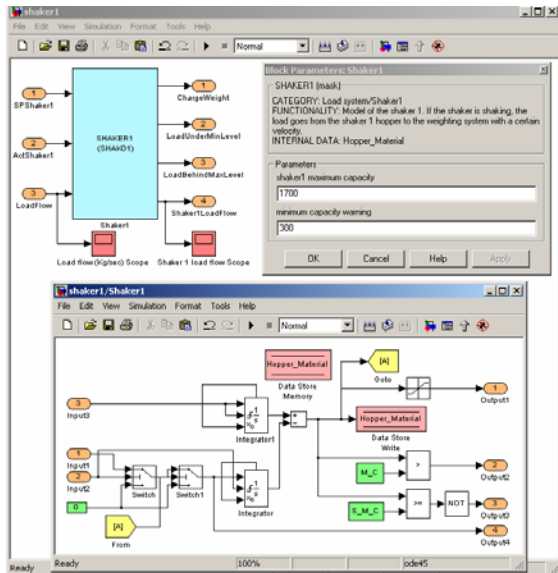


Figura 7: Componente genérico para el vibrador 1

4.2 HORNO DE AUSTENIZADO

El procesado continuo de la línea se va a modelar distinguiendo entre diferentes modos de operación que representan a los diferentes posibles estados del horno. Estos estados estarán presentes tan sólo en la primera zona. El resto de las zonas tan sólo tendrán estado de vacío, debido a que la carga ya alcanza la temperatura de referencia en la primera zona [14], y después ya no absorbe más calor [1].

El componente genérico utilizado para modelar la evolución de la temperatura se construye usando la misma filosofía para todos los estados. El fluido interior del horno se va a suponer que es un elemento cuya temperatura varía uniformemente a lo largo del tiempo en todos sus puntos. Con esta suposición, la variación de la temperatura de este fluido se puede

representar por medio de la ecuación 1 y el modelo en Simulink de la figura 8.

$$c \cdot m \cdot \frac{dT_{gas}}{dt} = q_{combustion} - q_{lost} = q_{gas} \quad (1)$$

- c: calor específico a volumen constante (KJ/kg°C)
- m: masa de gas (kg)
- Tgas: temperatura del gas (°C)
- qgas: diferencia entre la cantidad de calor que entra y sale del elemento de gas (KJ/h)

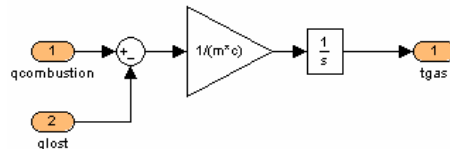


Figure 8: Modelo inicial del horno de austenizado

El balance de energía aplicado se basa en el diagrama de la figura 9 [5]. En esta figura se puede observar que la única cantidad de calor que entra en el elemento de gas proviene del calor generado durante el proceso de combustión. Se ha considerado la cantidad de calor que entra al elemento de gas como positiva y la que sale como negativa. Además, como el modelo está compuesto por cuatro zonas, habrá una cierta cantidad de calor que irá de una zona a las otras, pero más o menos se compensa y puede ser despreciada. Por tanto, ahora tan solo queda añadir al modelo los componentes genéricos de nivel 3 correspondientes a la combustión, el calor a la carga, el calor perdido por los humos y el calor perdido y absorbido por las paredes.

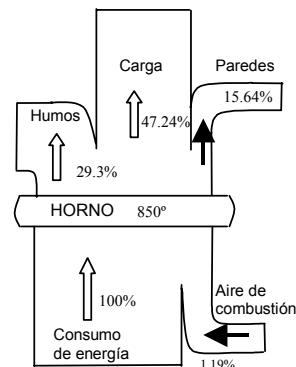


Figura 9: Diagrama del balance de energía

Como ya se ha dicho, los modelos para los diferentes estados y zonas tienen la misma estructura pero con pequeños cambios. Así por ejemplo, el componente genérico de nivel 2 para el estado de vacío no tiene el componente genérico de nivel 3 del calor a la carga, porque en vacío no habría calor absorbido por la carga. Además, este componente genérico del calor

absorbido por la carga será diferente según estemos en el estado en el que está entrando carga, en el que el horno está lleno o en el que la carga está saliendo. De la misma forma, el componente genérico de nivel 3 que representa el calor absorbido por las paredes tendrá diferentes parámetros de configuración en los diferentes componentes genéricos de nivel 2 para cada una de las diferentes zonas. Con estos ejemplos queda claro que la reutilización de los diferentes componentes genéricos se usa mucho en todos los niveles del modelo, y puede ahorrar mucho esfuerzo y tiempo. Todos los componentes tendrán el mismo aspecto que el bloque genérico representado en la figura 2.

El diseño jerárquico del horno de austenizado con los nombres de los distintos componentes genéricos de balances de energía puede observarse en la figura 10. Además, hay un componente genérico mecánico para la cinta. La conexión entre los diferentes subsistemas se realiza por medio de un diagrama de estados en Stateflow, que maneja la secuencia en la que se ejecutan los distintos modelos dependiendo del estado de operación del sistema. La transición entre estados se realiza por medio de diferentes condiciones. El diagrama en Stateflow que se presenta en la figura 11 también contiene los estados del horno parado y de operaciones de seguridad a realizar antes de encender los quemadores, como puede ser la purga para renovar el aire de dentro del horno. Pero estos estados tan sólo tienen un tiempo de transición, y no se pueden considerar como componentes genéricos.

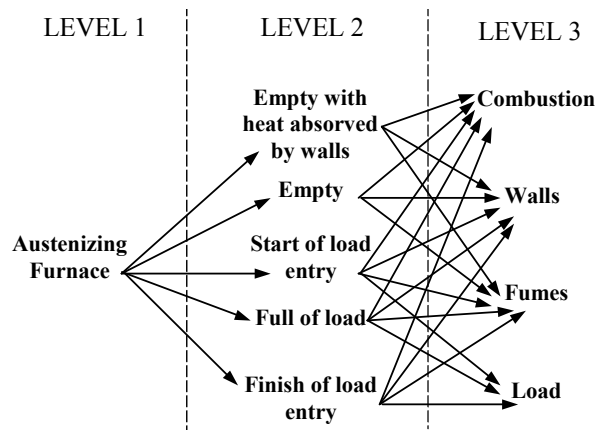


Figura 10: Diseño jerárquico del modelo del horno

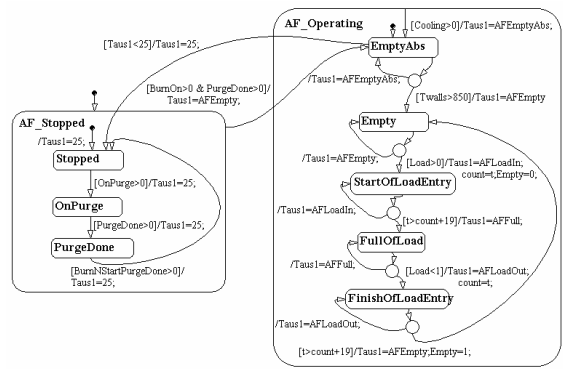


Figura 11: diagrama en Stateflow de la primera zona

5 DISEÑO JERÁRQUICO DEL SISTEMA DE CONTROL DE LA LÍNEA DE TRATAMIENTO EN CALIENTE

Siguiendo la metodología propuesta se ha desarrollado también el diseño jerárquico del sistema de control para el sistema de carga (formado por los FBCs de la figura 12) y para el horno de austenizado (formado por los FBCs de la figura 13).

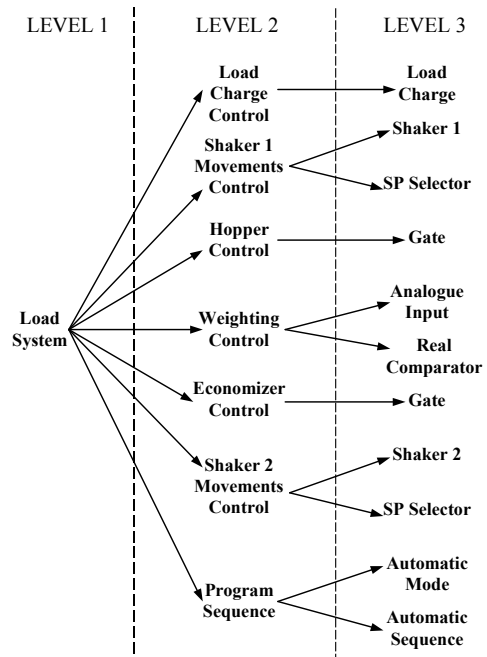


Figura 12: diseño del sistema jerárquico de carga

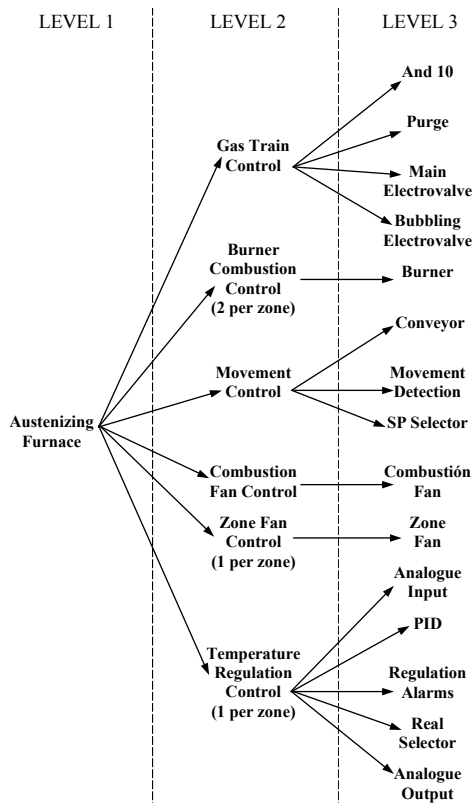


Figura 13: diseño del control jerárquico del horno

El FBC de nivel 1 del sistema de carga está compuesto por siete FBCs de nivel 2. Cada uno de ellos a su vez contiene un grupo de FBCs de nivel 3. La figura 14 nos muestra las características del FBC de nivel 3 para el vibrador 1 (denominado Shaker1). Este FBC podría reutilizarse en otras aplicaciones industriales para controlar los vibradores.

Tipo	Entradas	Salidas
Shaker	<i>Seguridades:</i> Fail in Shaker1 frequency driver	<i>Señalizaciones:</i> ninguna <i>Datos de Proceso:</i> Activate Shaker1
Categoría	<i>Enclavamientos:</i> Shaker1 automatic switch is not shoot	
HeatTreatmentLine / LoadSystem/ ShakerMovementsControl/ Shaker	<i>Comandos de Operador:</i> Shaker1 start button pressed, Shaker1 stop button pressed,	
Etiqueta	Sha01 Shaker	Función
Datos Internos	<i>Datos de Operador:</i> ninguno	shakOProc.actShak:= (((shakOp.shakStart OR shakOProc.actShak) AND NOT shakOp.shakStop AND NOT shakInData.autMan) OR (shakInData.autMan AND shakInData.shakAutStart AND NOT shakInData.shakAutStop)) AND shakSec.freqShakNOK AND shakInter.swShakNot;
Parámetros de Configuración	<i>Datos de Proceso:</i> ninguno <i>Datos Externos:</i> Auto/Man mode Shaker1 start automatic mode, Shaker1 stop automatic mode	
Ninguno		

Figura 14: componente básico funcional del vibrador

El FBC de nivel 1 del horno de austenizado está compuesto por cinco FBCs de nivel 2. Nuevamente cada uno de ellos a su vez contiene un conjunto de FBCs de nivel 3. Algunos de los FBCs de nivel 3 son comunes para el sistema de carga y el horno de

austenizado, por lo que si los FBCs de nivel 3 se implementan como FBs y se almacenan en una librería, después pueden ser reutilizados en ambos subsistemas como instancias de un POU de librería, o incluso se pueden reutilizar en otras aplicaciones si se importa la librería. En la figura 15 se muestra cómo se interconectan los FBCs de nivel 3 para formar el FBC de nivel 2 correspondiente al control del tren de gas (denominado Gas Train Control).

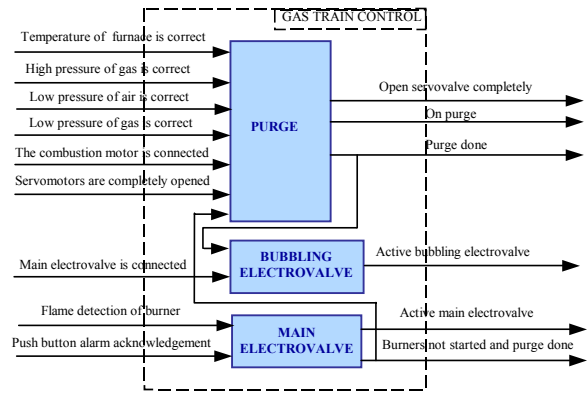


Figura 15: Interconexión de los FBCs del tren de gas

6 PRUEBAS DE VALIDACIÓN DEL SISTEMA DE CONTROL JERÁRQUICO

Se han realizado pruebas de validación unitarias dentro de la herramienta ISaGRAF Enhanced para poder probar así el correcto funcionamiento de los diferentes FBCs de nivel 3. Cada prueba unitaria se ha definido a través de un conjunto de casos de uso en los que se prueba la funcionalidad.

La validación de la respuesta en lazo cerrado se ha conseguido vía co-simulación entre el modelo del proceso (en Simulink) y el sistema de control (en ISaGRAF) Así, las pruebas de integración se han podido realizar a través del entorno de co-simulación en CORBA. Esto permite la simulación del sistema de control discreto en lazo cerrado con un modelo continuo de la planta. En la figura 16 se puede observar el entorno de co-simulación que se utiliza para validar el sistema de carga completo.

La aplicación de co-simulación (que corre en el PC02) se configura a través de un fichero de configuración en donde se define la interacción entre el modelo del proceso y el sistema de control. También permite la inyección de fallos en el sistema de control, al mismo tiempo que en el modelo del proceso para monitorización.

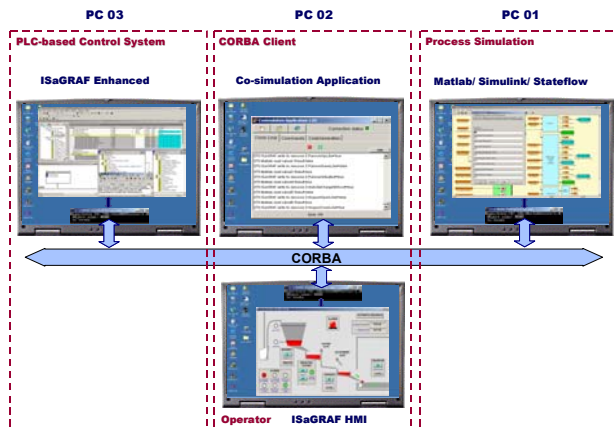


Figura 16: Co-simulación entre Matlab e ISaGRAF

7 CONCLUSIONES

En este artículo se ha propuesto y validado una metodología para el diseño jerárquico tanto de sistemas de control como de modelos de proceso. Esta metodología se ha implementado satisfactoriamente en una HTL. Concretamente se han diseñado diferentes niveles jerárquicos de FBCs para el sistema de control y de componentes genéricos para el modelo del proceso. Los FBCs se han validado por medio de pruebas unitarias y de integración. Las pruebas de integración se han hecho utilizando co-simulación por medio de CORBA entre diferentes partes del sistema de control en ISaGRAF Enhanced y diferentes partes del modelo del proceso en Matlab/Simulink. Quedan probadas las posibilidades de reutilización tanto de los FBCs como de los componentes genéricos a todos los niveles de la jerarquía, así como para otras aplicaciones industriales.

Agradecimientos

Este trabajo ha sido subvencionado por la UE en el marco del proyecto FLEXICON IST-2001-37269.

Referencias

- [1] Astigarraga J., (1994) "Hornos industriales de resistencias: teoría, cálculo y aplicaciones", McGraw-Hill.
- [2] Bradac Z., (2001) "Packet routing problem in hierarchical control system", *13th Int. Conf. on Process Control, Strbské Pleso*, Slovakia.
- [3] Broenink J.F., (1999) "20-SIM software for hierarchical bond-graph/block-diagram models", *Simulation Practice and Theory*, volume 7, pp. 481-492.
- [4] Broenink J.F., (1999) "Object-oriented modelling with bond graphs and Modelica", *Int.*

Conf. on Bond Graph Modeling and Simulation, San Francisco, California.

- [5] CADEM, otros, (1994) "Hornos industriales: fichas técnicas", *Ente Vasco de la Energía*.
- [6] Diaz-Calderon A., Paredis C.J., Khosla P.K., (2000) "Organization and selection of reconfigurable models", *Winter Simulation Conference*, pp. 386–392.
- [7] Freiseisen W., Keber R., Medetz W., Pau P., Stelzmueller D., (2002) "Using Modelica for testing embedded systems", *2nd Int. Modelica Conf.*, Oberpfaffenhofen, Germany, pp. 195-201.
- [8] Guldner J., Utkin V., Bauer U., Bauer R., (1995) "A three-layered hierarchical path control system for mobile robots: algorithms and experiments", *Robotics and Autonomous systems*, volume 14(2-3), pp. 133-147.
- [9] Huang H.M., (1996) "An operator experience with a hierarchical real-time control system (RCS)", *NISTIR 5862, National Institute of Standards and Technology*, Gaithersburg.
- [10] Keung K.W., Ip W.H., Yuen D., (2003) "An intelligent hierarchical workstation control model for FMS", *J. Materials Processing Technology*, volume 139, pp. 134-139.
- [11] Lacabanne B., Benjelloun-Dabaghi Z., Bouskela D., (2003) "Some examples of industrial problems modelling under SCILAB-SCICOS", *22nd IASTED Int. Conf. on Modelling, Ident. and Control*, Innsbruck, Austria.
- [12] Leonessa A., Haddad W., Chellaboina W., (2000) "Nonlinear robust hierarchical control for nonlinear uncertain systems", *Mathematical Problems in Eng.*, volume 5, pp. 499-542.
- [13] Lewis R.W., (1997) "Programming industrial control systems using IEC 1131-3". *IEE Control Engineering series 50*, ISBN-0 85296 950 3.
- [14] Limpens M., Rindt C., Van SteenHoven A., (2002) "Coupled radiation and convection modelling in an industrial furnace", Section Energy Technology, Faculty of Mechanical Engineering, Eindhoven University of Technology.
- [15] Mackulak G.T., Cochran J.K., (1990) "The generic-specific modelling approach: an application of artificial intelligence to simulation", *IIE Integrated Systems Conf. & Society for Integrated Manufacturing Conf.*, San Antonio, pp. 82-87.
- [16] Marcos M., Gangoiti U., Estévez E., Portillo J., Calvo I., (2004) "A CORBA based co-simulation framework for integrating COTS tools", *CONTROLO 2004*, Faro, Portugal.
- [17] McKay W., Monti A., Santi E., Dougal R., (2001) "A co-simulation approach for ACSL-

- based models”, *Huntsville Simulation Conf.*, Huntsville, Alabama.
- [18] Mesarovic M.D., (1970) “Theory of hierarchical, multilevel, systems”, *Mathematics in Science and Engineering*, volume 68, New York: Academic.
- [19] Moor T., Raisch J., Davoren J.M., (2003) “Admissibility criteria for a hierarchical design of hybrid control systems”, *IFAC*, St. Malo, France, 389-294.
- [20] Pappas G., Lafferriere G., Sastry S., (2000) “Hierarchically consistent control systems”, *IEEE Trans. on Automatic Control*, volume 45(6), pp. 1144-1160.
- [21] Ruppel F., Wysor W., (1997) “Guidelines for simulator-based control system testing”, *Technology Transfer*, SEMATECH.
- [22] Tolga I., Kurtulan S., Gören L., (2003) “Supervisory control of a pneumatic system using PLC”, *Int. Conf. on Electrical and Electronics Eng.*
- [23] Valeyev S.S., Vasilyev V.I., Kozyrev S.V., (1999) “Design of active expert control system for chemical technological process”, *Int. Conf. “Interactive Systems: Problems of Man-Machine Interaction”*, Ulianovsk, Russia, pp. 28-29.
- [24] Varaiya P., (1998) “Toward a layered view of control”, *Colloquium of Rocquencourt*, France.
- [25] Wang B., Gong K., Yang D., Li J., (2003) “Fine attitude control by reaction wheels using variable-structure controller”, *Acta Astronautica*, volume 52, pp. 613-618.