

SOFTWARE PARA EL DISEÑO DE CONTROLADORES ROBUSTOS CON FUNCIONALIDAD *HARDWARE IN THE LOOP*

José Lorenzo

Dpto. de Ingeniería de Sistemas y Automática. Escuela Superior de Ingeniería. Universidad de Cádiz.
C./ Chile, 1 – 11003, Cádiz. jose.lorenzo@uca.es

Manuel J. López

Dpto. de Ingeniería de Sistemas y Automática. Escuela Superior de Ingeniería. Universidad de Cádiz.
C./ Chile, 1 – 11003, Cádiz. manueljesus.lopez@uca.es

Luis García

Dpto. de Ingeniería de Sistemas y Automática. Escuela Superior de Ingeniería. Universidad de Cádiz.
C./ Chile, 1 – 11003, Cádiz. luis.garcia@uca.es

Resumen

En este artículo se describe el software integrado para sistemas de control ControlAv. Incorpora técnicas de identificación de sistemas para obtener experimentalmente un modelo del proceso a controlar, realiza el diseño del controlador, su análisis e implementación en tiempo real soft. A través de esta aplicación, el ingeniero de control puede diseñar, de una forma guiada, controladores H_2 , H_∞ y PID; pudiendo repetir el ciclo de diseño hasta obtener un controlador eficiente, que cumpla con unas especificaciones de diseño dadas.

Palabras Clave: simulación en tiempo real, diseño de sistemas de control por computador, control robusto.

1 INTRODUCCIÓN

En los últimos años, a las herramientas software que permiten el diseño, la implementación y la validación de sistemas de control se le demandan un nuevo requisito, consistente en la posibilidad de realizar simulaciones hardware dentro del lazo (HIL, *Hardware in the Loop*). Como es sabido, en la práctica, se utiliza la simulación de un proceso debido a su no disponibilidad, o a que el experimento con el proceso real es demasiado costoso, o por tener una dinámica demasiado lenta. La incorporación de la funcionalidad HIL hace que las simulaciones sean más realistas y consideren aspectos de realización práctica esenciales. La aplicación ControlAv está siendo desarrollada por el grupo GAPSIS de la Universidad de Cádiz, de forma que cumpla con esta especificación.

El artículo, comienza con una introducción de la aplicación ControlAv, continúa con la descripción del algoritmo de fases de operación que tiene implementado; presenta los diferentes estándares de comunicación que se han utilizado para cumplir con la funcionalidad hardware in the loop; por último informa sobre el conjunto de especificaciones que se han definido para la simulación de procesos dentro de la aplicación.

2 ControlAV

El software ControlAV soporta el diseño de controladores robustos de forma online y offline. El ingeniero de control puede diseñar, de una forma guiada, controladores H_2 y el H_∞ , así como el controlador clásico PID. El ingeniero puede repetir el ciclo de diseño hasta obtener un controlador eficiente, que cumpla con unas especificaciones de diseño. Las técnicas de control robusto basados en la teoría de optimización H_∞ se caracterizan por ser aplicables de forma natural a los sistemas multivariados, y por poder incorporar en el ciclo de diseño las cotas en la incertidumbre en el modelo matemático del sistema a controlar.

Para poder realizar el diseño de un controlador H_2 o H_∞ , es necesario que se disponga de un modelo del proceso a controlar. Para obtenerlo de forma experimental se hace uso de técnicas de identificación de sistemas y de las señales recibidas de sensores y enviadas a actuadores que puedan provocar la interacción entre el sistema y el controlador. Una alternativa para el diseño, consiste en la simulación del proceso, de modo que el ciclo de control se implemente en la propia aplicación. Por último, un componente (controlador - proceso simulado) o los dos, pueden implementarse en aplicaciones distintas que se ejecuten en máquinas

diferentes. El prototipo que se está desarrollando permite diferentes modos de operación[7]:

- Control y simulación de proceso en tiempo real, en un mismo equipo.
- Control y simulación en tiempo de cómputo, en el mismo equipo.
- Control y conexión a un proceso real.
- Control y conexión a un proceso simulado, en un equipo remoto.

ControlAv ha sido implementado con la herramienta de desarrollo rápido (RAD) *Builder C++ 5.0 Profesional* [4]. Puede trabajar bajo cualquier plataforma con sistema operativo *Windows*.

La interfaz gráfica con el usuario ha sido cuidadosamente elaborada para que sea fácil, cómoda y representativa, de forma que el operador pueda ver, en soft-real time, la evolución de las variables controladas y del proceso.

Esta aplicación pretende ser una herramienta útil para el diseño de controladores, orientada tanto al mundo profesional como al académico.

2 ALGORITMO DE OPERACIÓN

El ciclo de diseño general de sistemas de control[2], se puede observar en la figura 1, y en la figura 2 el algoritmo de diseño utilizado en *ControlAv*.

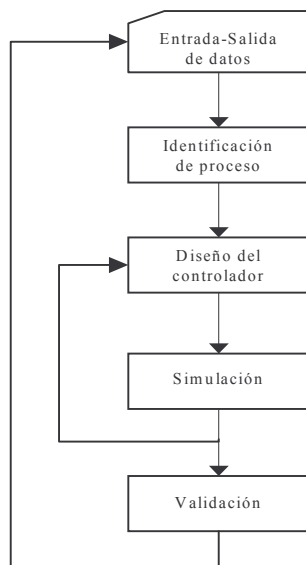


Figura 1: Ciclo convencional de diseño control

Debido a las especificaciones definidas en el software, y tomando como plantilla el ciclo de diseño de controladores genérico, se ha definido un nuevo esquema de ciclo de diseño para el nuevo software:

Mediante un sistema guiado basado en niveles, la aplicación dirige al ingeniero de control en el diseño del controlador, de tal forma, que habilita el acceso a las distintas operaciones conforme avanza en las diferentes fases del diseño del controlador.

El usuario, comienza definiendo las variables del sistema de control así como los valores de saturación. Tras ello, decide sobre el método de interacción de la aplicación de diseño con el proceso. Si el proceso es real, el usuario deberá de establecer los parámetros para llevar a cabo la comunicación de entrada/salida de datos:

- Especificación del estándar de comunicación.
- Canales de comunicación de cada una de las variables a través de los cuales se opera con los sensores y actuadores.

El proceso puede ser simulado mediante un equipo remoto, éste puede recibir, procesar, simular y generar los datos, usando estándares idénticos de comunicación industrial.

Por otro lado, el ingeniero puede indicar a la aplicación que sea esta, quien ejecute el lazo de control; de forma que simulará tanto al controlador que se desea diseñar como al proceso. Para ello se han de establecer los parámetros del modelo del sistema.

En la siguiente fase, el usuario debe de decidir si desea operar con la aplicación, para diseñar un controlador, o si por otro lado desea pasar directamente a controlar el proceso definido en la fase anterior. En el caso de decidir diseñar un controlador, podrá elegir entre el controlador clásico PID y controlador robusto. Si el usuario decide realizar control, podrá escoger entre los siguientes tipos de controladores[10]:

- Lazo Abierto: Permite generar señales de prueba al proceso, para caracterizar su comportamiento (acción necesaria para la fase de identificación).
- PID: Ampliamente utilizado en la industria
- Controlador Genérico: Caracterizado por no disponer de parámetros de ajuste. Es la realización final del controlador en tiempo discreto.
- Robusto: Permite diseñar controladores H_2 y H_∞ .

Independientemente del controlador elegido, se deberán establecer los parámetros asociados para el correcto funcionamiento. Una vez realizada tal operación, el usuario podrá activar el ciclo de control

y comprobar insitu la evolución del sistema de control.

Lo elección más habitual de un ingeniero de control, consiste en el diseño del controlador. Se ofrecen dos posibilidades PID, y controlador robusto.

En este punto, el usuario debe de especificar los diferentes atributos del controlador a diseñar.

Posteriormente, el operador puede optar por ejecutar la función de identificación, con el fin de obtener el modelo nominal completo, del proceso.

A continuación, se accede a la fase de presintonía. Esta se caracteriza por calcular los parámetros de operación del controlador a partir de una operación de identificación, cuyo resultado es el cálculo de un modelo más simple del proceso. Dicho modelo, puede ser obtenido por identificación, bien directamente a partir del proceso real, bien a partir del modelo nominal completo obtenido en la fase de identificación.

Modos de operación:

- Tanto la identificación como la presintonía son ejecutados en tiempo real si el proceso está enlazado con equipo mediante sistemas de comunicación.
- La fase de presintonía se procesa en tiempo de cómputo, si el usuario decide realizar los cálculos a partir del modelo conseguido en la fase de identificación.
- Si el proceso es simulado, las operaciones de identificación y presintonía se realizan en tiempo de cómputo u offline.

Una vez diseñado el controlador, la aplicación, permite al usuario lanzar el ciclo de control en tiempo real, ver el comportamiento del proceso y manipular los parámetros de ajuste así como la señal deseada o set point.

La fase de validación consiste en un test del ciclo de control y el posterior cálculo de indicadores de la respuesta temporal y en frecuencia. Si el usuario ha ejecutado una prueba de control, la aplicación permitirá a calcular los parámetros con respecto a dicha prueba. En caso contrario el usuario puede establecer los parámetros del test, y de forma offline, calcular los indicadores de validación.

Una vez validado el controlador, el ingeniero de control puede calcular el controlador genérico a partir del controlador diseñado. Este puede ser almacenado y posteriormente cargado en la propia aplicación, para realizar operaciones de control. Debido a que es sensiblemente diferente al de origen, la aplicación

ofrece la posibilidad de realizar un test de validación del mismo.

El usuario puede retroceder, a cualquier punto anterior a la fase en la que se encuentre en un momento dado, si bien la aplicación obliga al usuario volver a realizar las operaciones subsiguientes, como medida de seguridad.

2 ESTÁNDARES DE COMUNICACIÓN

Los sistemas de control de procesos están continuamente evolucionando, y la tendencia general consiste en la utilización de estándar abiertos para establecer la comunicación entre controladores y procesos [1]. Un sistema abierto permite al usuario unir en una misma aplicación los componentes hardware y software de distintos fabricantes, de forma que impone pocas restricciones de diseño al usuario y facilita una amplia gama de opciones.

Los estándares de comunicación que se han adoptado en la aplicación ControlAv para poder establecer el hardware dentro del lazo (HIL), han sido los siguientes:

- RS232.
- Intercambio de Datos Dinámicos en Red (NetDDE).
- OLE para Control de Procesos (OPC).

La aplicación además, permite operar directamente con tarjetas de comunicación de National Instruments, ampliamente utilizadas en la industria.

3.1 RS232

La normativa más extendida para la transmisión de señales digitales, en modo serie, es la definida por la *Unión Internacional de Telecomunicaciones* (UIT-T) en su recomendación V.24 (lista de definiciones de circuitos de intercambio entre equipos terminales de datos (EDT) y equipos terminales del circuito de datos (ETCD), que tiene su equivalente en la RS-232 en el año 1991.

Se caracteriza por ser una interfaz lenta con ratios de velocidad que no superan los 20 kbytes/seg. RS232 trabaja sobre todo en aplicaciones de laboratorio donde no se requiere grandes velocidades.

En la fase de implementación de la aplicación *ControlAv*, se ha desarrollado una librería propia basada en la llamada a las funciones estándar del API Win32. El tipo de dato base, a partir de la cual se

implementa el interfaz RS232 es la estructura definida por Windows llamada DCB (*Device Block Control*).

3.2 Intercambio de Datos Dinámicos en Red (NetDDE)

La tecnología DDE (*Dynamic Data Exchange*) fue diseñada como un mecanismo simple para la comunicación entre aplicaciones tales como procesadores de texto y hojas de cálculo.

DDE consiste en un esquema de transferencia de información creado por Microsoft Windows basado en el uso de mensajes del tipo *WM_DDE_xxx*.

NetDDE es una extensión de DDE, con el objetivo de poder realizar transferencias de información sobre una red que trabaja con el protocolo *NetBEUI*. La primera generación de sistemas HMI (*Human Machine Interface*) utilizaba esta tecnología como un mecanismo de compartición de datos entre PLC (*Programmable Logic Controller*) y aplicaciones HMI (*Human Machine Interface*).

Builder C++, provee diferentes componentes para desarrollar DDE en modo asíncrono [3], lo cual nos evita crear una librería básica que invoque a las funciones del API de Windows, tal como se ha realizado en la interface RS232.

Las pautas a seguir para conseguir la comunicación NetDDE, entre dos equipos son las siguientes:

- Comunicación vía red con los protocolos *NetBEUI* y TCP/IP activados.
- Ejecución del agente DDE, llamado NETDDE.EXE.
- En la máquina servidor, debe de ser creado una compartición DDE, esta se puede establecer mediante la utilidad *DDESHARE.EXE*.
- En la aplicación cliente, se debe de indicar el pathname del host de la forma \\Nombre-máquina\NDDE\$. Se deberá de especificar el nombre del Tema, el cual consiste en el nombre del servicio compartido, y el nombre del Campo.

3.3 OLE para Control de Procesos (OPC)

OPC es un estándar abierto para compartir datos entre dispositivos de campo y aplicaciones de ordenador basado en OLE (*Object Link Embedded*) de Microsoft. Permite a las aplicaciones leer y escribir valores de proceso y que los datos sean compartidos de forma fácil en una red de ordenadores conectados mediante la red Ethernet.

Está diseñado para permitir al las aplicaciones cliente, acceder a los datos de una planta de forma consistente.

Extensamente aceptado por la industria ofrece los siguientes beneficios:

- Los fabricantes de hardware sólo tienen que desarrollar un conjunto de componentes personalizadas para usar en sus aplicaciones.
- Los desarrolladores de software no tienen que reescribir los *drivers* ante cualquier nueva característica o nuevo hardware.
- Se evitan conflictos cuando dos aplicaciones desean acceder simultáneamente al mismo dato.

OPC pretende desarrollar un conjunto de especificaciones para la industria con el objetivo de proporcionar una vía común para que las aplicaciones puedan acceder a fuentes de datos como son dispositivos o base de datos, de esta forma un sistema de integración en un entorno de computadoras heterogéneo puede llegar a ser simple.

El Sistema Operativo requerido por la tecnología es Windows 95, Windows NT o superiores. Aunque ya existen versiones creadas por terceras empresas para el desarrollo de software OPC para Unix.

Los desarrolladores de Borland se hicieron eco de esta tecnología, de modo que en el año 1999 la crearon compatible con Delphi versión 3,4 y 5. En el año 2001 la hicieron compatible con Delphi 6.0. Sin embargo, Delphi y Builder C++ no contiene componentes que nos permita trabajar directamente con dicho estándar. Si queremos trabajar con este estándar, debemos crear nuestros propios componentes pero con la garantía de compatibilidad con respecto a los ficheros escritos en C y las DLL's necesarias para poder operar con las mismas y distribuidas de forma gratuita.

A partir de las especificaciones OPC DA [9], en las cuales se define las interfaces de las operaciones de lectura/escritura tanto síncrona como asíncrona entre aplicaciones, se ha desarrollado una librería que permite a la aplicación operar en modo cliente OPC.

3.4 Tarjetas de adquisición de datos de National Instruments

Uno de los métodos más extendido de adquisición de datos a través de un bus de comunicación, consiste en el uso de tarjetas de adquisición.

Uno de los fabricantes de tarjeta de adquisición de datos más importantes que existen en la actualidad, es National Instruments[8]. Ofrece una amplia gama

de dispositivos de adquisición para diferentes buses de transferencia de datos como AT, PCI, PXI, CardBus, VXI, SCSI y USB entre otros.

Además de la calidad de sus productos, este fabricante nos proporciona de forma libre librerías de desarrollo para poder trabajar con sus tarjetas mediante los principales lenguajes de programación para plataformas Windows, entre ellos Builder C++.

El paquete software proporcionado por National Instruments se denomina NI-DAQ. La aplicación nos permite seleccionar el software de National Instruments con el que deseamos trabajar así como el software guiado para la configuración de los diferentes dispositivos. Dicha librería ofrece software de desarrollo para plataformas Windows 2000/NT/XP/ME/9x.

El conjunto de operaciones que nos permite utilizar en su librería son las siguientes:

- Inicialización y configuración general.
- Calibración de dispositivos.
- Uso de eventos mediante mensajes.
- Grupo de funciones de entrada y salida analógica a diferentes niveles.
- Grupo de funciones de entrada y salida digital.
- Grupo de funciones de propósito general para el tratamiento de Reloj/Contador. (Dependiente del dispositivo).
- Disparador de bus *Real-Time System Integration* (RTSI). (Funciones que conectan y desconectan señales sobre disparadores de línea del bus RTSI).
- Funciones para manejo de módulos SCXI de adaptación de señales .

A partir de las interfaces proporcionadas por NIDAQ, se ha desarrollado una librería de una jerarquía superior para poder operar de forma sencilla con las tarjetas de adquisición de datos de NI.

3.5 Metaclases para los estándares de comunicación

El lenguaje base de Builder C++ 5.0, es C++ estándar, creado por Bjarne Stroustrup en 1983.

C++ es un lenguaje totalmente orientado a objetos, lo cual nos permite desarrollar clases, y hacer uso de la herencia y polimorfismo. A partir de estas cualidades, se ha desarrollado una metaclase de comunicación que permite de una forma transparente acceder a las diferentes librerías de comunicación ubicadas en una jerarquía inferior [6].

Las funciones que se han desarrollado son las siguientes:

- Instanciación de los elementos de comunicación.
- Obtención de los parámetros de configuración de una determinada conexión.
- Establecimiento de la configuración de un determinado canal de comunicación.
- Definición del timeout del canal o enlace.
- Operación de Lectura/Escritura.
- Conexión y desconexión de un canal de comunicación.
- Liberalización de recursos y restauración de los sistemas de comunicación

4 SIMULACIÓN DE PROCESOS

Si asumimos que un proceso o sistema real, y la simulación del mismo son realmente equivalentes, la simulación por computadora ofrece las siguientes ventajas [5]:

- El comportamiento del sistema puede ser observado bajo todas las condiciones ideadas.
- Los resultados de la respuesta del sistema puede ser extrapolada con un sistema simulado para las predicciones que se proponga.
- Ahorro de tiempo y dinero.

Existen una gran variedad de algoritmos para conseguir la simulación de procesos o sistemas.

Para conseguir un grado aceptable de simulación, se ha desarrollado una librería con las siguientes especificaciones:

- Desde el punto de vista estructural el sistema puede ser SISO o MIMO. El abanico de sistemas MIMO se restringe a sistemas 2x2, 2x3, 3x2 y 3x3 (debido a que un sistema de mayor orden, es bastante complejo de controlar por parte del usuario).
- Libre elección entre diferentes métodos de simulación: Euler, Euler modificado, Runge-Kutta 2º orden, Runge-Kutta 4º orden, Runge-Kutta-Butcher orden superior.
- Diferentes modalidades para la definición del sistema:
 - Coeficientes de funciones de transferencia de orden 2.
 - Especificación de todos los coeficientes de una función de transferencia de orden n.

- Utilización de matrices de espacio de estado.
- Recuperación de matrices de espacio de estado almacenadas en ficheros en formato MATLAB.
- Restricciones físicas de las variables controlada y manipulada:
 - Cotas inferior y superior.
 - Incremento máximo y mínimo de las variables por unidad de tiempo.
 - Retardos.
- Posibilidad de simulación de agentes externos:
 - Ruido en las variables controladas.
 - Perturbaciones en el sistema.
- Facilidad de simulación tanto en tiempo real como en tiempo de computación.
- Simulación de sistemas definidos en tiempo continuo y discreto.

Para llevar a cabo su implementación, se ha tenido que desarrollar el siguiente conjunto de librerías:

- Sistema de almacenamiento dinámico de datos.
- Librería básica de matrices.
- Librería de manejo de funciones de transferencia.
- Librería para simulación de señales de ruido y perturbación.

A partir de las librerías de comunicación y de simulación, se pueden desarrollar aplicaciones stand-alone que funcionen en distintos equipos bien simulando a un proceso, bien simulando un controlador genérico.

2.6 CONCLUSIONES

Las simulaciones con *hardware in the loop* proporcionan una herramienta eficiente para desarrollar nuevas funciones de control, para analizar el software y el hardware en ingeniería de control.

Este artículo presenta una aplicación que permite diseñar controladores robustos y controladores PID con simulaciones HIL.

El texto pone de relieve aquellos aspectos necesarios para la implementación de software relacionado con las tecnologías de adquisición de datos, simulación de procesos, diseño de controladores y restricciones de tiempo real soft.

La filosofía de la aplicación obedece a la programación orientada a objetos, siguiendo las

directrices de las normas de calidad en la ingeniería del software.

Agradecimientos

Este trabajo está subvencionado en parte por el C.I.C.Y.T, proyecto DPI2002-02995.

Referencias

- [1] Åström K.J., B. Wittenmark, *Computer-Controlled Systems. Theory and design*, Prentice Hall (1997).
- [2] Ayza, Jordi, "Software de adquisición, supervisión y control. Una evolución permanente". *Automática e instrumentación* n° 299 (1999)pp 117-131.
- [3] Bazi V., Moore M.L., Passino K.M., Shackleford W.P., Proctor F.H., Albus J.S., John Wiley & Sons, Inc. (2001).
- [4] Bennet, Stuartt. "*Real Time Computer Control. An Introduction*". Prentice Hall 1998.
- [5] Burns A. Wellings A. *Sistemas de tiempo real y lenguajes de programación*. Addison Wesley (2003)
- [6] Charte, Francisco. "*Programación con Builder C++*". Anaya multimedia 1999.
- [7] Inprise Corporation "*Borland Builder C++ 5.0, Microsoft Win32 reference*". California U.S.A. 2000.
- [8] Isermann R., Shaffnit J., Sinsel S., "*Hardware in the loop simulation for the desing and testing of engine-control systems*". *Control Engineering Practice* 7 (1999) pp. 643-653.
- [9] Johnson R., Foote B.. "*Designing reusable classes*". *J. Object-Oriented Programming*, vol. 1, no.2, pp 22-35, June/July 1998.
- [10] Koga M. et al . "*An integrated environment for the desing and real-time implementacion of control systems*". *Control Engineering Practice* 7 (1998) pp. 1287-1293.
- [11] National Instruments: www.ni.com.
- [12] OPC Foundation : www.opcfoundation.org.

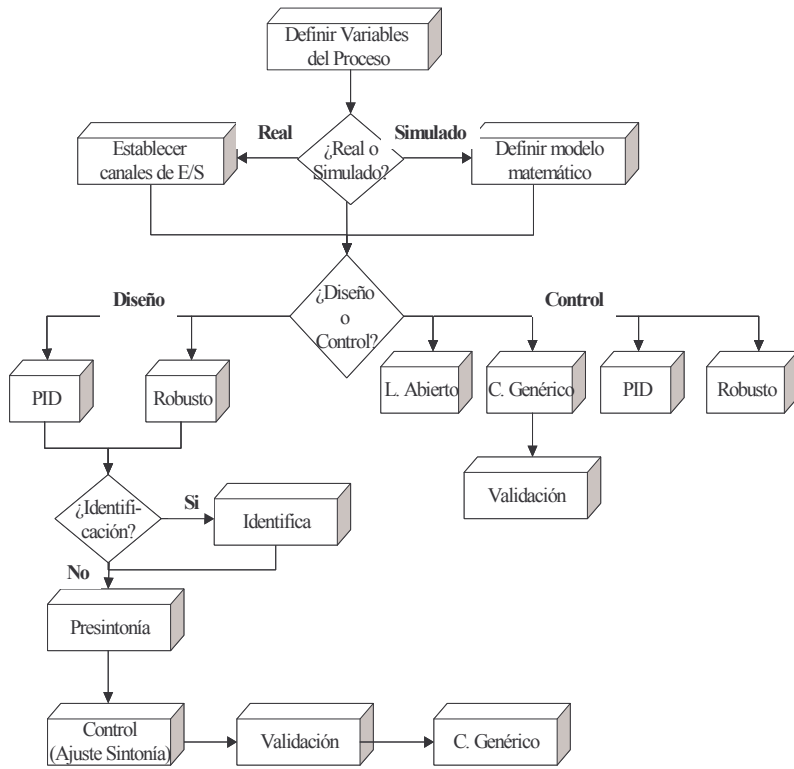


Figura 2. Algoritmo de diseño utilizado en ControlAv