

UNIVERSIDAD DE SALAMANCA
FACULTAD DE CIENCIAS
DEPARTAMENTO DE FÍSICA APLICADA



**ESTUDIO, SIMULACIÓN Y CONTROL DE UN
PROCESO DE NIVEL EMPLEANDO
TÉCNICAS DE INTELIGENCIA ARTIFICIAL**

Ángel Fco. Zazo Rodríguez

SALAMANCA, MAYO 1994

D. ELADIO SANZ GARCÍA, Catedrático de Universidad del Área de Ingeniería de Sistemas y Automática de la Facultad de Ciencias de la Universidad de Salamanca,

CERTIFICA:

que el trabajo titulado "ESTUDIO, SIMULACIÓN Y CONTROL DE UN PROCESO DE NIVEL EMPLEANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL", ha sido realizado bajo su dirección por D. Ángel Francisco Zazo Rodríguez, y constituye la memoria que presenta para optar al Grado de Salamanca en la Facultad de Ciencias de esta Universidad.

Salamanca, 4 de Mayo de 1994.

Fdo. Eladio Sanz García.

Deseo expresar mi más sincero agradecimiento a todas las personas que, de un modo u otro, han colaborado en la realización de este trabajo. En particular a mi director, Dr. D. Eladio Sanz García, a los miembros de las Áreas de Ingeniería de Sistemas y Automática, y Lenguajes y Sistemas Informáticos por sus aportaciones técnicas; y también a los demás componentes del Departamento de Física Aplicada de esta Facultad por su ayuda y su apoyo. Y muy especialmente a mi familia y a Duvi, por su paciencia y confianza.

ÍNDICE

ÍNDICE	i
PRÓLOGO	iii
INTRODUCCIÓN	1
CONTROL DE PROCESOS.....	2
CONTROL INTELIGENTE DE PROCESOS.....	5
Sistemas Expertos.....	9
Lógica Difusa: razonar con incertidumbre.....	13
SIMULACIÓN.....	14
DESCRIPCIÓN DEL PROCESO. ELEMENTOS Y DISPOSITIVOS	17
DESCRIPCIÓN DEL PROCESO.....	17
DESCRIPCIÓN DE LOS ELEMENTOS Y DISPOSITIVOS.....	18
DEPÓSITO Y TUBERÍAS.....	18
ROTÁMETRO.....	21
CONVERSOR PRESIÓN-INTENSIDAD.....	22
transductor INTENSIDAD-PRESIÓN.....	25
VÁLVULA DE CONTROL.....	29
Proceso.....	40
G2	46
INTRODUCCIÓN.....	46
COMPONENTES DE LA BASE DE CONOCIMIENTO.....	47
Objetos y clases.....	48
Variables y parámetros.....	48
Conexiones y relaciones.....	51
Reglas.....	51
Procedimientos.....	53
Funciones.....	54
Fórmulas.....	54
EL MOTOR DE INFERENCIAS.....	55
EL SIMULADOR DE G2.....	57
OTRAS CARACTERÍSTICAS.....	58
Simulación	60
Representación de la Aplicación.....	60
VARIABLES y PARÁMETROS.....	62
VARIABLES.....	63
PARÁMETROS.....	63
CONEXIONES.....	64
OBJETOS Y CLASES.....	65
EQUIPO.....	65
RED-SUMINISTRO.....	66
Válvula.....	67
VÁLVULA-MANUAL.....	67
VÁLVULA-CONTROL.....	67
ROTÁMETRO.....	68
SENSOR-NIVEL.....	69
TRANDUCTOR-I-P.....	69
Depósito.....	70
REGLAS.....	71
RED-SUMINISTRO.....	71
VÁLVULA-MANUAL.....	72
ROTÁMETRO.....	73
SENSOR-NIVEL.....	73
TRANDUCTOR-I-P.....	73

DEPÓSITO.....	73
VALVULA-CONTROL.....	74
Fórmulas de simulación.....	76
VERIFICACIÓN de la simulación.....	77
control borroso.....	81
introducción.....	81
Lógicas y lógica borrosa.....	82
Lógica borrosa.....	85
Control borroso directo.....	88
CONTROLADOR PID DIGITAL.....	88
Controlador borroso.....	97
CONCLUSIONES.....	114
APÉNDICE I.....	115
APÉNDICE II.....	117
BIBLIOGRAFÍA.....	118

PRÓLOGO

Entre los proyectos que está llevando a cabo el Área de Ingeniería de Sistemas y Automática del Departamento de Física Aplicada de esta Universidad, se encuentra el titulado "Estudio, Desarrollo y Aplicación de un Sistema Experto en Tiempo Real", enmarcado en el Programa Nacional de Tecnologías Avanzadas de la Producción, de la Comisión Interministerial de Ciencia y Tecnología (CICYT), y con el que se pretende ampliar el campo de investigación desarrollado durante largo tiempo en dicho área en el control de procesos, tanto en su vertiente teórica como práctica.

El primer objetivo de este proyecto trata de obtener las ideas necesarias para el desarrollo de las nuevas tecnologías de control, dentro del campo de la Inteligencia Artificial (IA), y su aplicación posterior al control de procesos complejos, de difícil estructuración en su conocimiento y con multitud de variaciones en su comportamiento. De los resultados de simulación obtenidos y de su utilización a escala de laboratorio, dependerá la evolución de una aplicación definitiva en algún entorno industrial.

Dentro de esta línea, el trabajo que se expone a continuación es un primer diseño de un sistema de control "inteligente", en el que se aborda, a través de la simulación, la inclusión de estas técnicas sobre un proceso industrial a escala, existente en el laboratorio de prácticas del área, como es el control de un proceso de nivel de líquido.

Después de presentar unas nociones previas de teoría de control de procesos, en el capítulo primero se hace una introducción al concepto de control inteligente, analizando la incorporación de técnicas de IA que posibiliten construir la base de una teoría de control más avanzada. En particular, se hace hincapié en los conceptos de sistema experto y lógica borrosa. También se lleva a cabo un estudio breve de algunas nociones de simulación, enfocándolo desde el punto de vista de estas técnicas.

En el capítulo segundo se realiza un análisis pormenorizado del proceso de nivel sobre el que aplicaremos las técnicas antes comentadas. Para ello ha sido necesario, si se quiere realizar una simulación lo más precisa posible, observar cuidadosamente cada elemento y dispositivo de la instalación, analizando tanto sus características estáticas como dinámicas, y determinando las dependencias entre unos y otros. De la realización de una buena simulación se obtendrán resultados aplicables al proceso real.

Este trabajo se ha desarrollado sobre la herramienta de construcción de sistemas expertos más difundida en los entornos industriales más avanzados, G2, de Gensym Co., y que es uno de los elementos principales en el mismo, gracias a sus cualidades. Para tener una noción básica de ésta, se realiza una descripción somera en el capítulo tercero.

Una vez obtenidas las relaciones entre elementos del proceso, y analizada la herramienta que se va a emplear, en el capítulo cuarto se describe profundamente cómo se ha realizado la simulación, analizando los objetos de la aplicación, el tipo de variables, las reglas heurísticas que son lanzadas, y las fórmulas de simulación empleadas. Se finaliza este capítulo con la presentación de resultados obtenidos de la simulación, y su comparación con las del proceso real.

Uno de los objetivos de este trabajo es la realización del control empleando técnicas de Inteligencia Artificial. El capítulo quinto pretende realizar precisamente esto, empleando para ello nociones de lógica borrosa. Por otra parte, se ha implementado también un clásico controlador PID, para comparar su respuesta con el controlador borroso que se describe en este capítulo.

Por último, en el capítulo sexto se presentan las conclusiones del estudio realizado.

1

INTRODUCCIÓN

El control automático ha jugado un papel de vital importancia en el avance de la ingeniería y de la ciencia, de modo que se ha convertido en parte importante e integral de los procesos industriales actuales. Inicialmente, la idea del control estaba basada en la supervisión directa del hombre: la observación de la evolución del proceso a través de sus sentidos, la elección, de acuerdo con su experiencia, del modo de actuación más adecuado, y la realización de las acciones oportunas para conseguir que se comportase del modo deseado. En el momento en que estas tres acciones las realizan dispositivos "no humanos", es cuando se empieza a hablar de control automático.

El desencadenante histórico del control automático fue la revolución industrial. Se considera como primer controlador automático el regulador centrífugo de la velocidad de una máquina de vapor, realizado por James Watt en el siglo XVIII. Desde este rudimentario dispositivo a los sofisticados controladores digitales actuales ha habido distintas etapas.

Se comenzó inicialmente con dispositivos mecánicos de tipo todo-nada o de variación continua, sin tratamiento de señal. Posteriormente aparecieron los reguladores neumáticos PID, que realizan las tres acciones típicas de control: proporcional, integral y derivativa [28]. Más tarde se emplearon dispositivos eléctricos, y después se introdujo la electrónica en el campo del control. Aunque sin llegar a desplazar totalmente a los reguladores neumáticos, los electrónicos consiguieron una mayor precisión y la posibilidad de realizar compensaciones, tratamiento de señales y su envío a distancia con facilidad y garantía.

La aparición del computador y su utilización en control supone un nuevo avance importante. Se introdujo en los años cincuenta para tareas de supervisión, coordinación y monitorización, pero fue en los sesenta cuando empezó a usarse como controlador directo en lazo cerrado, y cuando lo introdujeron la mayor parte de las grandes industrias. Como eran equipos de gran potencia y elevado precio, razones de rentabilidad obligaban a centralizar en un solo computador todas las funciones de control, además de las burocráticas y de la contabilidad.

La utilización en los setenta de los microprocesadores y su gran difusión en los ochenta y noventa hace que, de forma general, los controladores dejen la electrónica analógica para emplear la digital y la informática, haciendo la realización de las operaciones y cálculo mucho más flexibles y rápidos, permitiendo la inclusión de algoritmos complicados, el cambio de unos a otros de forma sencilla, el tratamiento de señales y datos, etc. También cambia la tendencia del control centralizado hacia el distribuido, usándose pequeños computadores para el control local de las distintas partes, unidos todos ellos a uno central de vigilancia y supervisión. Se introducen además otras teorías de control más sofisticadas, con reguladores que varían sus parámetros según el comportamiento del proceso, e incluso se crean lenguajes específicos para aplicaciones de este tipo.

En los últimos años asistimos a una transformación del concepto de controlador, que tanto en diseño como en realización ha pasado de construirse en términos de sistema analógico o digital a concebirse como sistema de información. Bajo esta nueva perspectiva, sensores y actuadores se tratan como periféricos de una arquitectura computacional, conectados a algún elemento de almacenamiento y procesamiento de información. Asimismo, las acciones de recogida de datos se materializan en rutinas software asociadas a líneas de interrupción del procesador. En conjunto resulta un sistema teleinformatizado que realiza una acción de control [45].

Simultáneamente con la teleinformatización de la Ingeniería de Conocimiento, y debido a las mejores prestaciones que continuamente demanda la industria, la Inteligencia Artificial ha desarrollado un trabajo intenso en áreas de mucho interés para aplicaciones de control. Desde un punto de vista teórico, el énfasis de estos trabajos está en la simulación de comportamientos inteligentes; y desde una perspectiva de aplicación, en la realización de agentes autónomos que pueden realizar su actividad en entornos impredecibles, dinámicos y cambiantes. Desde la perspectiva de la Ingeniería de Control, se ha visto en la Inteligencia Artificial la posibilidad de dar solución tecnológica al crecimiento en autonomía, funcionalidad, adaptabilidad e inteligencia a los nuevos controladores teleinformatizados y sacar todo su jugo al enorme potencial de cálculo de las estaciones de trabajo utilizadas para construir las aplicaciones.

CONTROL DE PROCESOS

El propósito que se persigue en control automático es que un proceso cumpla un determinado objetivo, es decir, que una determinada variable del mismo (salida) alcance unos valores deseados. Dependiendo de si para conseguirlo se hace o no que dicha variable tenga un efecto directo sobre la acción de control se hablará, respectivamente, de *sistemas de control en lazo cerrado* o en *lazo abierto*. Los primeros tienen un marcado interés práctico, mientras que los de lazo abierto son poco frecuentes, pues necesitan estar perfectamente caracterizados, tanto en sus parámetros y modelos, como en las señales y posibles perturbaciones a las que puedan estar sometidos.

En la base de un sistema de control en lazo cerrado se encuentra un sistema realimentado. Este concepto engloba tres partes bien diferenciadas: *medida*, *control* y *actuación*. Si se desea controlar una variable del proceso el mecanismo es el siguiente: fijado el valor deseado de ésta en un instante dado (consigna), se interviene en el proceso a través de un actuador (materializado, por ejemplo, en un motor, un cilindro o una válvula), se observa su valor mediante un medidor (termopar, manómetro, contador, etc.) y la diferencia entre este valor y la consigna, conocida como *señal de error*, constituye la entrada al elemento de control. Este elemento está dotado de un algoritmo cuya función es proveer la acción correctora necesaria para que la señal de error se reduzca a cero. De forma esquemática se presenta en la figura siguiente.

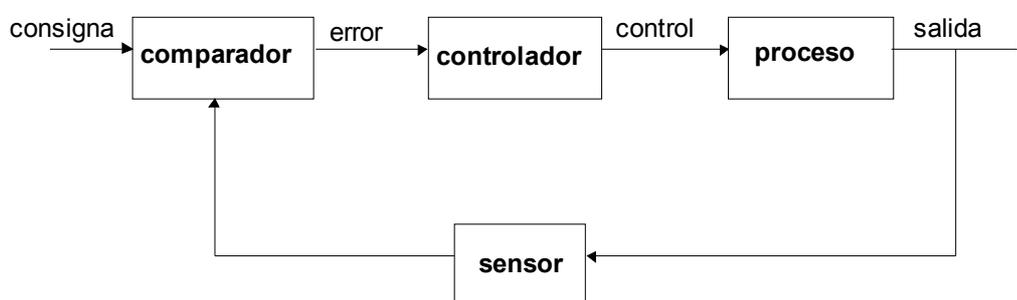


Fig. 1.1 Diagrama de bloques de un sistema de control realimentado.

El mecanismo presentado aquí en su forma elemental, puede ser expandido a un número indeterminado de variables de entrada y salida, las señales pueden ser constantes o variar en el tiempo, puede ser aplicado a un proceso continuo, discontinuo o discreto y tener lugar a distintos niveles, de forma que no sólo un componente, sino también un sistema completo puede ser considerado como un elemento de un lazo de realimentación de nivel jerárquico superior [31].

Dentro de los niveles más bajos de automatización, y dependiendo de la naturaleza del controlador, se tienen dos clases de control bien diferenciados. Por una parte, *control analógico*, si el dispositivo utilizado como controlador es de naturaleza continua, ya sea mecánico, neumático, hidráulico, electrónico, etc. Por otra, *control digital*, cuando se introduce el computador para realizar esta tarea. Hay dos diferencias fundamentales entre ambos tipos de control. La primera es que el computador sustituye a dos elementos analógicos: el comparador y el controlador; y la segunda es que el control digital introduce la necesidad de los conversores de señales, y también su modulación-demodulación, debido a que el computador trabaja con señales digitales y discretas, mientras el proceso normalmente es continuo y analógico, al igual que las señales procedentes del sensor. Es preciso pues, transformar valores analógicos continuos en trenes de pulsos digitalizados y viceversa (figura 1.2).

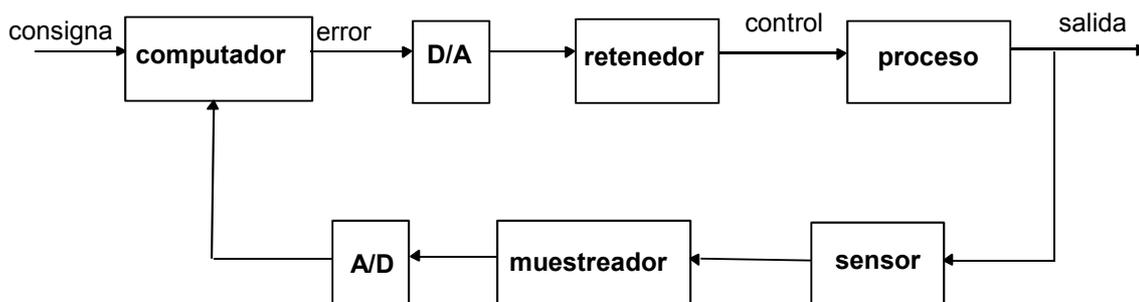


Fig. 1.2 Diagrama de bloques de un sistema de control digital en lazo cerrado.

El grado de automatización de las tareas disminuye a medida que se considera un nivel superior en la jerarquía que se muestra en la figura 1.3. En el nivel inferior se consideran problemas típicos de control en bucle cerrado mediante los cuales se pretende regular variables del proceso manteniéndolas en valores de consigna aunque se presenten perturbaciones externas. El seguimiento de señales de referencia también es propio de este nivel.

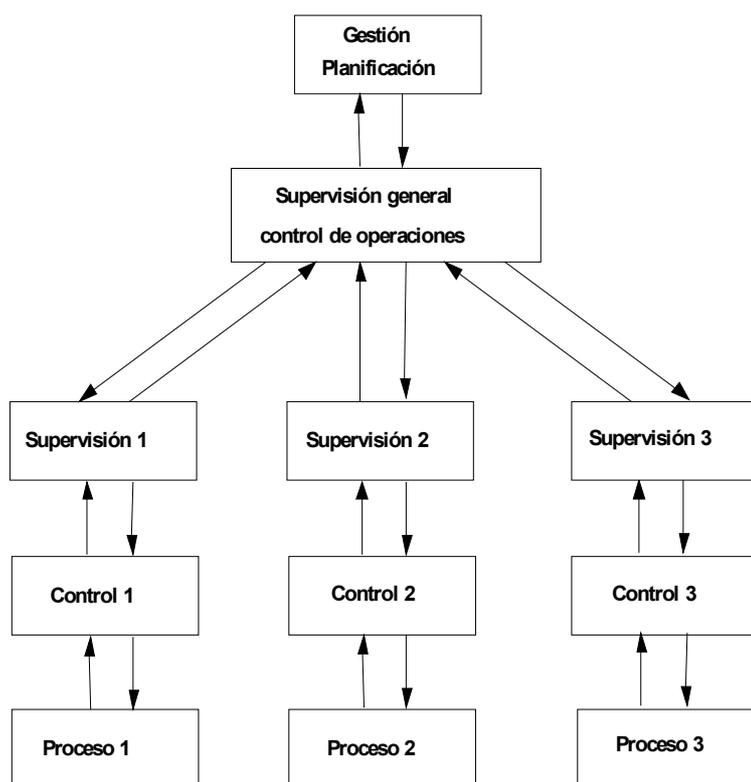


Fig. 1.3 Jerarquía de control.

En el segundo nivel se contempla la supervisión local de los controladores del primer nivel; en el tercero se considera la supervisión global de un sistema en el que pueden existir diversos

controladores locales, y el control de operaciones. Asimismo, se representan niveles superiores que están relacionados con la planificación y la gestión.

Cuando se desciende en esta jerarquía se va hacia de escalas de tiempo más rápidas, y por tanto más restrictivas, hasta llegar a la que permite representar el comportamiento dinámico del proceso que se quiere controlar.

Nótese que en la actualidad, además de los criterios de productividad, aumentan las exigencias de seguridad de funcionamiento y calidad, de manera que las tareas de supervisión cobran mayor importancia.

CONTROL INTELIGENTE DE PROCESOS

El control de procesos industriales intenta mejorar continuamente el nivel de prestaciones que ofrece, para aumentar la seguridad, calidad y productividad de los mismos. Entre los servicios que demanda está la toma o ayuda a la toma de decisiones, una mejor identificación entre operador y el sistema de control, robustez, la inclusión de criterios de optimización en los requerimientos de control, etc. Para conseguir estos objetivos muchos autores proponen incluir en las técnicas habituales de control procedimientos de la Inteligencia Artificial, a cuyo conjunto de técnicas denominan *control inteligente de procesos*.

Como se mencionó anteriormente, la utilización de computadores en procesos industriales, además de facilitar la incorporación de estrategias de control automático, permite realizar de forma eficiente tareas de supervisión y tratamiento de datos de un notable interés práctico. Sin embargo, las tareas que involucran capacidad de percepción y experiencia no se prestan a ser automatizadas con las técnicas convencionales. El control inteligente comprende una serie de técnicas con las que se pretenden resolver problemas de control inabordables por los métodos clásicos [14].

Por otra parte, aunque la teoría de control digital se ha venido aplicando satisfactoriamente en numerosos sistemas, y ha servido de fundamento para la definición de la teoría de control moderna, ésta presenta limitaciones que impiden construir la base de una teoría de control más avanzada [4, 43]. Seguidamente se analizan las más importantes.

Modelos matemáticos

En un sistema de control convencional, el diseño de controladores se basa en un conocimiento preciso del proceso, descrito por ecuaciones matemáticas (ecuaciones de estado, función de transferencia, etc.) deducidas de leyes físicas y datos experimentales. Cuando los procesos son complejos o su control requiere de mayores prestaciones, encontrar un algoritmo de control adecuado puede convertirse en un problema matemáticamente intratable. Además existen muchos procesos en los que el operador continua siendo necesario, incluso en el nivel inferior en bucle cerrado. En efecto, aunque en este nivel se han automatizado un gran número de procesos,

utilizando fundamentalmente reguladores convencionales, tales como PID, continúan sin resolverse numerosos problemas de control automático, en especial los relacionados precisamente con la ausencia de modelos fiables que representen convenientemente la dinámica del proceso que se pretende controlar.

Desde hace más de una década diversos investigadores han adoptado una estrategia de diseño de controladores digitales consistente en evitar la elección del modelo matemático del proceso, para centrarse en obtener un modelo del comportamiento del operador o técnico, basado en reglas del tipo *si4entonces*, que incorporen su experiencia [27, 41], y de esta forma intentando modelar las relaciones de entrada4salida del sistema. En este sentido se han desarrollado los denominados *controladores borrosos*, a los que dedicamos un capítulo completo de esta memoria. El esfuerzo se centra en que el controlador modele directamente el proceso de decisiones del operador (figura 1.4).

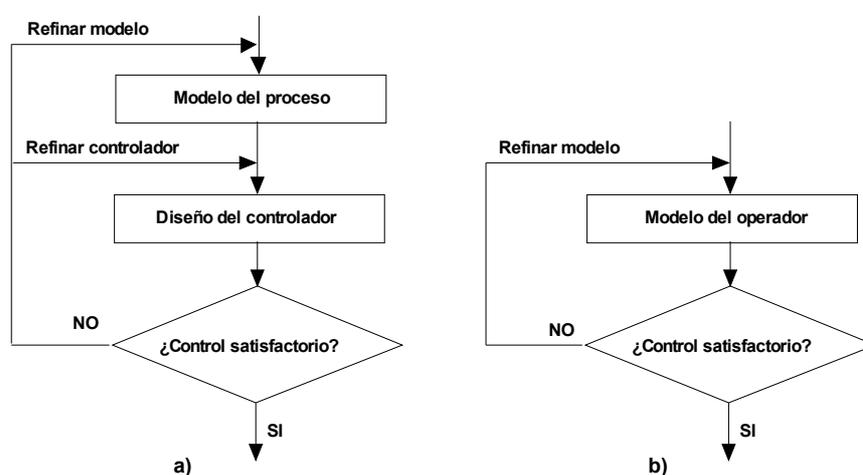


Fig. 1.4 a) Control convencional, b) Control basado en el modelado del operador.

Integración de operaciones de control con otras más avanzadas.

En Ingeniería de Control el concepto de acción de control está muy limitado a una actividad primaria. El diseño de controladores sobre este concepto limita su posibilidad de crecimiento, imposibilitando la integración de otras actividades más avanzadas. Sin embargo, puede hacerse posible con la teleinformatización de los sistemas de control, que facilita la ampliación del concepto de acción de control, tanto en un sentido vertical, posibilitando una organización jerárquica de las posibles acciones de un controlador, empleando para ello la estructura de información común creada para representar la información relativa al sistema y operando de acuerdo con ella para determinar las acciones que deben ejecutarse en cada nivel; como horizontal, para extenderse en el tiempo mediante distintos planes de acción y ejecución.

Rigidez del algoritmo de control.

No hay flexibilidad para la *reprogramación de objetivos*, para la *interacción con operadores*, para *adaptarse a nuevas contingencias*, o para hacer posible la *incorporación de cambios en el*

sistema bajo control. En sistemas tradicionales de control no existe una separación clara entre el conocimiento relativo al problema que se pretende resolver y los procedimientos generales que lo manipulan. Por ejemplo, los procesos incluyen numerosas especificaciones que indican cómo se genera una acción de control o cómo se activa una alarma. Si se pretende cambiar este conocimiento, en función de la experiencia adquirida, es necesario modificar el programa y generar un nuevo código, lo cual dificulta la actualización de este conocimiento, e incluso puede llegar a impedirlo si se pretende su modificación en tiempo real.

Algoritmo y heurística

En la mayoría de los reguladores industriales es necesario programar funciones lógicas auxiliares de gran interés práctico, que en la mayoría de los casos representan un porcentaje del programa mucho mayor que el propio cálculo de la acción de control. En efecto, un típico regulador PID digital realiza las siguientes funciones:

- Selección del modo de operación.
- Lógica para condiciones de funcionamiento anormales.
- Tratamiento de señales: corrección de no linealidades, ajustes de escala, filtrado, etc.
- Comprobación de límites y tratamiento de alarmas.
- Modificación de parámetros del controlador, si fuese necesario.
- Visualización y gestión del teclado.
- Cálculo de la acción PID.

Nótese que tan sólo en el procedimiento de cálculo de la acción PID es posible utilizar los resultados de la teoría de control. La implantación del resto de las funciones se realiza normalmente de forma heurística, generalmente por medio de numerosas sentencias *if4then4else*. Por otra parte, el conocimiento necesario es aún mayor en el diseño e implantación de cualquier aplicación en control de sistemas industriales complejos. La situación es paradójica, pues la utilización práctica de los métodos avanzados de control involucra normalmente la consideración de nuevas funciones de supervisión, que son críticas para la correcta aplicación de estos métodos [27].

En cualquier caso, cabe señalar que se involucran tanto resultados de la teoría de control como numerosas funciones lógicas heurísticas, basadas en la experiencia que se recoge de los ingenieros y operadores de planta, sobre métodos de control y sobre las características particulares del proceso. La carencia de metodologías apropiadas para el desarrollo de la heurística en este sentido con las herramientas clásicas de la teoría de control, ha llevado a proponer la utilización de técnicas de la Inteligencia Artificial, y más concretamente los *sistemas expertos*, en el marco del control de procesos.

Reactividad

La reactividad se entiende como la facultad de atención a los eventos asíncronos que se producen en el entorno del controlador. En sistemas convencionales de control, la reactividad está

garantizada por el propio proceso de diseño. Las rutinas de control realizan la adquisición de datos, los procesan y activan los actuadores en el mismo cuerpo de código. Ejecutando todo el código en un tiempo menor que la mínima separación entre eventos externos se garantiza la atención a todo lo que ocurre en el entorno. El coste, obviamente, es la necesidad de ejecutar el código en su totalidad en cada ciclo de funcionamiento.

Cuando las funciones de control se hacen más numerosas y complejas, y se agrupan en planes de acción, el volumen de código crece de forma que no se puede ejecutar en su totalidad en ciclos de funcionamiento de duración aceptable. La reactividad del sistema no se puede basar, pues, en la ejecución exhaustiva de todos los programas, sino que ha de estar necesariamente asociada a la asignación flexible en tiempo de ejecución de los recursos del controlador a las tareas activadas por los eventos.

Adicionalmente, los análisis temporales utilizados para la satisfacción de los objetivos de control se complican. En aplicaciones convencionales esto se hace de una forma simple. Pero un análisis tan simplificado no es válido cuando el controlador realiza operaciones de distinto nivel de inteligencia agrupados en planes y todas ellas con diferentes constricciones de tiempo asociadas.

Actividades conductistas y cognoscitivas

Cuando observamos en detalle la lista de actividades de un sistema de control, observamos claramente dos tipos diferenciados. Por una parte, actividades que conducen a la modificación del entorno, mediante la regulación de valores de variables; y por otra, actividades de procesamiento de información con un cierto grado de abstracción. Estos dos tipos de actividades son, en cierta medida, calificables de conductistas y cognoscitivas.

El control clásico consta esencialmente de tratamiento de información de forma numérica, no apareciendo en él actividades cognoscitivas de alto nivel de abstracción, como las derivadas de la intuición y previsión. Es por ello que en la mayoría de los casos debe añadirse a este tipo de controladores la supervisión directa del ser humano, que mediante un proceso de toma de decisiones determina las actuaciones que deben realizarse.

Teniendo en cuenta que estas labores involucran la experiencia acumulada por operadores y técnicos, si se pretende automatizarlas total o parcialmente, es necesario emplear técnicas que permitan tratar de forma apropiada este conocimiento heurístico, como las desarrolladas por la IA. En este sentido se han fomentado distintos sistemas de ayuda a la toma de decisiones, ejecutores de planes de emergencia, etc. [33]. En particular, el empleo de sistemas expertos para control y supervisión de procesos, resulta especialmente atractivo en los casos en que es necesario adquirir, procesar, mantener y aplicar un gran volumen de información heurística.

SISTEMAS EXPERTOS

Existen muchas definiciones de lo que es un sistema experto (SE), pero una comúnmente aceptada es "*un sistema computador, programado de tal modo que parece emular la capacidad de toma de decisiones de un experto humano en un dominio dado de un tema*" [38]. Con esta descripción sin más, muchos programas de ordenador podrían legítimamente pretender ser sistemas expertos. Sin embargo, una de las características más significativas de los SE, que facilita la adquisición y mantenimiento del conocimiento, es la separación que establecen entre el propio conocimiento y los datos, relativos al problema que se pretende resolver, y los procedimientos generales que los manipulan (razonamiento, inferencia y mecanismo de control), lo cual tiene importantes ventajas desde el punto de vista del diseño informático, mantenimiento, y transportabilidad de las aplicaciones. Nótese que esta separación no es clara en los sistemas tradicionales, en los cuales los programas y datos están a menudo estrechamente integrados, y, como vimos anteriormente, si se desea ampliar ese conocimiento en función de la experiencia adquirida, es necesario modificar el programa y generar un nuevo código.

En la bibliografía se usan de forma indiscriminada términos como *sistema experto de control* y *sistema de control basado en el conocimiento* [33]. Es preciso realizar una matización de estos términos para aclarar su significado. Un sistema basado en el conocimiento es una aplicación que hace uso de las técnicas de representación del conocimiento. Cuando se emplea para ello el paradigma *motor de inferencias base de conocimiento*, utilizando generalmente el formalismo de reglas obtenidas del conocimiento del experto humano, se dice que es un sistema experto de control.

Debido a que los sistemas basados en el conocimiento representan explícitamente y razonan con el conocimiento suministrado por expertos humanos, presentan las siguientes ventajas [23, 11] :

- Capturar y retener experiencia conseguida durante largos años de ingeniería.
- Ampliar la experiencia que es necesaria para desarrollar satisfactoriamente nuevas tecnologías y diseño de aplicaciones. Como el conocimiento se representa separado del código que lo aplica, las capacidades del sistema pueden extenderse aumentando el conocimiento.
- Facilidades de explicación del sistema, ya que los hechos y procesos de razonamiento empleados por el sistema experto son una imagen de los que emplea el experto humano, determinando de forma sencilla errores o insuficiencias sobre el conocimiento del sistema.
- Construcción de sistemas interactivos de ayuda a las labores de un operador de procesos [27], ofreciendo sistemas que razonen inteligentemente sobre las acciones necesarias en tiempo real.

Un sistema experto está formado fundamentalmente por tres componentes básicos. Uno, la **base de conocimientos**, en la cual puede recogerse tanto la experiencia general sobre ingeniería de control, como el conocimiento específico del dominio de la aplicación; otro, la **base de datos**, donde se va almacenando la información relativa al problema que se resuelve; y finalmente, un

motor de inferencias o mecanismo de control, que mediante técnicas de procesamiento simbólico seleccionará, aplicará y encadenará los elementos de la base de conocimiento pertinentes para la solución del caso concreto que se presente.

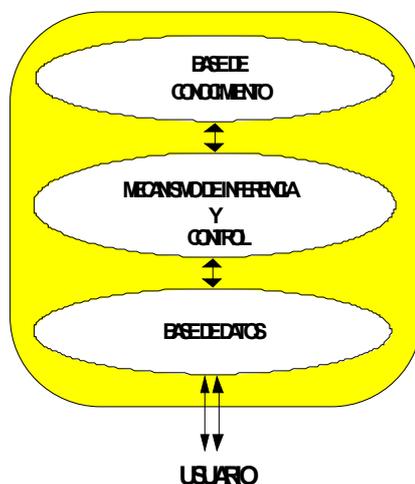


Fig. 1.5 Arquitectura básica de un SE.

En la figura 1.6 se muestra la arquitectura de un sistema experto on-line. Además de los elementos señalados anteriormente, existen dos interfases externas separadas, una para que el experto introduzca y actualice el conocimiento específico del dominio de la aplicación, y otra para la presentación de información al usuario del sistema [38]. También son necesarias interfases de datos, para obtener una descripción del estado del proceso, y poner en ejecución las conclusiones de la inferencia, mediante los procedimientos necesarios para generar las órdenes sobre los actuadores o controladores.

La base de conocimientos es el componente clave de cualquier sistema experto. Existen diversas técnicas para representar el conocimiento [21, 23, 32], pero la más convencional es la utilizada en los llamados sistemas de producción, en los que se emplean representaciones en forma de reglas del tipo: **si (condiciones) entonces (acciones)**. Mediante estas reglas se especifica la necesidad de la presencia de una serie de hechos para que puedan deducirse unas conclusiones. El término *acción* no tiene por qué implicar la generación de una acción externa, sino que puede interpretarse como la deducción de la presencia de otros hechos. Muchos de los sistemas expertos de hoy en día están principalmente basados en reglas.

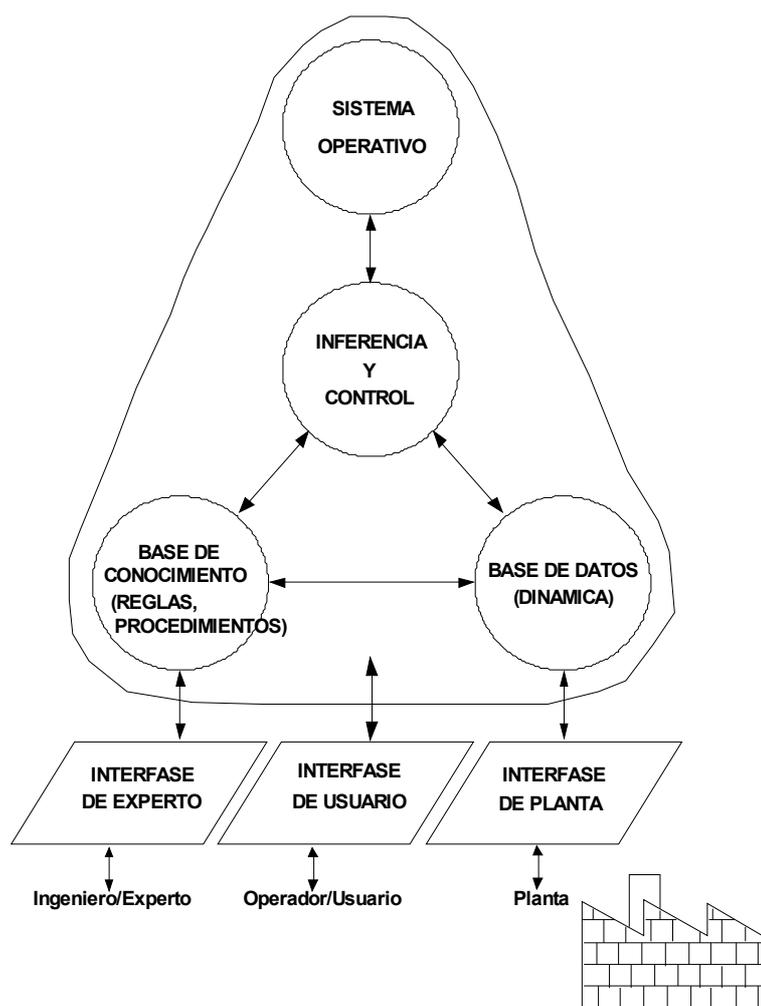


Fig. 1.6 Arquitectura de un Sistema Experto on-line.

Con una regla podemos realizar diferentes objetivos: expresar una definición, identificando el estado de algún componente del sistema en base a ciertos requerimientos; establecer condiciones en las que se deben realizar ciertas acciones; describir relaciones de causa-efecto en el funcionamiento de un mecanismo; asociar hechos en base al conocimiento sobre el dominio, etc. Estas reglas son procesadas por un motor de inferencias, independiente de la base de conocimiento, que trabaja utilizando la base de datos del sistema para inferir, en función de los hechos actuales, nuevos hechos que puedan dar lugar a la generación de acciones de control.

El motor de inferencias funciona básicamente siguiendo un ciclo, en el que en primer lugar se selecciona un conjunto de reglas, obteniendo las aplicables, y después de resolver posibles conflictos, se selecciona una de ellas para ser invocada. El ciclo continua con el disparo de la regla seleccionada, lo cual conduce a la actualización de los hechos, a la activación de acciones si procede, e incluso a la propia modificación de la base de reglas. A continuación se vuelve a iniciar el ciclo, que se repite hasta que no es posible realizar una nueva deducción. El mecanismo aquí presentado es muy parecido en la mayoría de los SE, y en particular el que emplea la herramienta G2 [17].

Ya se ha indicado el interés que presentan los SE cuando se considera el conocimiento heurístico sobre el control de procesos. Este suele representarse en forma declarativa utilizando hechos y reglas. Sin embargo, en control de procesos se utilizan gran cantidad de modelos analíticos convencionales de simulación, tratamiento de señales, estimación de parámetros y generación de actuaciones de control de acuerdo con una cierta ley. Estos modelos representan un comportamiento típicamente procedural. Entonces, para emular la resolución de problemas que utiliza un experto, no es suficiente con representar un conocimiento declarativo separado de los componentes procedurales de la experiencia. El desarrollo de sistemas expertos eficientes en control de procesos obliga a integrar ambas formas de conocimiento.

En este punto conviene recordar la distinción que se establece en el campo de la Inteligencia Artificial entre lo que se denomina conocimiento superficial y conocimiento en profundidad. Esta distinción se presenta también con otros nombres [12]. En cualquier caso, por el primero se entiende el conocimiento desde fuera, sin considerar los mecanismos internos ni la representación de principios físicos básicos, e incluso sin representación explícita de propiedades tales como la causalidad entre sus variables internas, sino en forma de representaciones de condición-acción. En el conocimiento en profundidad se observa el sistema desde dentro, tratando de descubrir las causas originales de su comportamiento, proporcionando la información del nivel inferior, causal, funcional y física del problema. Aunque el modelado en profundidad de sistemas físicos es un campo de investigación muy activo en los últimos años, el conocimiento analítico y las técnicas numéricas siguen siendo las herramientas usuales en la representación del conocimiento sobre los principios físicos de interés en sistemas de control. De esta forma, el conocimiento procedural sobre algoritmos de control, identificación y simulación involucra esencialmente el empleo de técnicas numéricas. En muchas aplicaciones se han integrado ambos métodos, de modo que una regla cuya conclusión sea "activar el control de seguridad" debe desencadenar una llamada a un procedimiento con un método numérico que calcule, en función de las medidas en tiempo real sobre las variables del proceso, la actuación de control correspondiente, y la aplique en el tiempo real deseado.

Debemos considerar además que en la mayoría de las aplicaciones on-line en situaciones dinámicas es necesario poder razonar a lo largo del tiempo. La tarea de adquirir datos con suficiente rapidez y de razonar para llegar a una conclusión en tiempo real (es decir, antes de que los datos hayan cambiado significativamente) es extremadamente difícil. La información adquirida en tiempo real como las conclusiones pueden tener un tiempo limitado de validez. El tiempo de inferencia depende en gran medida de la representación del conocimiento y de las prestaciones de la arquitectura que se emplee. Una solución puede plantearse en la utilización de un enfoque selectivo de la información, emulando el comportamiento del operador humano cuando concentra su atención en un dominio particular de su conocimiento. De esta forma, se emplea lo que algunos autores denominan *metaconocimiento* [26], que mediante *metarreglas* permite concentrar la atención en un solo grupo de reglas. En efecto, si una aplicación posee cientos de reglas, pero en un momento dado sólo se necesitan una docena para resolver una situación específica, se concentraría la inferencia

exclusivamente en ese grupo de reglas, lo cual tiene una gran importancia en cuanto a la rapidez de la inferencia.

Un sistema experto puede ser construido de muchas formas distintas, y puede tener diferentes grados de complejidad y flexibilidad. La diferencia principal entre un sistema experto y un sistema computador convencional radica en la aproximación a la resolución del problema y en la representación e implementación de la solución utilizando un ordenador. En lugar de analizar problemas y programar soluciones específicas en forma de procedimientos, la aproximación de IA parte de un paso anterior y examina cómo son abordados los problemas, y busca para ofrecer la solución de un modo tal que los procesos de razonamiento y conocimiento subyacente se mantienen separados y se explicitan. Utilizando estas técnicas no sólo se comprende más fácilmente la base de resolución del problema y la toma de decisiones, sino que se puede añadir incrementalmente un nuevo conocimiento sin tener que volver a concebir el resto del sistema. Esto está en contraste directo con la aproximación convencional, donde es casi siempre necesario hacer cambios importantes del programa para acomodar nuevas situaciones.

LÓGICA DIFUSA: RAZONAR CON INCERTIDUMBRE

Aunque existe una cierta cantidad de conocimiento sobre el control de procesos que usualmente se considera definido de forma precisa, también existe un volumen considerable del conocimiento heurístico de operadores y técnicos que suele expresarse de forma imprecisa, aunque conteniendo importante información. La incertidumbre en esta información produce incertidumbre en la validez de las conclusiones; por consiguiente, es interesante disponer de las técnicas que permitan propagar la incertidumbre desde las premisas a dichas conclusiones. Los sistemas expertos actuales emplean diversas técnicas para ello, aunque la que más partidarios tiene en control de procesos es la denominada *lógica borrosa* o *difusa* (procedente del término inglés *fuzzy*), que pretende formalizar el razonamiento con incertidumbre. En este sentido, el *control difuso* puede considerarse como un control basado en reglas que utiliza las técnicas difusas para manejar imprecisión, implantando estrategias de control expresadas en términos lingüísticos por los operadores del proceso.

La ventaja fundamental de utilizar precisamente términos lingüísticos, como por ejemplo en la sentencia "la temperatura es alta", está en que permite plantear el problema con la misma terminología con que lo haría un experto humano. De esta forma, se elude buscar la solución a un problema no perfectamente definido por medio de un planteamiento matemático muy exacto, cuando el ser humano es el primero que razona empleando la inexactitud.

Un controlador difuso esencialmente se basa en un conjunto de reglas de la forma:

SI A es grande y B es pequeña ENTONCES C es mediana.

Todo el conjunto de reglas equivale a una función del espacio de entradas en el espacio de salidas. Un aspecto fundamental de este tipo de reglas es el número de niveles de cuantificación de entradas, que corresponde al número de términos lingüísticos utilizados para representar los valores de las mismas. Habitualmente se toman entre 5 y 9 términos, ya que menos no definen bien el problema, y más se salen fuera de los límites que usualmente emplea un ingeniero de procesos. Las etapas de control son las siguientes:

- Conversión de los valores medidos del proceso a su representación difusa.
- Aplicación de las reglas para obtener los valores de las actuaciones.
- Obtención de valores nítidos para las actuaciones a partir de sus valores difusos.

En el capítulo dedicado a los controladores difusos se hace una descripción completa de las técnicas y características fundamentales de éstos, de modo que seguiremos allí su estudio.

SIMULACIÓN

El diseño óptimo de los parámetros que definen un regulador está ligado de forma muy directa al proceso al que va destinado, de modo que el conocimiento del mismo, tanto de sus características estáticas como dinámicas, es imprescindible a la hora de determinar el buen ajuste del regulador. Tres son los métodos utilizados en Ingeniería de Control para caracterizar el comportamiento de un proceso a controlar:

- a) Modelación física. Supone el desarrollo de un modelo continuo a partir de un cuidadoso reconocimiento teórico que represente la situación física del proceso bajo consideración. Por ello es preciso investigar el orden del modelo matemático y las relaciones funcionales entre variables. Los resultados que se consiguen suelen ser casi siempre insatisfactorios: se obtienen modelos de orden alto, difíciles de utilizar, cuyos coeficientes no siempre pueden determinarse con exactitud, y que pueden presentar no linealidades complicadas de caracterizar.
- b) Modelación por respuesta a señales prueba. A partir de un procedimiento experimental se sigue el análisis de la respuesta temporal del sistema a unas señales de prueba normalizadas. La observación de la respuesta transitoria y permanente a las entradas nos permitirá conocer y clasificar el comportamiento del mismo. La información acerca del sistema obtenida de esta manera es muy útil, pues es en general suficiente para la predicción del comportamiento del mismo en cualquier situación.
- c) Modelación estocástica. Es muy parecido al método anterior, en el que se hace un estudio estadístico de la respuesta del sistema a diferentes entradas generadas aleatoriamente. El modelo predice las respuestas en función de las entradas dadas.

En nuestro caso se han empleado una combinación de los dos primeros, como se verá en el capítulo dedicado a la descripción de los distintos componentes del sistema de nivel. Por una parte, para aquellos elementos incorporados al proceso de los cuales disponíamos de las características, tanto estáticas como dinámicas, proporcionadas por el fabricante, y los márgenes de error de sus diseñadores, hemos realizado unas pruebas sencillas para determinar los parámetros ajustados a nuestra instalación. Por otra, para aquellos elementos de los que no disponíamos de esa información, y para el proceso en sí, se ha realizado un análisis de las respuestas temporales de los mismos, para determinar sus comportamientos.

Con el conocimiento propio del proceso, y del correspondiente a los demás elementos de la instalación (sensores, transductores, elementos de control, etc.) puede establecerse un modelo general, que puede ser simulado en un ordenador, y sobre el que deben realizarse las pruebas experimentales que se hubiesen practicado sobre el sistema real. Con el fin de conseguir una simulación ajustada lo más posible a la realidad, es necesario disponer de todos los parámetros, y características de los distintos elementos que forman nuestra instalación. El procedimiento de simulación es válido cuando las pruebas a realizar sobre el proceso no son factibles por problemas de estabilidad, sensibilidad de los instrumentos de medida, o tiempo de realización de las mismas (en el capítulo siguiente veremos que el tiempo de establecimiento del sistema es de alrededor de 3 horas). Las ventajas de realizar una simulación son claras: entrenamiento de operadores sin peligro para la planta real, diseño y ajuste óptimo de los parámetros de controladores, planificación de acciones industriales generales, caracterización de dispositivos complejos difíciles de regular, análisis temporales de gran importancia, etc.

Un objetivo más interesante sería tener la oportunidad de emplear la simulación ejecutándose en paralelo con las operaciones reales de la planta. Por ejemplo, comprobando cada cierto tiempo el comportamiento de algunas de las partes de la instalación, como sensores, actuadores, etc., comparando valores reales y valores simulados, empleando éstos si el elemento en cuestión falla en su operación; o también utilizando la simulación para estimar valores que no pueden ser medidos por sensores, empleándolos en el flujo de control. Veremos en el capítulo dedicado a la herramienta que hemos empleado para realizar la simulación, que ésta está diseñada para realizar este objetivo de una forma eficiente.

Pueden diseñarse simulaciones superficiales o profundas. Una simulación superficial es una colección de reglas heurísticas (condición4acción) que modelan características empíricas. Su desarrollo no es complicado al basarse precisamente en la observación del sistema, y no en los principios físicos del mismo, pero no es capaz de anticiparse a todos los posibles sucesos. En contraste, una simulación profunda modela los principios de causalidad de los mecanismos internos en que se basan las características observables, donde la representación de los razonamientos físicos involucrados es un aspecto fundamental. Hacer una simulación profunda es difícil, pero es mucho más eficiente al poder modelar todos los posibles estados del sistema. Esto está íntimamente relacionado con lo estudiado cuando se describieron los sistemas expertos, referente al

conocimiento profundo y superficial, en el que se mencionó la dificultad de desarrollar adecuadamente modelos detallados, debido a la complejidad inherente de los procesos.

En nuestra simulación del proceso de nivel se ha empleado una combinación de los tipos a los que hacemos referencia. La herramienta de sistemas expertos que hemos utilizado, G2, incorpora un simulador totalmente integrado que ofrece las dos posibilidades. Por un lado, podemos modelar el estado del sistema mediante ecuaciones en diferencias o diferenciales, y por otro, mediante el motor de inferencias podemos establecer relaciones en base a reglas heurísticas. Debido a las características presentadas en el apartado anterior, podríamos emplear una base de conocimiento separada totalmente de los mecanismos de inferencia que la controla.

Es en este sentido en el que la Inteligencia Artificial, y más en concreto los sistemas expertos, pueden añadir una visión distinta a los sistemas o herramientas de simulación tradicionales. Primero, empleando técnicas de SE que desarrollen la estructura de componentes declarativos de la base de conocimiento, se proporciona un entorno que es muy fácil de instalar y modificar por personal no relacionado con computadores o con IA. Segundo, los métodos de simulación subyacentes y técnicas de modelado pueden ser dirigidos por los datos, en un sentido paramétrico, desde los componentes declarativos de la base de conocimiento [36]. Como vimos, esto significa que se elimina la necesidad de hacer modificaciones en el código de los procedimientos, o en la comprobación de subrutinas, al variar la información en función de una nueva situación o de un nuevo conocimiento.

Por otra parte, en cualquier simulación se emplean gran cantidad de tratamientos numéricos procedurales. Entonces, para realizar una simulación completa sería conveniente disponer, tanto de herramientas capaces de proporcionar un modelado profundo y superficial de aspectos físicos y heurísticos del sistema, como de técnicas que pudieran implantar de una forma eficiente el empleo de procedimientos numéricos. Un entorno de simulación de este tipo, podría proporcionar grandes ventajas, en cuanto a entrenamiento, ganando experiencia sin tener que manipular los elementos reales de la planta, diseño, debido a las facilidades de interpretación y explicación de los resultados que se alcanzasen, planificación y control, debido a la comodidad de realizar exploraciones cambiando la base de conocimiento.

2

DESCRIPCIÓN DEL PROCESO. ELEMENTOS Y DISPOSITIVOS

DESCRIPCIÓN DEL PROCESO

Cualquier proceso de control de nivel consiste en mantener una determinada altura de un líquido en un depósito, con caudales de entrada y salida, controlando las desviaciones que se producen en dicha altura debido a las variaciones en los caudales. Para efectuar el control se hace uso del concepto de realimentación, es decir, se va insertando en el propio sistema el comportamiento del mismo, de modo que se compara en cada momento la altura real del líquido y la deseada, utilizando la diferencia como medio de control.

Existen dos formas distintas de ejercer el control en un sistema de nivel, dependiendo si se actúa sobre el caudal de entrada, o sobre el de salida. En nuestro caso, se emplea una válvula de control sobre el caudal de salida del depósito. Si el nivel real del líquido sobrepasa el valor deseado, se abre la válvula, con lo cual la altura empieza a disminuir. En caso contrario la válvula se cierra, aumentando el nivel.

La variable del proceso que va a ser controlada es la altura del líquido. Ésta se mide empleando un convertidor de presión en intensidad de corriente, situado en la parte inferior del depósito, ya que la presión que ejerce el fluido en este punto es proporcional a la altura del mismo. Este elemento debe ser muy preciso, pues al estar en el camino de realimentación, su sensibilidad es muy alta, de modo que pequeñas desviaciones influyen notablemente en el comportamiento del sistema controlado. Esta señal de corriente llega al controlador, en el que precisamente se ha seleccionado la altura deseada escogiendo el valor de intensidad que se espera llegue del proceso para esa altura. El bloque regulador produce una señal de control, también en corriente, actuando sobre el proceso, abriendo o cerrando la válvula. Para que esto se produzca es necesario un convertidor intensidad-

presión, pues la válvula se acciona por presión. De esta forma varía el caudal de salida del depósito, modificando la altura del líquido.

La otra entrada que hay sobre el proceso es el caudal de que entra al depósito. Dependiendo del valor de este caudal, y fijado un valor para la altura del líquido, variará el de salida por actuación de la válvula, obteniéndose en el estacionario un equilibrio dinámico. Las perturbaciones que se producen en el caudal de entrada provocan un desequilibrio que el lazo de control se encarga de corregir. La figura 2.1 representa el diagrama de bloques del sistema de control.

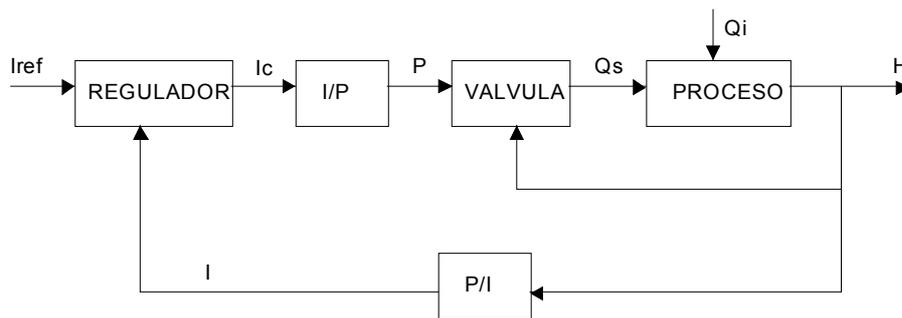


Fig. 2.1 Diagrama de bloques del sistema de control.

DESCRIPCIÓN DE LOS ELEMENTOS Y DISPOSITIVOS

En este apartado pretendemos analizar cada dispositivo de nuestra instalación. La figura 2.2 muestra un esquema de la misma. Primero describiremos las dimensiones físicas del depósito y de las tuberías; después analizaremos elementos imprescindibles en un proceso de control de nivel, como son el rotámetro, el convertor presión-intensidad, el transductor intensidad-presión, y sobre todo, la válvula de control, que es fundamental al determinar las principales características de los procesos de nivel [9]. Por último, calcularemos las constantes de tiempo asociadas al sistema global.

DEPÓSITO Y TUBERÍAS

Se dispone de un depósito de uralita de 82 cm de altura y con una sección media aproximada de 1196 cm², en el que se pretende mantener la altura de líquido con el lazo de control. El depósito no es perfectamente cilíndrico, sino más bien de sección elíptica. Además la variación de sección desde un extremo a otro del mismo es de alrededor de 100 cm², detalle de gran importancia, no sólo para la comprobación posterior de nuestra simulación, sino también para la obtención de la constante de altura de la válvula, que más adelante se verá.

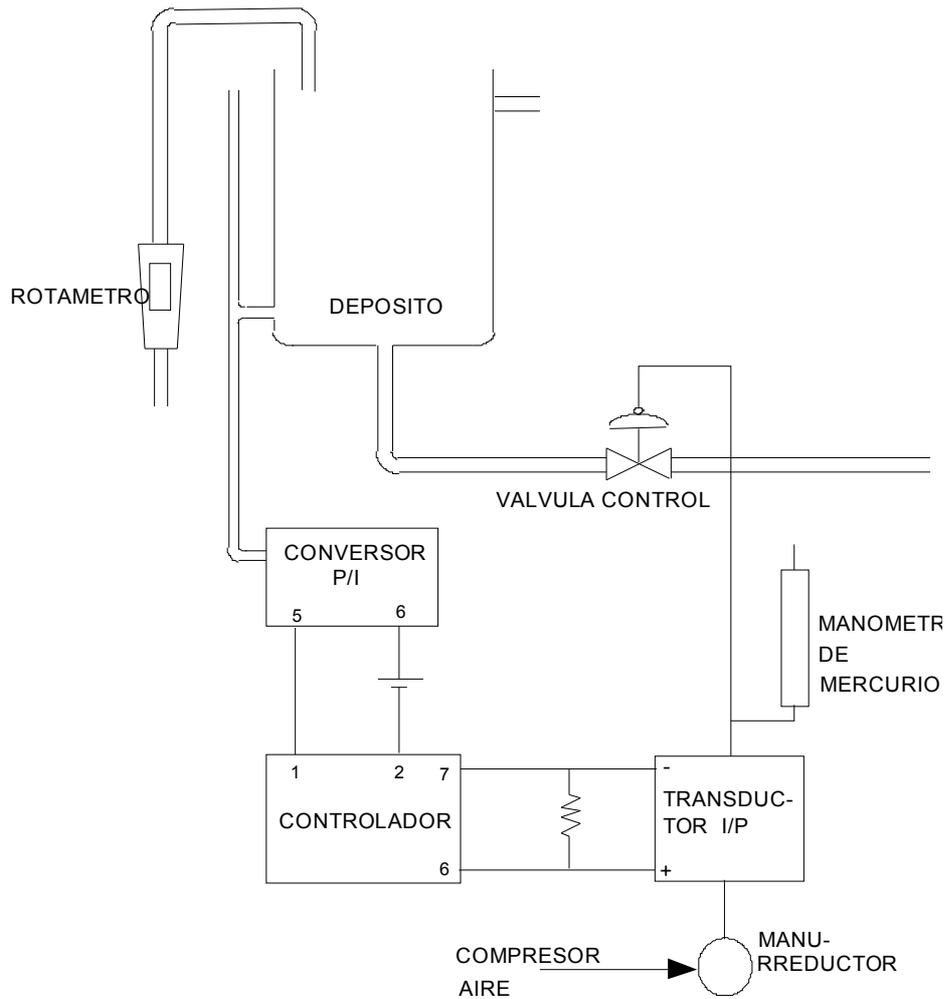


Fig. 2.2 Esquema de la instalación.

El depósito posee tres orificios, que se detallan a continuación:

- El primero, situado en el centro de la base, es circular con los cantos vivos, con un diámetro de 2,1 cm. Este orificio comunica con la tubería de desagüe, de diámetro nominal 3/4 de pulgada.
- El segundo, situado en la parte superior a 7,5 cm del borde, es circular con los cantos vivos, también de 2,1 cm de diámetro y conecta con una tubería también de 3/4 ". Este orificio sirve de aliviadero para evitar rebosamiento.
- El último, situado a 10,5 cm del borde inferior, posibilita el paso del líquido desde el depósito hacia un tubo en U, en una de cuyas ramas puede observarse la altura del líquido, mientras la otra está conectada a la entrada del transductor presión-intensidad, a 100,0 cm de la base del depósito. Este transductor detecta la presión existente debido a la altura de esta columna, convirtiéndola en una señal en intensidad de corriente.

Nuestro interés se centrará en la altura comprendida entre los dos últimos orificios, pues, debido al diseño de la instalación, no obtenemos datos fiables del nivel del líquido en el depósito fuera de este intervalo. Tomaremos las alturas respecto al sensor de nivel (convertor presión-intensidad) de manera que, por lo dicho anteriormente, el 0 % de altura corresponde con 110,0 cm y el 100 % de altura con el 173,5 cm.

Altura (cm)	d1 x d2 (cm x cm)	Sección (cm ²)	Sección media (cm ²)	Volumen Total (cm ³)
173,5	40,4 x 39,2	1243,8	1243,8	4353,3
170,0	40,4 x 39,2	1243,8	1240,7	6203,5
165,0	40,2 x 39,2	1237,6	1233,0	6165,0
160,0	40,0 x 39,1	1228,4	1223,7	6188,7
155,0	39,8 x 39,0	1219,1	1209,8	6049,2
150,0	39,4 x 38,8	1200,6	1197,5	5987,8
145,0	39,3 x 38,7	1194,5	1194,5	5972,5
140,0	39,3 x 38,7	1194,5	1191,5	5957,2
135,0	39,2 x 38,6	1188,4	1188,4	5942,0
130,0	39,2 x 38,6	1188,4	1183,8	5919,2
125,0	39,0 x 38,5	1179,3	1174,7	5873,7
120,0	38,8 x 38,4	1170,2	1162,6	5813,2
115,0	38,5 x 38,2	1155,1	1152,1	5760,5
110,0	38,4 x 38,1	1149,1		

Tabla 2.1 Variación de la sección con la altura.

El volumen de control (0 - 100 %) lo determinamos considerando sección elíptica y midiendo los ejes correspondientes, comprobando que efectivamente la sección no es uniforme. Para nuestros cálculos posteriores será importante considerar este hecho. En la tabla 2.1 se detallan las alturas tomadas como referencia, la sección para esa altura, la sección media para cada intervalo de altura de 5 cm, y el volumen correspondiente. De los cálculos obtenemos un volumen de control de aproximadamente 76100 cm³.

En la figura 2.3 se indican las medidas, en cm, del depósito y de las tuberías.

La tubería de desagüe del depósito posee un codo exterior a 83,0 cm de la base del depósito y una abertura en T, también exterior a 37,0 cm de este codo, que sirve para medir la caída de presión desde el depósito hasta ese punto. A 29,0 cm se sitúa la válvula.

Debemos hacer una mención especial al diámetro nominal de las tuberías, pues aunque indiquen 3/4 NPT, lo cual supone 1,9 cm, el diámetro interior real es de 2,1 cm [44]. Según esto, la sección de las tuberías corresponde a 3,44 cm².

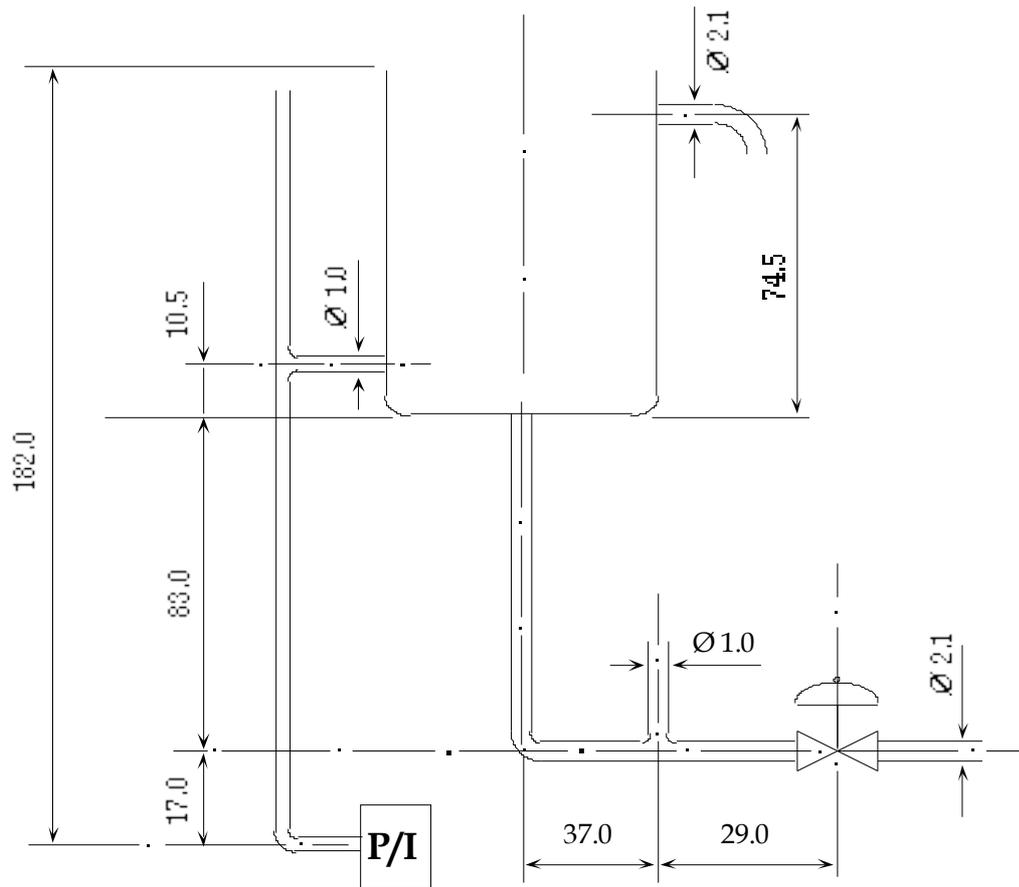


Fig. 2.3 Depósito y tuberías.

ROTÁMETRO

El rotámetro es un medidor de caudal que consiste en un tubo de área variable, dispuesto en vertical, en el que un flotador cambia su posición dentro del tubo proporcionalmente al flujo del fluido. Su funcionamiento se basa en el equilibrio dinámico del flotador, que se encuentra sometido a tres fuerzas: su peso, el empuje del fluido y la fuerza de arrastre. Con él medimos el caudal de entrada al depósito.

Por sí mismo no es un elemento importante en la instalación, al ser simplemente un indicador. Sin embargo, vamos a necesitar de su ayuda en muchas partes de nuestro análisis posterior; y sobre todo ha sido fundamental a la hora de determinar algunas características de otros elementos, que más tarde estudiaremos, pues es el único que poseemos que proporciona una medida de caudal.

Nuestro rotámetro es de tubo de vidrio con flotador cilíndrico de borde saliente y de cara inclinada contra el flujo. Está graduado en porcentaje de 10 a 100 % de la escala total en lugar de en unidades de caudal. Esto es bastante frecuente, ya que las fuerzas de arrastre y empuje dependen del peso específico del fluido, y por tanto la escala es diferente según el líquido utilizado [8], y aunque

el fabricante proporciona el valor de conversión cuando el líquido empleado es agua (1,14 cm³/s), ha sido necesaria su calibración.

La calibración se consigue básicamente manteniendo constante el paso de caudal a través del rotámetro, y midiendo la cantidad de líquido en un tiempo dado, en nuestro caso un minuto para toda la escala. Para caudales menores del 30 % la medida no fue posible debido a las fluctuaciones de la red de suministro, imposibles de evitar.

%	cm ³ /min	cm ³ /s
30	2330	38,83
35	2650	44,17
40	2920	48,67
45	3280	54,67
49	3560	59,33
55	3910	65,17
61	4340	72,33
66	4670	77,83
71	4930	82,17
77	5300	88,33
82	5630	93,83
87	5990	99,83
92	6260	104,33
97	6630	110,50
100	6740	112,33

Tabla 2.2

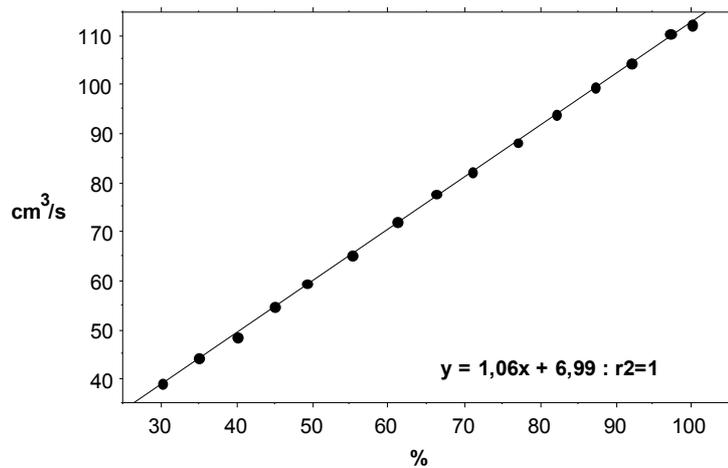


Fig. 2.4 Calibrado del rotámetro.

La figura 2.4 presenta la regresión simple de los valores de la tabla 2.2, con un intervalo de confianza del 95 %. Como puede apreciarse, existe linealidad entre el % de escala del rotámetro y el caudal, a pesar de que las medidas no son muy exactas (debido fundamentalmente al método de medida y a que el flujo de entrada sufre oscilaciones apreciables que, aunque se pretende, no pueden ser evitadas). Debemos considerar además que para cualquier fluido, la densidad depende de la temperatura; si bien, en nuestro caso no existirán grandes cambios en las condiciones de trabajo. Adicionalmente debemos señalar que debido a lo antes comentado, se introducen inexactitudes muy superiores a la que supone el despreciar los efectos de la temperatura.

Para nuestros fines, tomaremos la recta obtenida de la regresión, sin tener muy en cuenta la característica dinámica del dispositivo, pues las variaciones de caudal en la red de suministro representan precisamente una de las perturbaciones que tendremos en el sistema, y podemos pensar que la situación transitoria puede ser debida a esas perturbaciones.

$$Q = 1,06x + 6,99 \tag{2.1}$$

donde **Q** viene dado en cm³/s y **x** representa el % de la escala del rotámetro.

CONVERSOR PRESIÓN-INTENSIDAD

Este convertidor se emplea como sensor para detectar la presión que ejerce el líquido contenido en el depósito, lo que equivale a medir su altura, transformándola en una señal eléctrica que pueda ser transmitida al regulador para que éste proporcione la señal de control adecuada. En este sentido, se trata tanto de un sensor como de un transmisor.

En nuestra instalación se ha empleado un sensor-transmisor PHILIPS, con capacidad para convertir la presión de una columna de líquido de un depósito atmosférico, entre 0,08 y 0,4 bar (80,0 y 400,0 cm de altura de agua) en una señal proporcional de intensidad entre 4 y 20 mA de corriente continua. Debido a este hecho, ha sido necesario elevar el depósito por encima de los 80,0 cm, hasta una altura de 100 cm sobre el convertidor, si bien, como comentamos anteriormente, la toma de altura se va a encontrar a 110 cm.

El empleo de señales eléctricas de corriente continua entre 4 y 20 mA tiene su explicación. Estos valores se emplean debido a que tienen un nivel suficiente y de compromiso entre la distancia de transmisión y la robustez de los equipos empleados habitualmente en la industria. Al ser continua y no alterna, elimina la posibilidad de captar perturbaciones y está libre de corrientes parásitas. Por tanto, emplea sólo dos hilos que no precisan blindaje y permite actuar directamente sobre amperímetros, potenciómetros, calculadores analógicos, etc., sin necesidad de utilizar rectificadores ni modificar la señal. Además, el nivel mínimo seleccionado de 4 mA elimina el problema de la corriente residual que se presenta al desconectar los circuitos electrónicos a transistores. Este "cero vivo" ofrece las ventajas de poder detectar una avería por corte de un hilo, y permitir diferenciar todavía más el ruido de la transmisión cuando la variable está en su nivel más bajo.

El sensor está formado por un cristal de silicio con propiedades piezorresistivas, que al deformarse físicamente por la acción de la presión, genera una señal eléctrica, que desequilibra un puente de Wheatstone en su interior, proporcionando la intensidad de salida. Su señal de respuesta para una variación de presión es lineal, siendo además adecuado para medidas dinámicas, al ser capaz de respuestas rápidas para altas frecuencias [8]. Su principal inconveniente, en general para cualquier transductor electromagnético de tipo piezoeléctrico, es su sensibilidad a los cambios de temperatura, de modo que el error de cero por influencia de la temperatura ambiente puede ser de hasta un 4 %.

La alimentación de los transmisores puede realizarse con el mismo par de hilos del convertidor. En nuestro caso así es. Más aún, estos mismos hilos sirven también para aplicar la intensidad de salida a la entrada del regulador, que es por donde recibe la alimentación. Esquemáticamente se muestra en la figura siguiente.

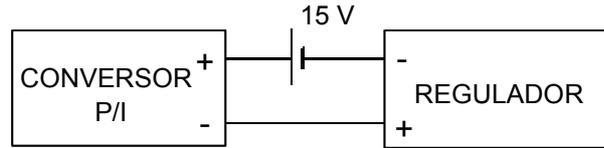


Fig. 2.5 Acoplamiento del sensor y del regulador.

El regulador y el transmisor se acoplan directamente, ya que ambos utilizan el rango de intensidades de 4 a 20 mA. Sin embargo, la escala de presiones de entrada al conversor es muy amplia para la altura del depósito empleado. Debido a ello, en nuestro caso debemos graduar el conversor P/I para las alturas de nuestro proceso. Es decir, debemos obtener 4 mA para una altura del depósito del 0 %, que se corresponde con una altura de la columna de líquido de 110,0 cm, y 20 mA para el 100 %, que se corresponde con una altura de 173,5 cm.

Esto se ha logrado ajustando los tornillos del 0 y del 100 % que el conversor posee para tal efecto. Una vez realizado este ajuste, procedimos a determinar la característica estática intensidad-presión del conversor. En vez de medir la presión a la entrada del conversor, registramos la altura de líquido y la intensidad de salida directamente del sensor. Esta tabla de valores nos permite construir la gráfica intensidad-altura, que será proporcional a la intensidad-presión, por serlo la presión del líquido a su altura.

$$P = \rho g \Delta h \tag{2.2}$$

Siendo **P** la presión del líquido, ρ su densidad, Δh es la diferencia entre la altura del líquido y la del sensor, y **g** la aceleración de la gravedad. Aquí aplicamos las mismas consideraciones sobre la temperatura que hacíamos en el caso del apartado referente al rotámetro.

De esta forma se ha conseguido calibrar el P/I de acuerdo a la siguiente tabla:

H(cm)	I(mA)
110,0	4,0
118,0	6,0
126,0	8,0
134,1	10,0
142,0	12,0
149,8	14,0
157,7	16,0
165,5	18,0
173,5	20,0

Tabla 2.3

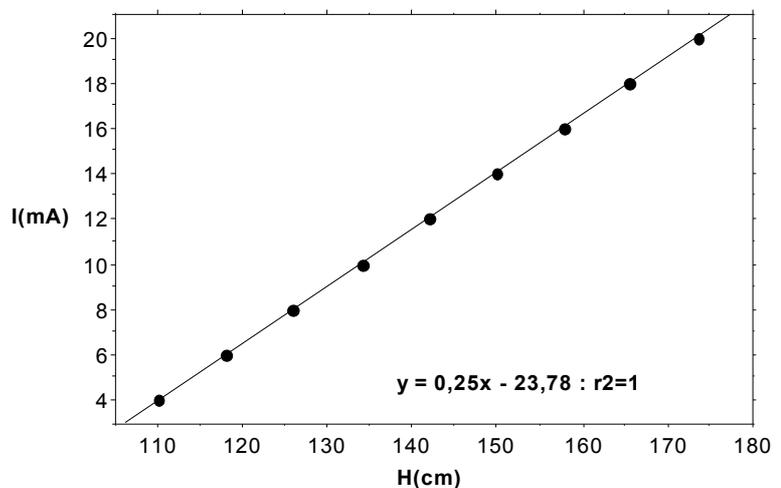


Fig. 2.6 Calibrado del sensor presión-intensidad.

Como cabría esperar, la característica estática del convertor P/I es lineal. Se observa que efectivamente, el rango del 0 al 100 % en altura del líquido se corresponde con el rango de 4 a 20 mA.

Como comentamos anteriormente, este tipo de convertor posee una rápida respuesta en su comportamiento dinámico, de modo que para nuestros fines supondremos que la respuesta es instantánea. Por ello la ecuación que liga la evolución temporal de las variables del convertor vamos a suponerla de la siguiente forma:

$$I(t) = 0,25H(t) - 23,78 \quad (2.3)$$

donde **I** viene dada en mA y **H** en cm.

TRANSDUCTOR INTENSIDAD-PRESIÓN

Es un convertor electro-neumático que opera sobre el elemento final de control de acuerdo con la señal de corriente que le llega procedente del regulador. Este instrumento convierte una señal estándar de 4 a 20 mA de corriente continua en una señal neumática de 0,2 a 1 bar, si bien, tanto el cero como el alcance pueden graduarse para adecuarse lo más posible a las necesidades de cada proceso. Necesita una alimentación de 1,4 bar, que en nuestro caso es proporcionada por un compresor de aire, aunque es necesario un manorreductor, pues dicha presión habitualmente es mayor.

Se dispone además de un manómetro de mercurio a la salida del transductor para medir la presión proporcionada por el mismo. Esta presión es la que actuará sobre la válvula, cerrando o abriéndola, ejerciendo así la acción de control. Aunque las especificaciones de la válvula indican que los valores de inicio de cierre y cierre total son 1,5 y 4,4 psi (7,7 y 22,7 cm de Hg) respectivamente, experimentalmente se comprueba que la válvula empieza a cerrarse en torno a los 11,5 cm de Hg, y se cierra totalmente en torno a los 22,0 cm de Hg. La falta de exactitud en la exposición viene provocada por la existencia de histéresis, que analizaremos cuando describamos la válvula de control.

Se tiene entonces un rango de variación de presiones de entrada a la válvula de aproximadamente 10,5 cm de Hg, sin embargo el convertor intensidad-presión puede proporcionar un alcance mucho mayor (en su valor nominal de 60 cm de Hg). La posibilidad de poder graduar este alcance es importante, así se ha graduado en su valor mínimo para ajustarse lo más posible a nuestro rango de presiones. No obstante, sigue siendo bastante superior a la variación de presión de accionamiento de la válvula.

Debido a esta situación, las intensidades de control provenientes del regulador, de 4 a 20 mA, producen un rango de presiones demasiado elevado, que es debido limitar. Es por ello que se ha diseñado un dispositivo que reduce la intensidad de entrada al convertor. Este dispositivo está

formado por una resistencia en paralelo con el conversor. La resistencia de entrada al conversor es de $84,7 \Omega$, y la resistencia colocada en paralelo es de 100Ω , de modo que la intensidad de entrada al conversor es el 54,4 % de la total. Esto significa que las intensidades de entrada al conversor estarán en el rango de 2,17 y 10,88 mA.

Aun así, este rango de intensidades proporciona una variación superior a la presión de accionamiento de la válvula. Sería más conveniente emplear una resistencia menor que redujera el rango de intensidades de entrada al conversor, por ejemplo de 50Ω . Pero para niveles bajos de intensidades de control cercanas a 4 mA, la intensidad de entrada al conversor sería en torno a 1,6 mA, que está muy por debajo de los 4 mA de diseño para los cuales es lineal. Nos hemos decidido entonces por una solución compromiso, empleando la resistencia de 100Ω , pues establece un rango de accionamiento de la válvula conveniente, a pesar de que hay valores de intensidad que no accionan la válvula, como puede verse en la tabla 2.4.

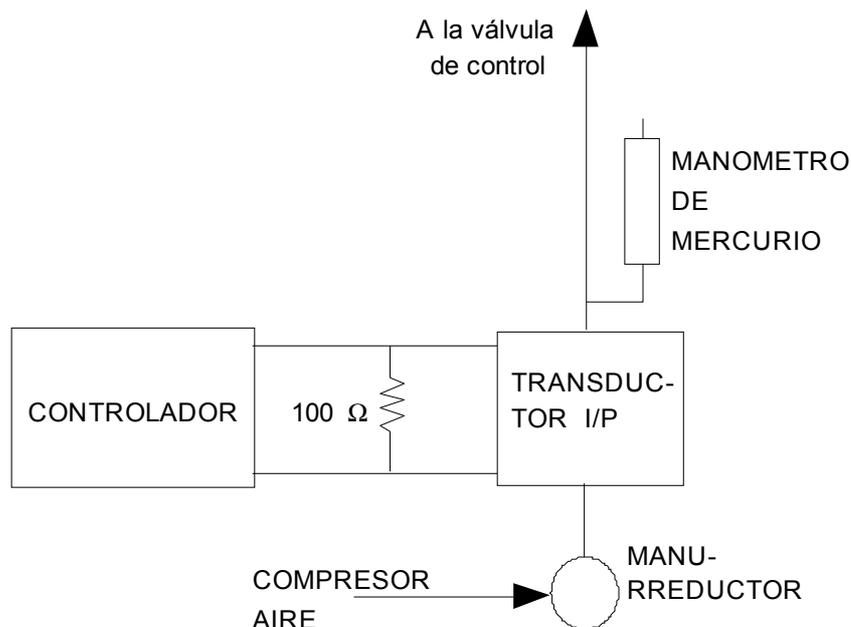


Fig. 2.7 Resistencia en paralelo.

Además se ha calibrado el cero del aparato para que las presiones de utilización se mantengan dentro de las presiones de actuación sobre de la válvula de control.

Adicionalmente se han realizado pruebas con el proceso en funcionamiento, para determinar la influencia de la altura de líquido en el depósito sobre la presión de salida del conversor, obteniendo que no existen diferencias apreciables en su comportamiento. Este resultado es importante, pues

indica que la carrera de la válvula va a depender solamente de la presión procedente del convertor, y no de la altura del líquido en el depósito.

Para realizar el calibrado se ha empleado una fuente de corriente, un miliamperímetro para medir la intensidad y un manómetro de mercurio con el que se registran las presiones, en el estacionario, para una presión de alimentación de 1,4 bar. Este detalle es de importancia, pues dependiendo de la presión de alimentación, se obtienen distintas características. Debemos mencionar aquí que la respuesta para un manómetro de este tipo se ajusta a un sistema de segundo orden subamortiguado, que podría confundir pareciendo que es la respuesta dinámica del transductor. Más adelante se detallará como es su respuesta dinámica.

I_c (mA)	I_i (mA)	P (cm Hg)
3,73	2,03	5,9
5,51	3,00	8,7
7,39	4,02	11,2
9,26	5,04	14,0
11,12	6,05	16,8
12,92	7,03	19,5
14,70	8,00	22,2
16,59	9,03	25,0
18,39	10,01	27,6
20,30	11,05	30,5

Tabla 2.4

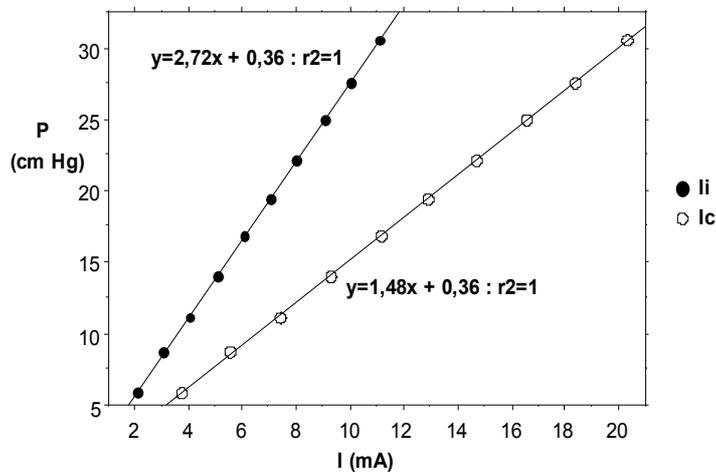


Fig. 2.8 Calibrado del transductor intensidad-presión.

Ambas rectas representan la regresión lineal con un intervalo de confianza del 95 %, de la presión de salida frente a la intensidad de aplicación al convertor (I_i), y a la intensidad de control (I_c). De este modo puede observarse que la característica estática para la intensidad de entrada viene dada por la siguiente recta:

$$P = 2,72 I_i + 0,36 \tag{2.4}$$

donde I_i viene dada en mA, y P en cm de mercurio. La conversión de I_c a I_i se realiza simplemente multiplicando por 0,54.

Característica dinámica

Se dispone de información sobre el comportamiento en frecuencia de este convertor. El fabricante proporciona las características de amplitud y ángulo de fase en el dominio de la frecuencia, de modo que puede obtenerse la característica dinámica del aparato.

Dichas características pueden verse en las figuras 2.9 y 2.10.

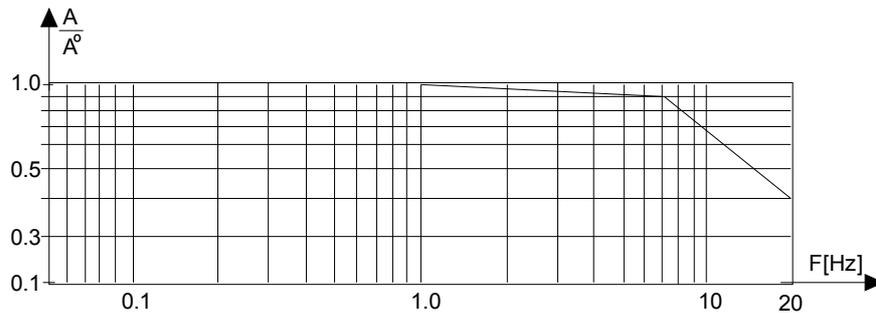


Fig. 2.9 Diagrama de amplitud.

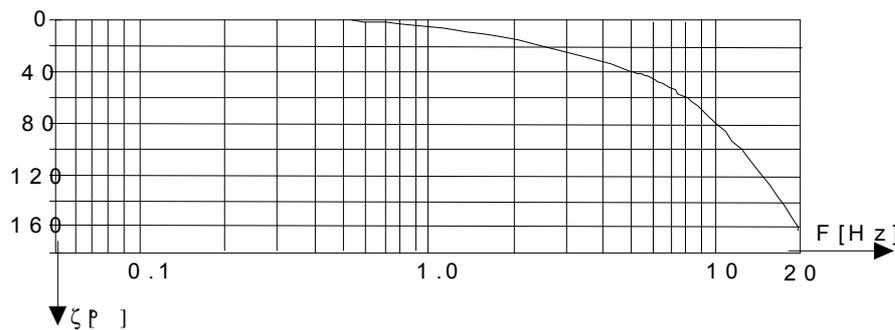


Fig. 2.10 Diagrama de fase.

Se ha realizado un estudio sobre dichas características, encontrándose dos resultados. Por un lado, la característica de ganancia indica la existencia de un polo en $T=0,02$ s. Por otro, la característica de fase a bajas frecuencias indica un retardo exponencial en su función de transferencia del orden de $\tau=0,01$ s.

La función de transferencia total del convertidor viene dada entonces por:

$$\frac{P(s)}{I_i(s)} = \frac{k e^{-0,01s}}{1+0,02s} \quad (2.5)$$

Realizando la transformada inversa de Laplace, obtenemos en el dominio temporal:

$$0,02\dot{P}(t) + P(t) = k I_i(t - 0,01) + k' \quad (2.6)$$

En el estacionario, la derivada se anula, de modo que comparando con la ecuación (2.4), nos queda:

$$0,02\dot{P}(t) + P(t) = 2,72 I_i(t - 0,01) + 0,36 \quad (2.7)$$

Para nuestro estudio nos interesa más la intensidad proveniente del regulador. Esta no va a ser retrasada ni adelantada por ningún elemento eléctrico hasta llegar al transductor, pues la reducción en la corriente se produce por una resistencia en paralelo.

$$0,02\dot{P}(t) + P(t) = 1,48 I_c(t - 0,01) + 0,36 \quad (2.8)$$

Parece claro pensar que tanto el retardo puro como la influencia del polo son muy pequeños, más aún cuando los comparamos con la constante de tiempo de respuesta de la válvula a una acción de control, que es del orden de 0,6 s para un desplazamiento total de la carrera. Por esto vamos a considerar, a todos los efectos, que el transductor intensidad-presión se comporta como un elemento que no va a producir ningún retardo, es decir, de respuesta instantánea. Lo que se ha hecho ha sido, precisamente, despreciar la contribución de los polos más alejados del origen. Así pues, la ecuación 2.8 nos queda de la siguiente forma:

$$P(t) = 1,48 I_c(t) + 0,36 \quad (2.9)$$

VÁLVULA DE CONTROL

Introducción

En el control automático de los procesos industriales, la válvula de control juega un papel muy importante en el bucle de regulación. Realiza la función de variar el caudal del fluido de control, que modifica su vez el valor de la variable medida, comportándose como un orificio de área continuamente variable. En nuestro caso la variable medida es la altura del líquido en el depósito, que viene determinada por los caudales de entrada y salida; siendo a su vez el caudal de salida regulado por la válvula, que abre o cierra el orificio de salida según las órdenes del controlador, a través del transductor intensidad-presión.

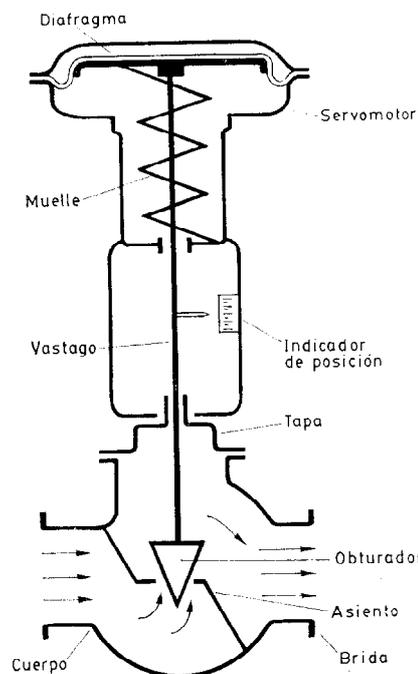


Fig. 2.11 Válvula de control representativa.

La válvula se compone básicamente del cuerpo y del servomotor. El cuerpo contiene en su interior el obturador y los asientos. El obturador es el que realiza la función de control de caudal, gracias al orificio de paso variable que forma al modificar su posición relativa respecto de los asientos. El obturador está unido a un vástago que pasa a través de la tapa del cuerpo y que es accionado por el servomotor. Este es de tipo neumático y consiste en un diafragma con resorte, que en nuestro caso trabaja entre 11,5 y 22,0 cm de Hg (es decir, las posiciones extremas de la válvula, o carrera, corresponden a esas presiones). Al aplicar presión sobre el diafragma, el resorte se comprime de tal modo que el mecanismo empieza a moverse, y sigue moviéndose hasta que se llega a un equilibrio entre la fuerza ejercida por la presión del aire sobre el diafragma y la fuerza ejercida por el resorte. En la figura 2.11 puede verse una válvula de control típica, bastante ajustada en su forma a la de nuestra instalación.

En nuestro caso se dispone de una válvula de globo, de asiento simple y de acción directa, es decir, el obturador cierra al aplicar presión (lógico al controlar el caudal de salida, pues en caso de fallo en la alimentación o de avería, la válvula abre, evitando rebosamiento en el depósito). En principio este tipo es adecuado en la instalación pues la presión del fluido es baja, si bien, veremos más adelante, que la característica de la válvula provoca una regulación muy pobre.

Características de caudal

La forma del obturador determina las características de caudal de la válvula, es decir, la relación existente entre la posición del obturador y el caudal de paso del fluido. La *curva característica inherente* se representa generalmente considerando como abscisas el porcentaje de caudal máximo bajo una presión diferencial constante, y como ordenadas la carrera del obturador [9]. Las curvas características más significativas son la de apertura rápida, la lineal y la isoporcentual (figura 2.12). En nuestro caso concluiremos que la válvula de control es del tipo de apertura rápida.

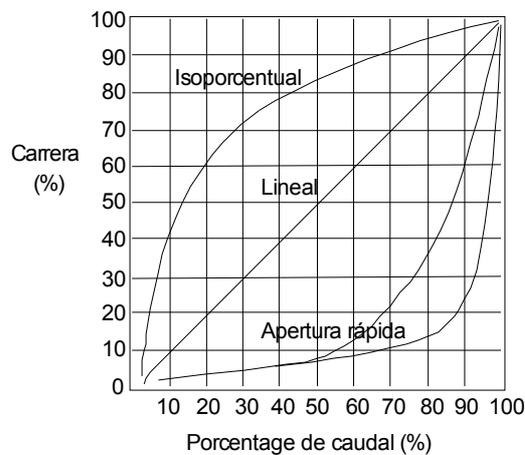


Fig. 2.12 Tipos de características inherentes de la válvula.

No obstante, tenemos que indicar que cuando la válvula trabaja en condiciones reales, la presión diferencial cambia cuando varía su apertura, por lo cual la curva de caudal real que relaciona la carrera con el caudal, se aparta de la característica inherente. Esta nueva curva se denomina *característica de caudal efectiva*, y depende de la instalación, debido a las resistencias asociadas a las tuberías y a las características de las bombas, tanques, etc., siendo distinta en cada instalación para la misma válvula.

Para determinar la característica de caudal efectiva se toman valores de carrera frente a caudales, a presión diferencial constante y en distintas situaciones (niveles diferentes). Al no disponer de un dispositivo que mida el flujo a través de la válvula, tenemos que hacerlo de forma indirecta, aprovechando el rotámetro que mide el caudal de entrada al depósito. En condiciones de equilibrio dinámico, cuando el nivel del líquido en el depósito se mantiene constante, el caudal de entrada es el mismo que el de salida, y éste es el que pasa a su través, además la presión diferencial es constante. De esta forma, se ha medido el caudal de entrada con el rotámetro (haciendo la conversión referida en la ecuación 2.1) y la carrera de la válvula, para poder obtener la curva de caudal efectiva.

Carrera (mm)	Q(25% H) (cm ³ /s)	Q(50% H) (cm ³ /s)	Q(75% H) (cm ³ /s)	Carrera (%)	Caudal (%)
11,1	114,62	—	—	100,0	100,0
11,0	114,62	—	—	99,1	100,0
10,0	114,62	—	—	90,1	100,0
9,0	114,62	—	—	81,1	100,0
8,0	113,47	—	—	72,1	99,1
7,0	112,33	—	—	63,1	98,1
6,0	111,18	—	—	54,0	97,1
5,0	107,74	—	—	45,0	94,0
4,0	97,18	104,60	111,57	36,0	84,8
3,0	83,38	89,74	97,17	27,0	72,7
2,0	57,92	61,10	66,40	18,0	50,5
1,0	32,45	34,57	36,69	9,0	28,3
0,0	0,0	0,0	0,0	0,0	0,0

Tabla 2.5 Medidas de carrera y caudal para distintas alturas.

Debemos señalar además que se han realizado las pruebas para distintos niveles de líquido (25, 50 y 75 % de altura), obteniendo resultados análogos, al menos para valores pequeños de la carrera, pues la falta de una escala suficientemente amplia del rotámetro nos ha impedido poder medir caudales mayores. Este resultado es importante, pues la analogía se establece para valores pequeños de carrera, cuando son más importantes las fuerzas estáticas y dinámicas debidas a la presión diferencial del fluido sobre el obturador, y cuando la diferencia en los resultados pudiera ser más grande. Así pues, la característica de caudal efectiva de la válvula, para cualquier valor del nivel de líquido en el depósito, vamos a tomarla de los resultados de la tabla 2.5, que se representan en la

figura 2.13. Comparando esta figura con la figura 2.12 podemos considerar que nuestra válvula es de apertura rápida.

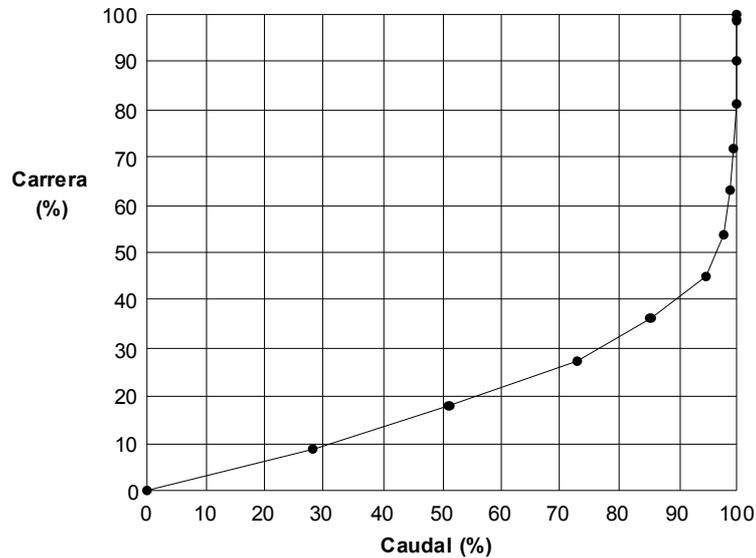


Fig. 2.13 Característica de caudal de la válvula.

El obturador con característica de apertura rápida tiene forma de un disco plano (figura 2.14). En la figura 2.13 puede observarse que el caudal aumenta mucho al principio de la carrera, llegando rápidamente al máximo, y provocando por tanto una regulación muy pobre. Esta característica va a modular el comportamiento del sistema, pues influye notablemente sobre el nivel del líquido en el depósito.



Fig. 2.14 Obturador de apertura rápida.

Para comprobar la característica de caudal de la válvula se ha realizado un experimento sencillo, pero laborioso, consistente en registrar la altura de líquido en el depósito (dentro del intervalo de 110,0 a 173,5 cm) a lo largo del tiempo, para distintas aperturas de la válvula, en situación de vaciado del depósito. Para ello se ha registrado la señal proveniente del sensor presión-intensidad (habilitando una resistencia para medir la tensión entre sus extremos) en un registrador gráfico, frente al tiempo (figura 2.15). Esta gráfica puede proporcionarnos más información, sobre todo cuando tratemos del tema de la constante de altura de la válvula. A nosotros nos interesa el tiempo de vaciado total del volumen de control (recordemos que aproximadamente es de 76100 cm³) para distintas aperturas de la válvula, del cual obtendremos el caudal medio, que nos permitirá entonces establecer una relación entre la carrera y el caudal medio.

FIGURA NO DISPONIBLE

Fig. 2.15 Vaciado del depósito para distintas carreras de la válvula.

Carrera (mm)	Tiempo (s)	Caudal Medio (cm³/s)	Caudal medio (%)
11,1	625	121,5	100,0
11,0	625	121,5	100,0
10,0	625	121,5	100,0
9,0	625	121,5	100,0
8,0	625	121,5	100,0
7,0	630	120,7	99,4
6,0	641	118,5	97,5
5,0	667	114,0	93,8
4,0	714	106,5	87,6
3,0	869	87,4	71,9
2,0	1180	64,4	53,0
1,0	2490	30,5	25,1
0,0	1	0,0	0,0

Tabla 2.6 Tiempos de vaciado del volumen de control para distintas aperturas.

Podemos observar claramente la similitud existente con la tabla 2.5. La ligera desviación en los datos es debida a la falta de precisión al poner la carrera de la válvula. En la tabla 2.5 los resultados obtenidos equivalen a considerar el caudal máximo que pasa a través de la válvula para una carrera dada y a una determinada altura (presión diferencial constante). El caudal máximo se produce en la situación de válvula totalmente abierta; considerando a éste el 100 %, se ha obtenido la figura 2.13. Sin embargo, la tabla 2.6 se ha obtenido midiendo el tiempo total de vaciado, para situaciones de carrera constante, y sin tener en cuenta el nivel de líquido (es decir, sin importarnos la presión diferencial en la válvula), tomando como referencia el tiempo para la misma situación de válvula totalmente abierta. En conclusión, con las dos pruebas realizadas, una a presión diferencial constante y la otra no, nos permiten considerar que la figura 2.13 representa con bastante exactitud la característica efectiva de la válvula de control.

Característica estática presión-carrera

Se ha tratado de determinar la relación existente entre la presión ejercida sobre el servomotor neumático y el desplazamiento del vástago de la válvula, es decir, la característica presión-carrera. Para determinar la característica estática se ha medido la presión empleando el manómetro de mercurio, y midiendo la carrera en mm. Como comentamos en el apartado referente al transductor intensidad-presión, se han realizado pruebas con distintas alturas de nivel de líquido en el depósito, para determinar la influencia sobre la característica de distintas presiones de líquido, comprobando que en esta instalación, no se observan diferencias. Este resultado es importante, pues indica que la carrera de la válvula va a depender solamente de la presión procedente del convertidor, y no de la altura del líquido en el depósito.

Del análisis de la tabla que hemos obtenido puede apreciarse un resultado esperado, la existencia de histéresis en la válvula. Esta histéresis debemos tenerla en cuenta en nuestra

simulación. Señalar también la relación lineal existente entre la presión de entrada y la carrera, tanto para la situación en que la válvula está cerrando, como en la que está abriendo, y esto para distintas alturas de nivel de líquido en el depósito. Sin duda es debido a los niveles bajos de presión diferencial que vamos a tener.

La figura 2.16 representa la regresión lineal de los valores de la tabla 2.7 en las situaciones de cierre y apertura. Observamos claramente la existencia de histéresis. La característica estática para ambas situaciones se representa a continuación:

$$\text{Válvula cerrando} \quad C = -1,18P + 25,86 \quad (2.10)$$

$$\text{Válvula abriendo} \quad C = -1,23P + 26,08 \quad (2.11)$$

donde **P** viene dado en cm de Hg, y **C** en mm.

P(cierra) (cm Hg)	P(abre) (cm Hg)	Carrera (mm)
12,4	12,1	11,1
12,5	12,2	11,0
13,7	13,0	10,0
14,4	13,8	9,0
15,3	14,6	8,0
16,0	15,4	7,0
16,8	16,3	6,0
17,7	17,1	5,0
18,4	17,9	4,0
19,4	18,7	3,0
20,3	19,6	2,0
21,1	20,2	1,0
22,0	21,0	0,0

Tabla 2.7.

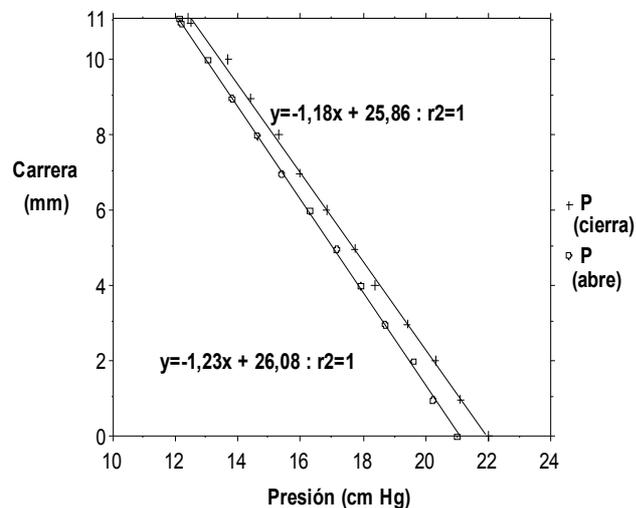


Fig. 2.16 Valores de carrera frente a presión, en situaciones de cierre y apertura.

Característica dinámica presión-carrera

Sería conveniente determinar cuál es el comportamiento dinámico de la carrera de la válvula respecto de la presión de entrada, para determinar si existen retrasos que influyan en nuestro proceso. El servosistema neumático se corresponde con un sistema típico de segundo orden. Es conocido que los sistemas de segundo orden, dependiendo de su comportamiento, pueden clasificarse en tres tipos bien diferenciados: subamortiguados, sobreamortiguados o de amortiguamiento crítico. Nosotros pretendemos ver a qué tipo de sistema de segundo orden se ajusta más. Una forma de determinarlo es proceder a someter al sistema a una entrada escalón en la presión de entrada, y observar su salida. El análisis de esta salida mostrará de qué tipo de sistema se trata.

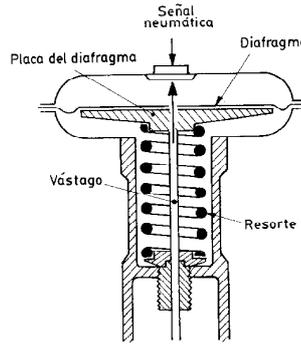


Fig. 2.17 Servomotor neumático.

Para observar la variación temporal de la carrera ha sido necesario un método indirecto, tanto en la realización del escalón de la señal de presión, como en la medida dinámica de la carrera de la válvula. Por una parte se ha empleado el transductor intensidad-presión en el que se ha provocado una variación escalón en la intensidad de entrada, que proporcionará "rápidamente" (estamos considerando también los retrasos dinámicos que posee este conversor) una señal cuasiescalón de salida. Por otra, para medir el desplazamiento del vástago se le ha acoplado un potenciómetro, de modo que podemos representar la diferencia de tensión entre sus terminales respecto del tiempo.

Este método introduce demasiados errores, sin embargo de su análisis podemos afirmar que el sistema se comporta como uno de segundo orden sobreamortiguado, de respuesta relativamente lenta, y que además depende del incremento de la carrera. Podemos simplificar el sistema hacia uno de primer orden, en el que se elimine la raíz no dominante, es decir, despreciando la contribución del polo más alejado del origen [25]. Las variaciones en la constante de tiempo debidas a la carrera son muy pequeñas, máxime cuando el método de medida es poco exhaustivo. Hemos observado que para un recorrido completo de la carrera, la constante de tiempo es aproximadamente de 0,6 s, mientras que para variaciones de 8 y 5 mm, está en torno a 0,5 y 0,4 s, respectivamente.

De esta forma, vamos a suponer que la característica dinámica presión-carrera se comporta como un sistema de primer orden, con una constante de tiempo, que tomaremos de 0,5 s.

$$\frac{C(s)}{P(s)} = \frac{k}{1 + 0,5s} \quad (2.12)$$

Además deberíamos considerar la existencia de histéresis, la cual puede entenderse como un retardo puro (aunque no indica eso exactamente) que puede representarse por una exponencial. Sin embargo no lo vamos a tener en cuenta en el estudio que sigue, pero sí consideraremos el efecto de la histéresis en nuestra simulación.

Con estas suposiciones en el dominio temporal tenemos:

$$0,5\dot{C}(t) + C(t) = kP(t) + k' \quad (2.13)$$

Del estacionario podemos determinar las constantes, a partir de la característica estática:

$$\text{válvula cerrando:} \quad 0,5\dot{C}(t) + C(t) = -1,18P(t) + 25,86 \quad (2.14)$$

$$\text{válvula abriendo:} \quad 0,5\dot{C}(t) + C(t) = -1,23P(t) + 26,08 \quad (2.15)$$

Caudal a través de la válvula

Al analizar sistemas que involucran el flujo de fluidos se hace necesario dividir los regímenes de flujo en laminar y turbulento, de acuerdo con el valor del número de Reynolds (su nombre se debe a Osborne Reynolds (1842-1912), un ingeniero británico que fue quien lo introdujo en 1883). La razón es que a relativamente bajos números de Reynolds se produce un cambio profundo y complicado en el comportamiento de los fluidos, denominado transición hacia la turbulencia [44], en que el flujo deja de ser suave (laminar) y se convierte en fluctuante y agitado (turbulento). La teoría para flujo laminar está muy desarrollada y se conocen multitud de soluciones. Sin embargo, la teoría existente sobre flujo turbulento es semiempírica, basada en análisis dimensional y razonamientos físicos, si bien, existen muchos datos experimentales al ser más común que el laminar.

El número de Reynolds es un parámetro adimensional dado por la siguiente expresión:

$$R_e = \frac{\rho UL}{\mu} = \frac{UL}{\nu} \quad (2.16)$$

donde:

U = velocidad media (m/s)

L = longitud característica transversal (m)

ρ = densidad (kg/m³)

μ = viscosidad (kg/m·s)

ν = viscosidad cinemática (m²/s)

Lo que pretendemos ver es qué rango de caudales proporciona un régimen laminar, en la tubería de desagüe, y cuál un régimen turbulento, para así determinar con mayor seguridad que estudio se aplica a la válvula de control.

La viscosidad cinemática del agua a 1 atm y 20 °C es $\nu = 1,01 \cdot 10^{-2}$ cm²/s, y la sección de la tubería $A = 3,44$ cm². Con estos parámetros, y para distintos valores de caudal se ha obtenido la tabla 2.8.

El valor de diseño aceptado para la transición hacia la turbulencia para tubos comerciales es del orden de 2300. Por tanto, caudales por debajo de 30 cm³/s nos proporcionarán flujo laminar y por encima de 50 cm³/s flujo turbulento. Comparando esta tabla con la 2.5, podemos observar que solamente vamos a tener flujo laminar para posiciones de la válvula muy cercanas al cierre total, y para alturas por debajo del 20 %. Así pues, debemos considerar que en la mayoría de los casos

vamos a tratar con flujo turbulento. Además, en este estudio no se ha hecho referencia a la rugosidad de las paredes. Esta rugosidad reduce la zona de transición hacia la turbulencia, de modo que con más razón podemos considerar que vamos a estar siempre en una situación de régimen turbulento.

Q (cm ³ /s)	V = Q/A (cm/s)	Re
10,0	2,9	602
30,0	8,7	1807
50,0	14,5	3012
70,0	20,3	4216
90,0	26,2	5421
110,0	32,0	6626
130,0	37,8	7831

Tabla 2.8 Número de Reynolds.

La situación para la válvula es aún más restrictiva [29]. Prácticamente siempre vamos a tener régimen turbulento, pues lo expuesto anteriormente, y los obstáculos que posee a la libre circulación del flujo a su través, provocan un adelantamiento rápido hacia la turbulencia. Si el flujo es turbulento, de modo general podemos suponer que el caudal que pasa por la válvula corresponde a la siguiente ecuación¹.

$$Q_v = C_d A \sqrt{\Delta p} \quad (2.17)$$

en la que

Q_v = caudal a través de la válvula

C_d = coeficiente de descarga (adimensional)

A = área de paso

Δp = presión diferencial a través de la válvula.

Manteniendo constante el valor de la carrera en su valor más alto, es decir, para válvula totalmente abierta, que corresponde con la situación de caudal máximo, el área de paso permanecerá constante, de modo que la ecuación 2.17 podemos ponerla

$$Q_v = K \sqrt{\Delta p}$$

Considerando además la fórmula introducida por Torricelli en 1644, aplicada a nuestra instalación, tenemos:

$$Q_v = K_h \sqrt{H} \quad (2.18)$$

donde K_h es una constante con dimensiones de caudal dividido por raíz cuadrada de alturas, y H es la altura del nivel de líquido desde la posición de la válvula. Podemos obtener K_h por experimen-

¹Esta ecuación puede deducirse de la ley de conservación de la masa para flujos incompresibles, que ya obtuvo Leonardo da Vinci en 1500, y de la ecuación que Bernoulli introdujo en 1738.

tación, sin más que medir caudales instantáneos a través de la válvula y alturas en el depósito. Sin embargo, como comentamos anteriormente, no disponemos de ningún dispositivo que mida dicho caudal, de manera que hemos efectuado una medida indirecta. Podríamos haber realizado un experimento análogo al seguido para determinar la característica de caudal efectiva de la válvula, pero ello supondría esperar a que el equilibrio dinámico se alcanzase para cada altura. Este método es desaconsejable, ya que, según observaremos posteriormente, el sistema tiene un tiempo de establecimiento del orden de 3 horas.

Hemos realizado el siguiente experimento: llenamos el depósito hasta una altura de 170,0 cm respecto del sensor de nivel, y medimos el tiempo de vaciado cada 5 cm, hasta los 120,0 cm, pues por debajo de ésta medida es difícil. De este resultado, sabiendo la sección en cada altura (recordemos que la sección no es constante), podemos calcular el caudal medio para cada tramo de 5 cm (según se mostró en la tabla 2.1). Aplicaremos la ecuación 2.18 a la altura correspondiente a la mitad de cada tramo, para obtener el valor de K_h , pero considerando la altura respecto de la válvula.

Altura media (cm)	Tiempo (s)	Caudal medio (cm ³ /s)	K_h (cm ^{2,5} /s)
150,5	46	134,85	10,99
145,5	46	134,02	11,11
140,5	47	130,18	10,98
135,5	47	128,70	11,05
130,5	48	124,74	10,92
125,5	49	121,88	10,88
120,5	50	119,14	10,85
115,5	50	118,84	11,06
110,5	51	116,06	11,04
105,5	52	112,95	10,99

Tabla 2.9 Determinación de K_h .

Calculando la media aritmética obtenemos un valor para K_h , de modo que:

$$Q_v = 10,98\sqrt{H} \quad (2.19)$$

Esta ecuación es válida para la situación de válvula totalmente abierta, es decir, cuando el caudal es máximo. Teniendo en cuenta que la característica de caudal efectiva (figura 2.13) nos proporciona la relación existente entre la carrera de la válvula y el caudal relativo, que llamaremos $Q_{\%}(C)$, podemos obtener el caudal para cualquier carrera y a cualquier altura, según se indica en la siguiente ecuación.

$$Q_v = K_h \frac{Q_{\%}(C)}{100} \sqrt{H} \quad (2.20)$$

Esta ecuación se ha comprobado experimentalmente, y se ajusta bastante bien a la situación real. En particular, puede probarse que coincide con bastante exactitud con los resultados obtenidos para la característica de caudal con respecto a la carrera (tabla 2.5).

Conclusiones

Como conclusiones a este apartado relativo a la válvula de control, podemos citar:

- 1 Se ha obtenido la característica de caudal típica de una válvula de apertura rápida (figura 2.13).
- 1 La relación existente entre la presión sobre el servomotor neumático y la carrera de la válvula, en el estacionario, es lineal, observándose la existencia de histéresis. El comportamiento dinámico puede ajustarse a un sistema de primer orden, con una constante de tiempo de 0,5 s (ecuaciones 2.14 y 2.15).
- 1 Se ha obtenido una relación entre la altura y el caudal máximo a través de la válvula. Posteriormente se ha conseguido una relación que proporciona el caudal a través de la válvula en función de la carrera y el nivel del líquido en el depósito (ecuación 2.20).

Podemos determinar el caudal a través de la válvula en función de la presión sobre el servomotor y la altura del líquido en el depósito. Así pues, el bloque correspondiente a la válvula de control en el diagrama de bloques mostrado en la figura 2.1 queda analizado.

PROCESO

El modelo del sistema puede obtenerse aplicando el principio de conservación de la masa, considerando que el depósito tiene una sección A , con un flujo de entrada Q_i , y un flujo de salida Q_o . La masa que entra en el depósito en un cierto intervalo de tiempo Δt será:

$$Q_i \rho \Delta t$$

siendo ρ la densidad del líquido.

La masa que sale del depósito en el mismo intervalo de tiempo será:

$$Q_o \rho \Delta t$$

La acumulación en el depósito en el mismo intervalo de tiempo será:

$$\frac{d}{dt} (\rho A h) \Delta t$$

siendo h el nivel que alcanza el líquido en el depósito en cada instante de tiempo. Escribiendo para estas magnitudes el principio de conservación de la masa [3], obtenemos la relación:

$$\frac{d}{dt} (\rho A h) \Delta t = \rho (Q_i - Q_o) \Delta t \quad (2.21)$$

Haciendo las mismas consideraciones para la temperatura referidas anteriormente en este capítulo, y suponiendo que el líquido es incompresible, ρ es constante, de modo que podemos poner:

$$\frac{d(Ah)}{dt} = Q_i - Q_o \quad (2.22)$$

Esta ecuación representa el comportamiento dinámico del proceso, en función de los caudales de entrada y salida. Como el caudal de salida depende de la altura en forma no lineal (ecuación 2.20), el comportamiento del proceso será también no lineal.

Altura de equilibrio

En condiciones de equilibrio dinámico la altura no varía, y A sería una constante, de tal forma que tendríamos:

$$Q_i = Q_o$$

De esta fórmula y teniendo en cuenta la ecuación (2.20) del flujo a través de la válvula de control, podemos obtener la altura de equilibrio dados un caudal de entrada y una carrera de la válvula.

$$Q_i = Q_o = K_h \frac{Q_{\%}(C)}{100} \sqrt{H}$$

$$H = \left(\frac{100 \cdot Q_i}{K_h Q_{\%}(C)} \right)^2 \quad (2.23)$$

donde H representa la altura del líquido en el depósito respecto de la válvula de control.

Por ejemplo, para caudales de entrada de 111,0 y 118,6 cm³/s, en situación de válvula totalmente abierta, las alturas de equilibrio serían 119,0 y 133,5 cm, respectivamente, sobre el sensor de nivel. De la misma forma, para un caudal de entrada de 97,2 cm³/s, y posiciones de carrera 4,0, 3,0 y 2,0 mm, las alturas de equilibrio serían 125,9, 165,3 y 324,3 cm, respectivamente, sobre el sensor de nivel. Estos valores teóricos se emplearán en apartados sucesivos.

Análisis de la respuesta temporal

Nuestro sistema posee dos entradas, el caudal de entrada al depósito y la intensidad de control. Esta última se traduce al final en la modificación de la apertura de la válvula. Consideraremos entonces que las entradas en nuestro proceso serán, por un lado el caudal de entrada al depósito, y por otro la carrera de la válvula. Para determinar la respuesta en lazo abierto del sistema para cada una de las entradas, debemos mantener la otra constante.

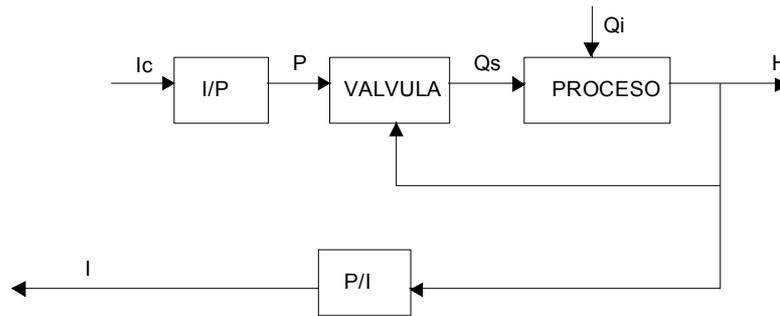


Fig. 2.18 Nuestro sistema en lazo abierto.

Respuesta del sistema en lazo abierto para un salto en el caudal de entrada

Para estudiar la respuesta temporal del sistema en lazo abierto suele someterse al sistema a señales estándar. Normalmente se aplica una señal escalón en la entrada que nos interese. En este caso aplicaremos una señal escalón en el caudal de entrada al depósito, manteniendo la carrera de la válvula constante, en particular, en la situación de apertura total.

La prueba se ha realizado del siguiente modo: para un caudal de 111,0 m³/s se ha esperado a que el sistema alcance el estacionario, con la válvula totalmente abierta. La altura de equilibrio en esta situación corresponde con 119,5 cm de altura respecto del sensor de nivel (el valor teórico es de 119,0 cm, la desviación es muy pequeña). A continuación se aplica un salto en el caudal, hasta alcanzar 118,6 cm³/s, y se representa la respuesta temporal del sistema en el trazador gráfico (la fig. 2.19 muestra una fotocopia reducida de ésta, que no se ha podido reducir más manteniendo una mínima calidad visual), de la misma forma como se ha hecho en otras pruebas. El nuevo equilibrio dinámico se establece para una altura de 133,0 cm. Del análisis de esta gráfica se han obtenido los siguientes resultados:

- Respuesta típica de un sistema de primer orden.
- Retardo no apreciable con la escala de la gráfica, aunque podemos asegurar que es inexistente.
- Constante de tiempo de aproximadamente 41 minutos.
- Ganancia de aproximadamente 2 s/cm².

Respuesta del sistema en lazo abierto para un salto en la carrera de la válvula

En este caso se mantiene constante el caudal de entrada al depósito, que ha sido de 97,2 cm³/s. La carrera inicial de la válvula fue seleccionada en 4 mm, de modo que la altura de equilibrio correspondiente es de 125,0 cm respecto del sensor de nivel (valor según la ecuación 2.23 de 125,9 cm). A continuación se aplicó un salto en la carrera hasta alcanzar los 3 mm, y se registró la respuesta temporal en el trazador gráfico (fig. 2.20). La nueva altura de equilibrio se alcanza para 166,0 cm (el valor teórico es de 165,3 cm). Del análisis de la gráfica se han obtenido los siguientes resultados:

- Respuesta típica de un sistema de primer orden.
- Retardo no apreciable con la escala de la gráfica.
- Constante de tiempo de aproximadamente 56 minutos.

La constante de tiempo es distinta que en el caso anterior pues, como ya se estudió en el apartado relativo a la válvula de control, depende del salto en la carrera.

Análisis del retardo introducido por la tubería de desagüe

Se han realizado distintas pruebas para determinar los posibles retardos introducidos por la tubería de desagüe, unas a caudal de entrada constante, variando la carrera de la válvula, y otras a la inversa. Para el caso en que se varió el caudal a carrera constante no se observaron retardos. Cuando se varió la carrera a caudal de entrada constante, se observaron retardos muy poco importantes, en torno a 2 ó 3 s. Debemos considerar además la existencia de una constante de tiempo en la carrera de la válvula, que es mayor cuanto mayor es la carrera, como ya vimos anteriormente. En nuestra simulación, por lo tanto, introduciremos un retardo de 1 s, y trataremos de que no existan grandes cambios en la carrera.

FIGURA NO DISPONIBLE

Fig. 2.19 Respuesta en lazo abierto para un escalón en el caudal de entrada.

FIGURA NO DISPONIBLE

Fig. 2.20 Respuesta temporal del sistema para un salto en la carrera de la válvula.

3

G2

INTRODUCCIÓN

En este capítulo describiremos de forma breve la herramienta empleada en nuestra aplicación, G2, de Gensym Corporation. No pretendemos hacer aquí un manual de usuario [17], pero nos parece conveniente dar unas breves explicaciones de los elementos fundamentales de la herramienta de trabajo, que servirán para entender mejor cómo se han realizado las distintas partes de nuestra aplicación.

G2 es una herramienta para el desarrollo de sistemas expertos en tiempo real, dirigida a aplicaciones complejas que requieran monitorización continua e inteligente, diagnóstico y control. Está diseñada para realizar fundamentalmente las siguientes tareas:

- Investigar dentro de una aplicación, como lo podría hacer un operador humano, y dirigir en cada momento las tareas de interés.
- Permite elaborar razonamientos y control de eventos dentro de entornos continuamente cambiantes, respondiendo rápidamente a acontecimientos en el momento que ocurren, sin necesidad de mantener un sondeo continuo de los sensores, por ejemplo.
- Aplicación tanto de conocimiento procedural como heurístico basado en reglas.
- Expresar y utilizar relaciones entre objetos.
- Adquirir información de cualquier fuente de datos, tanto local como remota.
- Proveer información y responder a peticiones del usuario.
- Comunicarse con otros G2 y otras aplicaciones (por ejemplo, con programas externos de simulación).
- Integrar herramientas para el desarrollo del interfaz de usuario muy potentes [43].
- Todo ello en un entorno orientado gráficamente y amigable, con un lenguaje natural e intuitivo.

Podemos observar que la mayoría de requerimientos que debían hacerse a un sistema experto, analizados en el capítulo primero, se cumplen con esta herramienta. Por ello no es extraño que aparezcan continuamente en la bibliografía aplicaciones desarrolladas tomando como base a G2.

En la etapa de desarrollo primero se describe cada clase de objeto en la aplicación, sus atributos y características especiales, y como se conectan a otros objetos. Una vez definidas las clases de objetos (depósito, válvula, sensor, controlador, etc.) se crean elementos permanentes de la aplicación situando objetos en un espacio de trabajo (workspace)², y conectándolos entre sí según sus relaciones. El resultado es un diagrama esquemático de la aplicación (fig. 3.1).

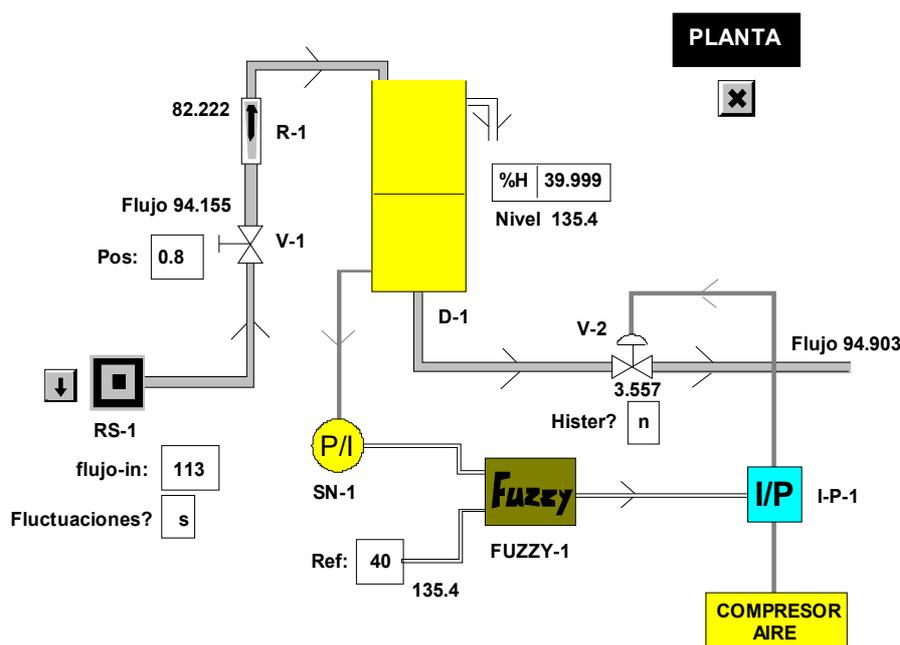


Fig. 3.1 Objetos en una aplicación.

Cada objeto dispone de una tabla que describe sus atributos. G2 automáticamente crea la tabla a partir de la definición de la clase del objeto. La forma en que se representa el conocimiento del sistema es a través de estos atributos, de las relaciones entre ellos y de las reglas y procedimientos, que proporcionan la conexión entre todas las partes de la aplicación. Este enfoque también incorpora diseño modular, que provee una forma eficiente para la adquisición y manejo del conocimiento.

El planteamiento orientado a objetos mejora la consistencia, manejabilidad, entendimiento y actualización sobre la base de conocimiento, de modo que este está perfectamente estructurado [21].

COMPONENTES DE LA BASE DE CONOCIMIENTO

²Siempre que nos sea posible intentaremos emplear la traducción española de los términos específicos de la herramienta. Sin embargo, debido a que la programación, desarrollo y explotación de aplicaciones en G2 se realiza en lenguaje natural en idioma inglés, algunos términos de difícil interpretación o cuya traslación enturbie la explicación, no los traduciremos, si bien, se comentará lo que significan. En el apéndice II se hace una traducción de las expresiones más utilizadas en reglas y fórmulas de simulación.

OBJETOS Y CLASES

Un *objeto* es una representación de alguna parte de la aplicación, por ejemplo depósitos, válvulas, sensores, etc. El dibujo que lo representa se denomina *icono*, y las tuberías y elementos de unión entre ellos, *conexiones*. Cada objeto posee una tabla de *atributos* que contiene conocimiento sobre él, como ejemplo podemos ver la asociada al objeto D-1.

D-1, a depósito	
Notes	OK
User restrictions	none
Names	D-1
Alarma	false
Nivel min	100
Nivel max	182
Nivel sensor min	110
Nivel sensor max	173,5
Posición sobre sensor	100
Estado estacionario	transitorio
Nivel	126.395
Flujo out reb	0

Cada objeto pertenece a una *clase*, que se define a partir de una *definición de objeto* y que especifica su nombre, su clase superior, su icono y las conexiones y atributos comunes a todos los objetos de esa clase, si bien, el valor de estos atributos puede variar de unos a otros. G2 tiene construidas clases básicas, como variables y parámetros, y proporciona otras en forma de librerías de conocimiento.

Las clases están organizadas de forma jerárquica. Cada clase en la jerarquía hereda los atributos, icono y conexiones de su clase superior, aunque pueden ser redefinidos en la actual. Además de los atributos heredados, cada una de ellas puede tener los suyos propios. La jerarquía de clases es parte de la jerarquía de *ítems*, que describe las relaciones entre todos los elementos (objetos, espacios de trabajo, reglas, procedimientos, etc.). Ésta controla como G2 aplica expresiones genéricas. Por ejemplo, una regla que comience por "for any equipo" se aplicaría a todos los objetos de la clase *equipo*, y también a todos los objetos de sus subclases. Esto permite escribir reglas que se puedan aplicar a diferentes tipos de elementos.

VARIABLES Y PARÁMETROS

Se emplean dos tipos especiales de objetos para representar valores en la aplicación. Cuando éstos son solamente válidos durante un determinado periodo de tiempo, deben emplearse *variables*. Si los son indefinidamente, se deben representar con *parámetros*. Esta representación implícita del tiempo es muy importante en aplicaciones en tiempo real [5].

Variables

Uno de los atributos de las variables es su *intervalo de validez*, que especifica el intervalo de tiempo para el cual el último valor registrado es válido. Debido a que G2 puede necesitar encontrar un nuevo valor para una variable, a cada una de ellas se le debe indicar un *servidor de datos*. Posibles fuentes de datos son el motor de inferencias, el simulador de G2, sensores, servidores de datos externos, etc. Cuando G2 necesita un valor para una variable primero chequea si el último

valor registrado ha expirado. Si no lo ha hecho, emplea éste, pero si ya no es válido, busca en su tabla de atributos su servidor de datos, para determinar de dónde puede obtener el valor.

a quantitative variable, the flujo in of RS-1	
Options	do forward chain, depth first backward chain
Notes	OK
User restrictions	none
Names	none
Tracing and breakpoints	defaults
Data type	quantity
Initial value	none
Last recorded value	113, valid indefinitely
History keeping spec	do no keep history
Validity interval	supplied
Formula	none
Simulation details	no simulation formula yet
Initial value for simulation	default
Data server	inference engine
Default update interval	none

En el atributo *opciones* de las variables se le especifica a G2 si debe hacer encadenamiento, hacia atrás o hacia adelante, o búsqueda de datos para obtener un nuevo valor (más adelante estudiaremos el significado de estos términos). Además, las variables pueden tener *fórmulas* y *fórmulas de simulación*, que se pueden emplear para calcular sus valores. Debemos indicar entonces cuando G2 hace una búsqueda de datos:

- Si la variable tiene un intervalo de actualización.
- Si algún visor de la variable tiene un intervalo de actualización.
- Si una regla referencia la variable.
- Si se referencia la variable en procedimientos.

Las variables también pueden recibir valores no solicitados de las siguientes formas:

- A través de encadenamiento hacia adelante, cuando una regla concluye un nuevo valor para una variable lógica o simbólica.
- Cuando G2 recibe un valor no requerido de un servidor de datos.
- A través de un evento de usuario.
- Cuando la variable se activa. Si la variable tiene un valor inicial, este será el que recibe al activarse, y puede provocar encadenamiento hacia adelante.

Parámetros

Los parámetros no inducen una búsqueda de datos, pues siempre tienen un valor. Todas las partes de G2 (el motor de inferencias, el simulador, procedimientos, reglas, etc.) tienen acceso directo al parámetro, y éstos reciben los datos de reglas, procedimientos, fórmulas genéricas o fórmulas de simulación. Pueden emplearse parámetros para representar cualquier tipo de datos, excepto los que se leen de dispositivos externos (deben emplearse variables).

Los parámetros también poseen un atributo *opciones* que controla si G2 puede o no hacer encadenamiento hacia adelante cuando tiene un nuevo valor. Este atributo difiere sensiblemente del correspondiente para las variables, pues no es necesario realizar una búsqueda de datos para obtener su valor. A continuación se presenta la tabla de un parámetro.

a quantitative parameter, the nivel of D-1	
Options	do forward chain
Notes	OK
User restrictions	none
Names	none
Tracing and breakpoints	default
Data type	quantity
Initial value	100.0
Last recorded value	125.896
History keeping spec	keep history with maximum age of data points = 10 minutes

Tanto para variables y parámetros, los tipos de datos que emplea G2 son los siguientes:

- *Simbólico*. Contiene una lista de valores enumerados.
- *Texto*. Contiene una cadena de texto.
- *Cantidad*. Es un tipo de dato cuyos valores pueden ser *enteros* o *float*.
- *Valores de verdad*. Que pueden ser *lógicos* (verdadero o falso), o *difusos* entre -1 (totalmente falso) y +1 (totalmente verdadero).

Debemos mencionar que las expresiones difusas indican un grado de certeza sobre algo. Estas expresiones solamente pueden emplearse en el motor de inferencias, reglas y procedimientos, pero no están disponibles en el simulador, ya que éste emplea exclusivamente valores discretos. Se pueden asignar valores difusos empleando una *banda de certeza difusa*. Por ejemplo, una expresión "temperatura > 50 (+- 5)", asignaría un valor difuso entre -1 y +1 cuando la temperatura estuviese comprendida entre 45 y 55, -1 cuando fuese menor o igual que 45 y +1 cuando fuese mayor o igual que 55 (fig. 3.2). También existen operadores difusos que devuelven el valor lógico verdadero o falso cuando se aplican a dichas expresiones.

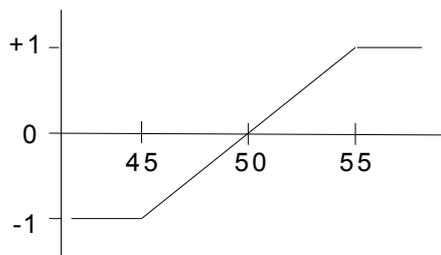


Fig. 3.2 Valor difuso de la expresión "temperatura > 50 (+- 5)".

CONEXIONES Y RELACIONES

Una *conexión* es un elemento que une gráficamente dos objetos, indicando una relación entre ellos. En un entorno físico, una conexión puede ser una tubería o un cable eléctrico, pero también puede representar relaciones abstractas y temporales. G2 permite definir clases de conexiones, referenciándose de un modo intuitivo. Por ejemplo, podemos escribir una regla que haga referencia a cualquier válvula conectada a la entrada de un depósito ("any válvula connected to the input of depósito-1") o incluso a las tuberías que posea un depósito ("any tubería connected to depósito-1"). En estos dos ejemplos, *depósito-1* es un objeto de la clase *depósito*, *válvula* es una clase de objeto y *tubería* es una clase del tipo *conexión*.

Igual que las conexiones, las *relaciones* también representan relaciones entre elementos, y se definen como una asociación entre ellos. Poseen básicamente tres diferencias: las relaciones pueden crearse dinámicamente a través de una acción "conclude", mientras las conexiones se crean manualmente; las relaciones no se pueden mostrar gráficamente, y no se salvan como una parte de la base de conocimiento, mientras las conexiones sí; por último, las relaciones se establecen entre dos elementos cualesquiera, mientras las conexiones solamente pueden ser establecidas entre objetos.

REGLAS

Las reglas contienen conocimiento sobre la aplicación que se pretende resolver, mediante pares condiciones-acciones. Posee fundamentalmente dos partes, un *antecedente* que especifica las condiciones, y un *consecuente* que especifica las acciones que se deben llevar a cabo, de acuerdo con esas condiciones. Cuando G2 invoca una regla, primero evalúa la expresión lógica en la parte condicional del antecedente para determinar si se verifica. Si el resultado es verdadero, se realizan las acciones correspondientes de la parte consecuente. G2 proporciona las siguientes clases de reglas:

- Reglas *if*. Por defecto, G2 invoca solamente este tipo de reglas a través de encadenamiento hacia adelante o hacia atrás.

- Reglas *iniciales*. Se invocan cuando se arranca la base de conocimiento o cuando se activa el espacio de trabajo que contiene la regla, no invocándose otra hasta que se completa. Si no existe ninguna regla de este tipo, se exploran las demás. Estas reglas también pueden hacer encadenamiento hacia atrás a otras, incluso a otras iniciales, para proporcionar valores para variables.
- Reglas *incondicionales*. G2 ejecuta la acción del consecuente cada vez que son invocadas.
- Reglas *when*. Similares a las reglas *if*, excepto que por defecto, G2 no las invoca a través de ningún mecanismo de encadenamiento.
- Reglas *whenever*. Son reglas especiales, dirigidas solamente por eventos, ejecutándose la parte consecuente cuando se producen. Por este motivo no pueden ser invocadas por medio de ningún tipo de encadenamiento.

A continuación explicamos algunos de los atributos más importantes de las reglas. Para disponer de una descripción más detallada puede consultarse la ref. [17].

Opciones: en este atributo se determina si la regla puede ser invocada mediante algún tipo de encadenamiento, y además si puede invocar a su vez otras reglas haciendo encadenamiento hacia adelante, o provocar búsqueda de datos cuando la regla intenta referenciar a una variable que no tiene un valor actualizado.

Intervalo de exploración: este atributo le dice a G2 cada cuanto tiempo debe explorarse la regla. De acuerdo con este atributo, la regla se invoca a intervalos regulares de tiempo.

Prioridad: es un valor entero de 1 a 10, indicando con 1 la máxima prioridad. Éste guía el orden en que deben invocarse las reglas. Es de gran importancia en casos de sobrecarga computacional, pues se ejecutan primero las reglas más prioritarias y luego, si es posible, las menos.

VM-R1, a rule	
Options	invocable via backward chaining, invocable via forward chaining, may cause data seeking, may cause forward chaining
Notes	OK
User restrictions	none
Names	VM-R1
Tracing and breakpoints	defaults
if (the posición-manual PM of V-1 < the pos-min PMIN of V-1) or (PM > the pos-max PMAX of V-1) then conclude that $PM = \max(PMIN, \min(PMAX, PM))$	
Scan interval	none
Focal classes	none
Focal objects	none
Categories	none
Rule priority	5
Depth first backward chaining precedence	1
Timeout for rule completion	use default

Cuando se invoca una regla, G2 pone la llamada en una cola de tareas, la cual se ejecutará tan pronto como sea posible, y cuando esté en lo alto de la cola. Las invocaciones de una regla se planifican generando una llamada para cada conjunto de valores referenciados en la regla (por ejemplo, para una regla que comience con "for any equipo" se genera una invocación para cada objeto de la clase equipo). G2 empieza ejecutando la regla intentando evaluar las expresiones lógicas en la parte antecedente. Cuando G2 evalúa esta parte, puede ocurrir uno de los siguientes resultados:

- Una o varias variables del antecedente no tienen un valor actualizado. G2 pone la invocación en estado de espera y coloca un flag de aviso en cada una de esas variables. Si alguna de ellas recibe un valor, la llamada a la regla ejecuta, intentando completar la parte antecedente.
- Si la evaluación del antecedente da un resultado lógico falso, se completa la invocación sin evaluar la parte consecuente.
- Si la evaluación del antecedente da un resultado lógico verdadero, se ejecutan las acciones de la parte consecuente.

En el caso poco habitual de que transcurra el tiempo máximo para que se complete la regla, ésta se intenta evaluar una vez más, completándose al final aunque no se ejecute la parte consecuente.

PROCEDIMIENTOS

Un procedimiento es un conjunto de instrucciones que se ejecutan secuencialmente. Pueden emplearse en vez de reglas cuando se desee realizar explícitamente una serie de operaciones y controlar el flujo de la ejecución. También se utilizan cuando se quieren llevar a cabo las mismas operaciones con distintos datos. El lenguaje de procedimientos de G2 dispone de las estructuras y la mayor parte de las características de otros lenguajes de programación. También posee unas propiedades específicas para aplicaciones que trabajen en tiempo real. En la siguiente tabla se muestra la definición de un procedimiento.

HIDE-WORKSPACE, a procedure	
Notes	OK
User restrictions	none
Tracing and breakpoints	default
Class of procedure invocation	none
Default procedure priority	6
Uninterrupted procedure execution limit	use default
hide-workspace(bot: class botón, W: class G2-window)	
begin	
hide the workspace of bot on W;	
end	

FUNCIONES

Una función es un conjunto de sentencias que realiza distintas operaciones, devolviendo un valor. Se llama a las funciones cuando su nombre y argumentos, si existen, aparecen como parte de una expresión. Pueden distinguirse distintos tipos de funciones:

- *Aritméticas*. Devuelven un valor numérico. G2 posee muchas funciones aritméticas estándar.
- *Call-funtions*. Llamam a una función específica con sus argumentos.
- *De evaluación de conexiones*. Se emplean para evaluar conexiones de entrada o salida, nombre de la conexión, etc.
- *Externas*: G2 puede llamar a funciones escritas en lenguaje C o FORTRAN, que se guardan en un fichero externo.
- *De texto, símbolos y tiempo*.
- *Tabulares de un argumento*: Son funciones de un argumento que devuelven un valor, el cual puede ser la interpolación simple entre dos valores dados en una tabla. Si no se indica la interpolación, G2 falla al evaluar la función si el argumento no está en la tabla.

FÓRMULAS

Una fórmula es una ecuación que proporciona valores para una variable o un parámetro. Existen los siguientes tipos de fórmulas:

- *Genéricas*, que hacen referencia a variables y parámetros, y siempre de un modo genérico.
- *Específicas*, que se especifican en la tabla de las variables, y se aplican solamente a esas variables. Se emplean si el servidor de datos es el motor de inferencias.
- *Fórmulas de simulación*, que a su vez se dividen en *genéricas* y *específicas*, de la misma forma que antes, si bien, las últimas se emplean cuando el servidor de datos de la variable es el *simulador de G2*.

Las diferencias existentes entre las fórmulas de simulación y los dos primeros tipos son fundamentalmente tres:

- Las fórmulas de simulación sólo pueden referenciar valores simulados de variables, mientras que las primeras pueden hacerlo tanto a valores simulados como no simulados.
- El simulador de G2 evalúa una fórmula de simulación a intervalos regulares de tiempo, sin embargo, solamente ejecuta las demás cuando se necesita un valor para la variable, siendo más eficientes, en relación al tiempo empleado que aquellas.
- Sólo se pueden resolver ecuaciones diferenciales de primer orden con fórmulas de simulación.

Todas las fórmulas de simulación pueden contener no linealidades, permitiendo un gran libertad a la hora de modelar sistemas. No pueden incluir valores dependientes del motor de inferencias o de otros servidores de datos, ya que el simulador de G2 considera que no son

accesibles, y entonces no realiza la búsqueda de datos. También pueden hacer referencia a valores de parámetros, pues en este caso no se realiza una búsqueda de datos. Existen tres tipos de fórmulas de simulación en G2:

- *Dependientes*: Consisten en una expresión aritmética que no puede depender de sus valores anteriores. Es decir, no es una variable de estado que dependa de estados previos.

the flujo of any válvula = the flujo-out of the equipo connected at the input of the válvula, with time increment 6 seconds

- *Discretas*: Se expresan en forma de ecuación en diferencias, donde el nuevo valor de la variable o parámetro se especifica en función de su valor anterior. Debe proporcionarse un valor inicial.

state variable: next value of the nivel N of any depósito D = N + dt*(the flujo-in of D - the flujo-out of D - the flujo-out-reb of D) / seccion, with initial value 0, and with time increment 0.5 seconds

- *Continuas*: Son similares a las discretas, si bien, se emplean para establecer relaciones en términos de ecuaciones diferenciales de primer orden. El método de integración puede ser el de Euler o el de Runge-Kutta de cuarto orden.

state variable: d/dt (the position of any robot) = the velocity of the robot + 3, with initial value 0

A todas ellas se le puede especificar el incremento de tiempo que G2 emplea para reevaluarlas. Si no se le especifica, existe un tiempo por defecto.

EL MOTOR DE INFERENCIAS

La parte más importante de G2 es el motor de inferencias en tiempo real, que razona sobre el estado actual de la aplicación, se comunica con el usuario, o realiza otras actividades relacionadas con la deducción. El motor de inferencias opera sobre las siguientes fuentes de información:

- El conocimiento contenido en la base de conocimiento.
- Valores simulados.
- Valores recibidos de sensores u otras fuentes de datos externas.

Las tareas que lleva a cabo el motor de inferencias son las siguientes:

- Explorar las reglas según el intervalo temporal indicado en las mismas.
- Enfocar a los objetos o clases referidos en las reglas.
- Invocar reglas específicas para una clase particular.
- Invocar reglas cuando sucede algún hecho particular.
- Hacer encadenamiento hacia atrás sobre otras reglas para encontrar valores.
- Hacer encadenamiento hacia adelante cuando se reciben nuevos valores.

- Responder a eventos de usuario, del simulador o de servidores externos de datos.
- Lanzar procedimientos para control secuencial.

Tanto el encadenamiento hacia adelante como hacia atrás son comunes a la mayoría de los motores de inferencia. La exploración, el enfoque y la acción de invocar son específicos para trabajar con aplicaciones en tiempo real.

Exploración de reglas

El motor de inferencias puede repetidamente invocar una regla a intervalos regulares de tiempo. Esto puede realizarse especificándolo en el atributo dedicado a tal efecto; G2 entonces examina la aplicación invocando la regla cada intervalo de tiempo.

Enfoque de reglas

Es un método de concentración de reglas sobre un objeto particular invocando aquellas que estén asociadas con ese objeto. Es decir, cuando se realiza una acción "focus on válvula", se invocan todas las reglas que tienen especificado en su atributo correspondiente a la clase válvula o a una clase superior. Ya se analizó en el capítulo introductorio que muchos autores definen esta característica como metaconocimiento [26].

Invocación de reglas por clases

Es un método de concentración sobre una categoría de reglas invocando todas las que estén asociadas con esa categoría. Por ejemplo, cuando el motor de inferencias realiza la acción "invoke peligro rules", se invocan todas las reglas que tienen *peligro* como categoría de enfoque.

Reglas lanzadas por hechos

Cuando una variable que ha estado esperando por un nuevo valor, lo recibe, el motor de inferencias despierta la regla correspondiente. Por ejemplo, la regla "if the nivel of depósito-1 > the nivel-máximo then inform the operator that 'Depósito-1 está rebosando' ". Para evaluar esta regla, el motor de inferencias debe encontrar un valor actualizado para el nivel del depósito-1. Mientras éste valor no se actualice, no se realiza ninguna acción. Cuando se recibe el dato, despierta la regla, informando al operador que depósito-1 está rebosando. La cantidad de tiempo que el motor de inferencias espera por un valor depende de un parámetro de la aplicación.

Encadenamiento de reglas

El motor de inferencias de G2 emplea dos mecanismos de encadenamiento para invocar reglas: hacia atrás y hacia adelante.

Como comentamos anteriormente, G2 debe encontrar los valores para las variables referenciadas en el antecedente de la regla. Si el valor de la variable no se proporciona por un sensor o por una fórmula, entonces el motor de inferencias emplea *encadenamiento hacia atrás* sobre otras reglas (que tenga la facultad de invocarse de esta forma) que puedan suministrar un

valor para la variable en su parte consecuente. Por ejemplo, sea una válvula a la que se le chequea su estado para determinar si está funcionando correctamente.

```
for any válvula
if the posición of the válvula is cerrada
and the flujo of the válvula >0
then conclude that the estado of the válvula is defectuosa
```

Para evaluar el antecedente de esta regla, el motor de inferencias necesita conocer la posición y el flujo de la válvula, y puede provocar encadenamiento hacia atrás para proporcionar valores a estas variables, por ejemplo, a la siguiente regla:

```
if exist an equipo connected at the input of any valvula
then conclude that the flujo of the válvula = the flujo of the equipo
```

No obstante, debe indicarse en la tabla de la variable si debe hacer encadenamiento hacia atrás, con una búsqueda *primero en anchura* o *primero en profundidad*. En la búsqueda primero en profundidad, el motor de inferencia recoge todas las reglas que puedan proporcionar un valor para la variable, ordenándolas de acuerdo con su precedencia. Las reglas con mayor precedencia se invocan primero, y las menor solamente si aquellas no han podido suministrar un valor. Esta es una forma para determinar el orden exacto en que las reglas deben ser lanzadas. En el caso de búsqueda primero en anchura, el motor de inferencias lanza todas las reglas que puedan proporcionar un valor para la variable. Cuando una de ellas suministra un valor, retira las peticiones de las restantes.

El motor de inferencias emplea *encadenamiento hacia delante* para invocar a una regla cuando al menos una de las condiciones de su parte antecedente es satisfecha por otra regla. Por ejemplo, si una regla concluye que el estado de una válvula es defectuoso, puede invocar la siguiente regla:

```
if the estado of v-1 is defectuoso and (...) then inform the operator that "Fallo en V-1"
```

Esta actividad es una forma de razonamiento deductivo, de forma que el motor de inferencia realiza acciones cuando se concluyen acciones de otras reglas. También puede hacerse encadenamiento hacia adelante a la propia regla; puede ser interesante entonces emplearlo para construir bucles.

EL SIMULADOR DE G2

El simulador de G2 es una clase especial de servidor de datos que proporciona valores simulados para variables y valores para parámetros, de tal modo que podemos modelar distintos componentes de aplicaciones. Está totalmente integrado con G2, aunque puede pensarse como un servidor de datos aparte que provee valores simulados. Sus características principales son:

- Al estar integrado con G2, se pueden especificar fórmulas de simulación en las tablas de las variables.

- Es capaz de evaluar ecuaciones algebraicas, en diferencias y diferenciales de primer orden, de forma que se pueden modelar sistemas continuos y de estados discretos.
- Permite distintos incrementos temporales para diferentes variables.
- Puede correr en paralelo con la aplicación, simulando valores mientras G2 controla operaciones reales.

El objetivo principal del simulador es verificar que la base de conocimiento está funcionando adecuadamente, por ejemplo, antes de decidir su conexión a un sistema real. Debido a su facilidad de ejecución en paralelo con la aplicación (modo on-line), el simulador está capacitado para realizar las siguientes tareas:

- Verificar la buena actuación de sensores, comparando valores reales y valores simulados, de modo que si un sensor falla, continuar las operaciones con valores simulados.
- Estimar valores que no pueden ser medidos por sensores, y emplearlos en el flujo de control.
- Mantener valores simulados de una variable para ayuda en el diagnóstico de fallos.

En modo off-line, el simulador puede modelar características dinámicas de objetos y de otros componentes del sistema, diseñar y verificar estrategias de control. Nuestro trabajo se ha encaminado en este sentido: modelar cada elemento de un sistema general, sus características dinámicas, diseñando estrategias de control, y comprobando que las partes modeladas se ajustan a la realidad. Nos hemos servido del simulador de G2 para proporcionar datos a algunos de esos elementos, es decir, a través de alguna fórmula de simulación para sus atributos. Después hemos recurrido al motor de inferencias para modelar, por medio de reglas, el comportamiento de otros.

En G2 pueden diseñarse simulaciones superficiales o profundas, como se mencionó en la parte de simulación del capítulo primero. Ambos tipos son muy útiles para explorar diferentes estrategias de control, implementando en el sistema real la óptima. Además, G2 permite reunir distintas partes de una simulación en un modelo separado, pudiéndose ejecutar procedimientos específicos para cada uno de ellos. Esta es una característica muy potente.

El simulador puede proporcionar valores para variables y parámetros, manteniendo historias de valores simulados, mientras se emplean otros servidores de datos para valores reales. En G2 cada variable puede tener dos conjuntos de valores distintos, su valor en G2 (proporcionado por el motor de inferencias, por servidores de datos externos, sensores, etc., o también por un programa de simulación externo) y su valor simulado (siempre proporcionado por el simulador de G2). Un parámetro solamente puede disponer de un conjunto de valores, como ya sabemos, y pueden venir dados por acciones "conclude", acciones "set" en procedimientos de simulación, fórmulas genéricas y fórmulas genéricas de simulación. Debido a que un mismo parámetro puede recibir valores de varias fuentes, sobrescribiendo el valor anterior, es aconsejable controlar el flujo de asignación de valores para evitar situaciones impredecibles.

OTRAS CARACTERÍSTICAS

En G2 una aplicación habitualmente no se escribe como un bloque único, sino que puede dividirse en *módulos*, que permiten comprobaciones más rápidas y precisas. De esta forma, varios programadores pueden trabajar sobre la misma aplicación, actualizando distintas partes de la misma, pero sin interferencia entre ellos. La forma en que se permite la utilización de distintos módulos, es mediante los *espacios de trabajo*, que pueden visualizarse independientemente, existiendo conexiones implícitas y explícitas entre ellos (postes de conexión). Además, se permite almacenar distintas partes de conocimiento en distintas bases (distintos archivos), que puedan ser agregadas posteriormente a una principal.

Por otra parte, el entorno gráfico de desarrollo de G2 es fácil de emplear y entender, al ser programable en forma de *lenguaje natural* en idioma inglés, con un *editor de texto* interactivo y un *editor de iconos* sencillo, que permite el desarrollo relativamente rápido de aplicaciones. Además, tiene creados objetos específicos, como botones de entrada de datos o de acción, controles, mensajes, visores (gráficos, tablas de lectura, etc.) y otros, que hacen la programación muy sencilla [2]. El programador utiliza estos elementos para crear la interacción específica con el usuario final, que es esencial en el diseño de este tipo de productos. Así pues, no debe entenderse como un mero añadido, sino que es el "escenario" en el que juegan su papel los aspectos psicológicos que aparecen en toda interacción persona4máquina. Este aspecto es crucial al estar directamente vinculado con el uso efectivo de las aplicaciones [43].

4

SIMULACIÓN

Una vez obtenido el conocimiento propio del proceso y del correspondiente a los demás elementos que componen nuestra instalación, vamos a realizar un modelo de simulación, sobre el que estableceremos las pruebas experimentales que hemos practicado sobre el sistema real. Podría resultar sencillo, después de elaborar la descripción de cada dispositivo y del proceso en sí, emplear un lenguaje de simulación específico, e incluso uno de programación genérico (utilizando alguna sencilla técnica de integración numérica), para llegar a modelar todo el sistema. Sin embargo, hemos decidido formalizar la simulación en el marco de los sistemas expertos descritos en el primer capítulo. Los motivos han sido principalmente dos: el primero, las ventajas que supone tener la base de conocimiento separada totalmente de los mecanismos de inferencia y control, y que ya se comentaron allí; y segundo, dar rendimiento y observar las capacidades de la herramienta G2, que ya se analizó en el capítulo tercero. Debido precisamente a este último motivo, en el desarrollo del tema debe tenerse muy presente lo comentado de G2 en dicho capítulo.

REPRESENTACIÓN DE LA APLICACIÓN

Recordemos aquí que uno de los métodos que emplea G2 para representar distintas partes de una aplicación es mediante espacios de trabajo, lo cual posibilita una organización eficiente de toda la base de conocimiento. En la figura 4.1 se presenta la jerarquía empleada en nuestra aplicación, y en la 4.2 el espacio de trabajo denominado "DEFINICIONES".

Debemos señalar que, además de los objetos principales que hemos creado y que se corresponden con los de la instalación real, se han empleado otros, como los que descenden de la clase BOTÓN, y de los que no hacemos una descripción detallada por no ser éste el objetivo fundamental de nuestro trabajo. No obstante, se presentan dos de ellos, por su eficacia en cuanto a la manipulación de espacios de trabajo. Son los objetos llamados HIDE-WS-BOTON y GOTO-SUBWS-BOTON, que tienen asociados procedimientos que se lanzan cada vez que son activados.

-  HIDE-WS-BOTON: El procedimiento asociado hace que se oculte el espacio de trabajo donde está situado el botón.
-  GOTO-SUWS-BOTON: Al activarse, aparece su subespacio de trabajo, si lo posee.

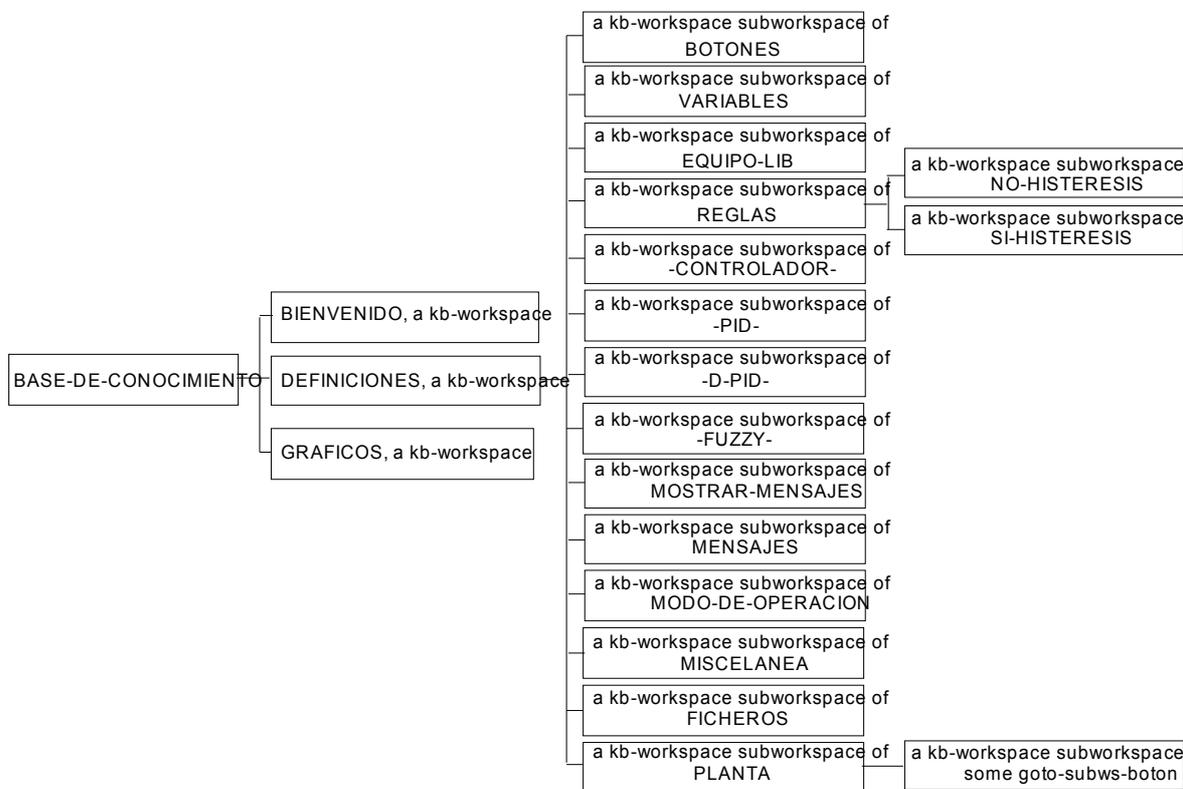


Fig. 4.1 Jerarquía de espacios de trabajo de nuestra aplicación.

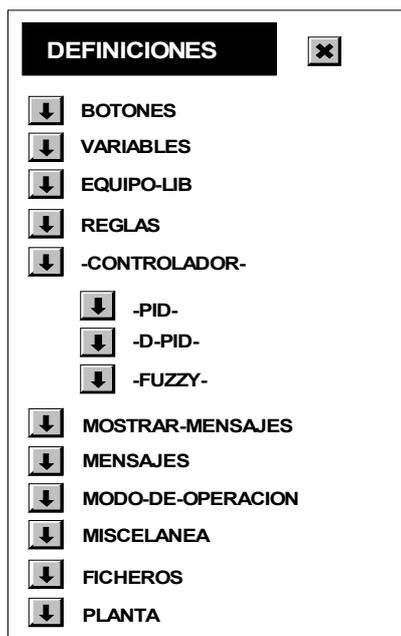


Fig. 4.2 Espacio de trabajo DEFINICIONES.

A continuación se presenta el espacio de trabajo llamado "PLANTA", en el que se aprecia como los elementos del sistema real se corresponden con los objetos definidos sobre éste. Además se observan las conexiones existentes, que se han diferenciado en tres tipos, para corresponder con el sistema real, y que se detallarán más adelante.

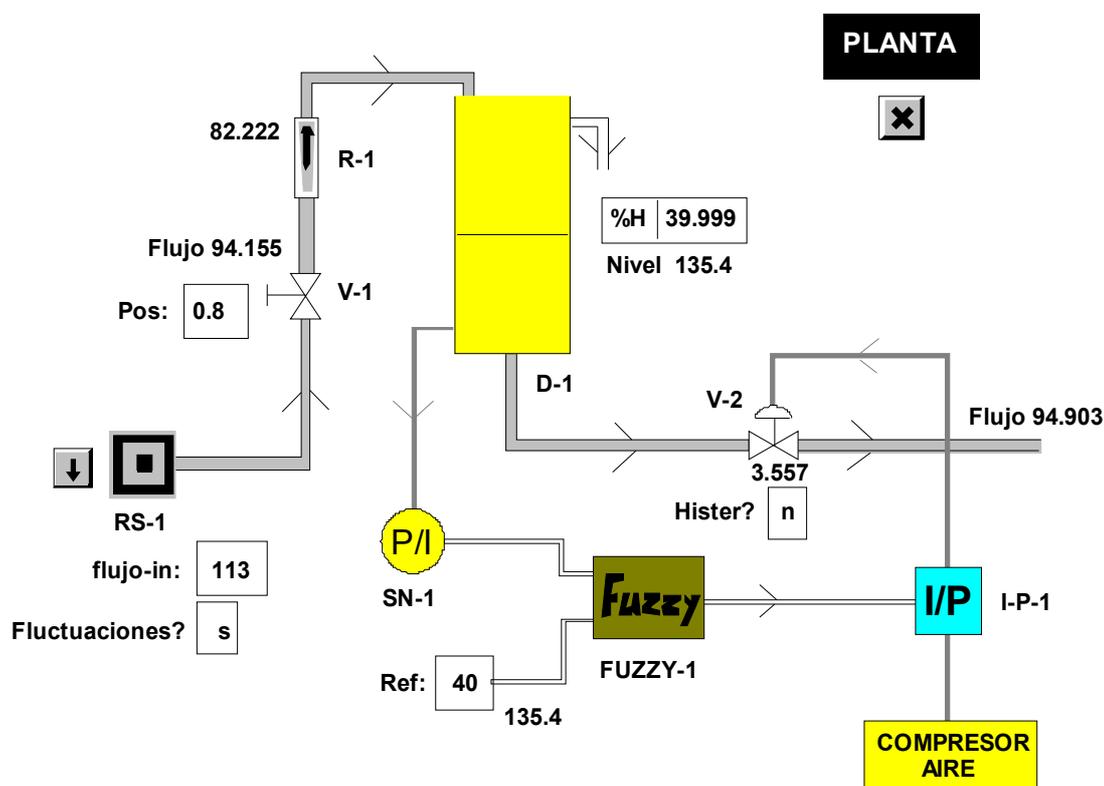


Fig. 4.3 Objetos principales de la aplicación, en el espacio de trabajo PLANTA.

VARIABLES Y PARÁMETROS

La diferencia fundamental entre variables y parámetros está en que el valor de una variable puede expirar, es decir, en un momento dado puede no tener un valor actualizado, lo cual provoca una búsqueda para proporcionarlo. Contrariamente, un parámetro siempre tiene un valor actual y G2 nunca realiza dicha búsqueda. Por este motivo, y aunque se trate de una simulación, se han empleado variables para representar aquellos datos que en el sistema real estarían directamente conectados con el proceso (a través de conversores de señal, como se estudió para la figura 1.2).

Cada variable y parámetro se define como una clase de objeto específica, con unas determinadas características. Resaltar que todas ellas son números reales. En la figura 4.4 se presenta la jerarquía de clases de variables y parámetros. Las que no están recuadradas son las que incorpora G2. No se han traducido los términos, al ser clases incorporadas por esta herramienta.

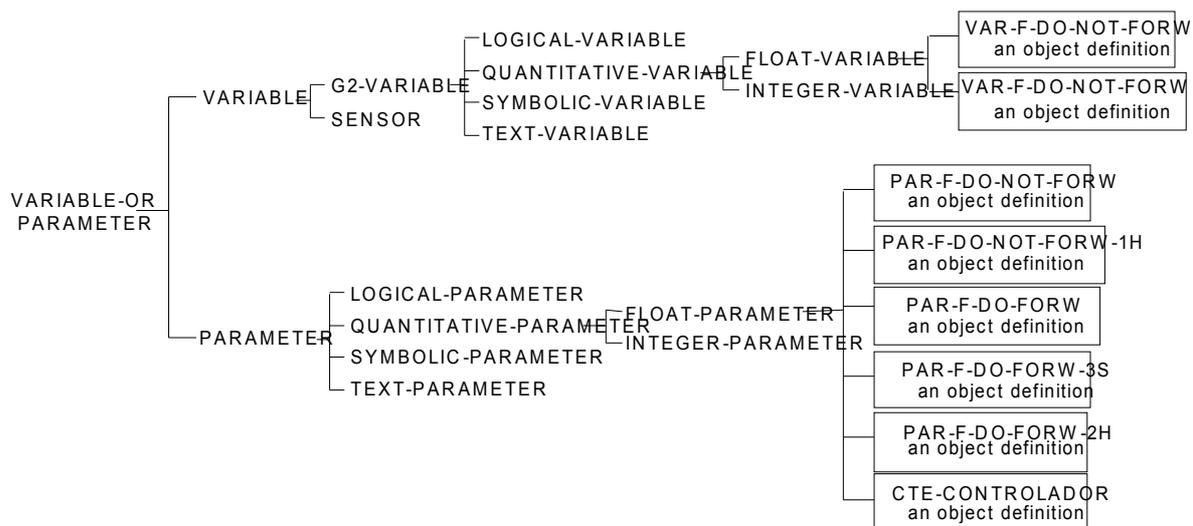


Fig. 4.4 Jerarquía de la clase "VARIABLE-O-PARAMETRO".

VARIABLES

Debemos comentar que se ha seleccionado la opción que posibilita que todas las variables realicen encadenamiento hacia atrás cuando requieren un valor actualizado, y además que éste se produzca primero en profundidad. Describiremos las características de cada tipo de variables que se han definido.

- 1 VAR-F-DO-NOT-FORW-0. A este tipo de variables se le ha prohibido hacer encadenamiento hacia adelante. Además tienen un valor inicial de 0,0.
- 1 VAR-F-DO-NOT-FORW-1H. Tampoco hacen encadenamiento hacia adelante. G2 guarda una historia de un dato de cada una de ellas, necesario para intervenir en gráficos.

PARÁMETROS

Todos los parámetros se han seleccionado para que tengan un valor inicial por defecto de 0,0. Recordamos que no pueden emplear el encadenamiento hacia atrás ni la búsqueda de datos. Seguidamente describimos cada clase utilizada.

- 1 PAR-F-DO-NOT-FORW. Para este tipo de parámetros se ha inhibido la posibilidad de provocar encadenamiento hacia adelante.
- 1 PAR-F-DO-NOT-FORW-1H. Análogo al tipo anterior, pero G2 guarda un histórico de un dato, necesario para intervenir en gráficos.
- 1 PAR-F-DO-FORW. Este tipo provocará encadenamiento hacia adelante cuando reciba un nuevo valor.
- 1 PAR-F-DO-FORW-2H. Es igual que el tipo anterior, pero mantiene un histórico con dos datos.

1 PAR-F-DO-FORW-3S. También provoca encadenamiento hacia adelante. Se mantiene una historia de los valores de los datos de los últimos 3 segundos.

1 CTE-CONTROLADOR: No tiene ninguna opción especial por defecto, pero se pone por separado para posteriormente hacer referencia explícita a este tipo de parámetros.

El empleo de tipos predeterminados de datos por una parte facilita la estructura de los atributos de los objetos que se van a emplear en la aplicación, y por otra, pueden referenciarse de forma explícita, como por ejemplo en la regla que se muestra a continuación, sin tener que especificar cada dato por separado, lo cual constituye una ventaja considerable. Esta regla fija los valores de los coeficientes de la ecuación en diferencias del PID cuando alguno de sus atributos recibe un valor.

```

whenever any cte-controlador receives a value then
  conclude that the q0 of every PID P = the kp of P + 0.5*the mt of P*the ki of
P + (the kd of P/the mt of P) and
  conclude that the q1 of every PID P2 = 0.5*the mt of P*the ki of P - the kp
of P - 2.0*(the kd/the mt of P2) and
  conclude that the q2 of every PID P3 = the kd of P3/the mt of P3

```

CONEXIONES

Una conexión es un elemento que une dos objetos, indicando una relación entre ellos. Se han definido tres tipos distintos, que derivan de la clase CONNECTION. Todas ellas son conexiones dirigidas, es decir, se establecen con un cierto sentido. Automáticamente se crean los bloques de unión entre conexiones, para el caso que dos de ellas tuviesen que acoplarse.

1 TUBERÍA: Une dos objetos que se relacionan a través del flujo de líquido entre ellos.

1 SEÑAL-INSTRUM: Representar señales eléctricas (en las que interviene el controlador).

1 SEÑAL-TRANSMISIÓN: Son las señales de transmisión hacia el sensor y el elemento final de control.

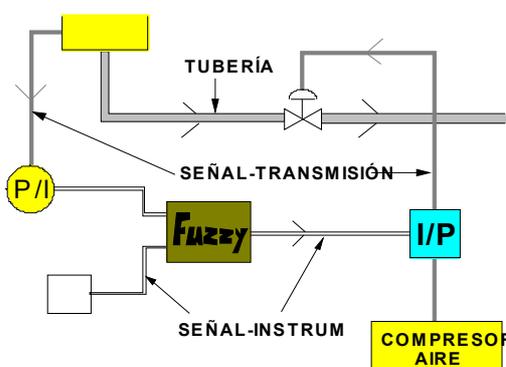


Fig. 4.5 Tipos de conexiones empleadas.

OBJETOS Y CLASES

En este apartado describiremos los objetos que se han creado en nuestra aplicación, y que se relacionan con los dispositivos descritos en el capítulo segundo. Ya dijimos que la forma en que G2 representa el conocimiento del sistema, además de con reglas y procedimientos, es a través de los atributos de cada clase de objeto. Este planteamiento estructura el conocimiento de una forma eficiente, pues al establecerse una jerarquía de clases, se posibilita la referencia de forma genérica a un gran número de elementos pertenecientes a distintas clases dentro de la jerarquía.

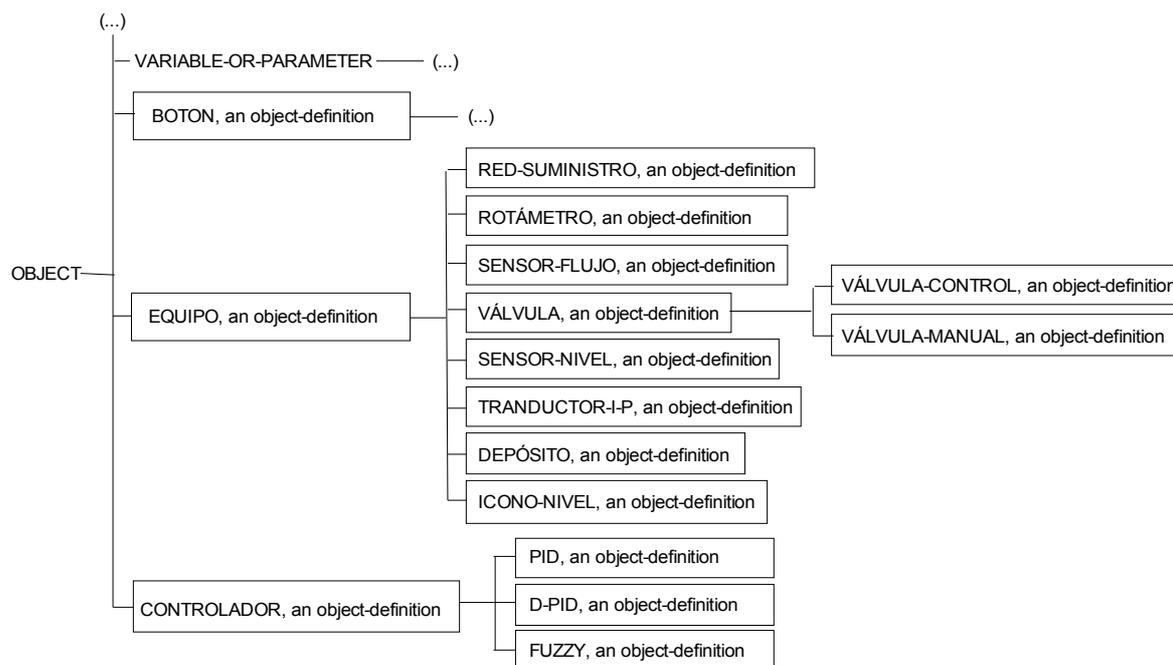


Fig. 4.6 Jerarquía de clases.

EQUIPO

Se emplea como clase genérica de los elementos de la instalación, y aunque solamente posee un atributo, puede hacerse alusión a otros atributos de alguna de sus clases descendientes, en la forma que comentamos antes. Tal es el caso de la regla siguiente, que hace referencia al atributo "flujo" de equipo, cuando éste no lo posee, pero sí el objeto conectado a la salida de V-1, y que pertenece a una clase descendiente de equipo.

whenever (...) then (...) and conclude that the flujo of the equipo connected at the output of V-1 = FO * the posición-manual of V-1

1 ALARMA: Es un parámetro lógico. Indica una situación de alarma, que puede ser utilizada para lanzar procedimientos especiales que eviten un fallo en algún elemento de la instalación, o también advertir al operador de tal situación. En este sentido, la monitorización de señales es una de las aplicaciones importantes a las que se dedican los sistemas expertos.

RED-SUMINISTRO

Esta clase de objeto se ha creado para simular el comportamiento del caudal de entrada al proceso físico, de valor máximo aproximado 200,0 cm³/s. La circunstancia habitual de variaciones en el flujo, debidas a las fluctuaciones propias de la toma de agua de la instalación, también se han previsto, aunque permitiendo que la red de suministro proporcione tanto valores fijos como variables de forma aleatoria (en torno a un valor asignado) para realizar distintas pruebas. Esto último ha sido posible debido a una característica excepcional de G2: modificar en tiempo de ejecución el texto de cualquier atributo de cualquier objeto. En particular el intervalo de exploración (scan-interval) de la regla que regula las variaciones en el caudal de entrada (regla RS-R3), que estudiaremos en otro apartado. A continuación describimos cada atributo de esta clase:

1 FLUJO-IN: Es un parámetro del tipo PAR-F-DO-FORW, y almacena el valor fijo del flujo de la red de suministro, o el valor en torno al cual se producen las fluctuaciones. Su valor se recoge en un *type-in-box* o botón de entrada de datos. Existe una regla (RS-R2) que fija los límites permitidos para el mismo (entre FLUJO-MIN y FLUJO-MAX). Además invoca las reglas *whenever* RS-R1 y RS-R4. La primera asigna un valor al atributo FLUJO-OUT. La segunda es la que regula la existencia de fluctuaciones. Más adelante se estudiarán en detalle todas estas reglas.

1 FLUJO-OUT: Pertenece al tipo PAR-F-DO-NOT-FORW. Es el valor de salida efectivo de la red de suministro. Este viene dado en principio por la regla RS-R1, como se mencionó anteriormente, pero cuando las fluctuaciones se establecen, su valor lo determina la regla incondicional RS-R3.

1 FLUJO-MAX y FLUJO-MIN: Constantes de valor 200,0 y 0,0 respectivamente. Son los límites permitidos de FLUJO-IN.

1 FLUCTUAR: Es un parámetro simbólico, que determina si se van a producir las variaciones de flujo a las que hacemos referencia. Su valor puede modificarse en un *type-in-box*, lanzando la regla RS-R4 mediante encadenamiento hacia adelante. Sus dos posibles valores son "s" o "n", y por defecto tiene "n".

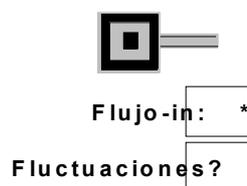


Fig. 4.7 RED-SUMINISTRO

En la figura adlátere se muestra el icono de la red de suministro. Señalar además que posee una tubería de salida que sirve para conectarlo al objeto siguiente. Se indican también los botones de entrada de datos para dar valores a FLUJO-IN y FLUCTUAR.

VÁLVULA

Es una clase genérica que posee dos clases descendientes, que representarán la llave de paso del caudal de entrada y la válvula de control. Posee solamente dos atributos:

- 1 PRES-IN y PRES-OUT. Son parámetros cuantitativos. Realmente no se emplean en nuestra aplicación, pero se incluyen desde un punto de vista general, pues muchos de los problemas que envuelven válvulas se resuelven con las presiones de entrada y salida de las mismas [9].

VÁLVULA-MANUAL

Representa la llave de paso del caudal de entrada al proceso. Se ha supuesto que es una válvula lineal, de modo que su abertura es proporcional al flujo que pasa a su través. Posee los atributos siguientes:

- 1 POSICIÓN-MANUAL. Pertenece al tipo PAR-F-DO-FORW, y representa la abertura de la válvula. Su valor se fija en un *type-in-box*, que además se chequea en la regla VM-R1 para que esté comprendido entre 0,0 y 1,0. Por otra parte, invoca a las reglas *whenever* RS-R4 y VM-R2. Esta última establece el valor del atributo FLUJO, y también el flujo del equipo conectado a la salida de este objeto.
- 1 POS-MIN y POS-MAX. Constantes que limitan el valor de POSICIÓN-MANUAL entre 0,0 y 1,0.
- 1 FLUJO: Es del tipo PAR-F-DO-NOT-FORW. Este parámetro es precisamente el flujo que pasa a través de la válvula, cuyo valor viene dado por la regla VM-R2.

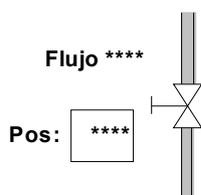


Fig. 4.8 VÁLVULA-MANUAL

En la figura 4.8 se presenta su icono, con dos conexiones de tipo tubería y el valor del atributo FLUJO. También se puede observar el botón que proporciona valores a POSICIÓN-MANUAL.

VÁLVULA-CONTROL

Simula el comportamiento de la válvula de control de nuestro proceso. Para concretar sus características físicas se han incluido muchas constantes en sus atributos. Este elemento es uno de los más importantes de la instalación, como ya sabemos. Tiene los siguientes atributos:

- 1 P-CONTROL-MIN-CERRAR. Constante = 12,5.
- 1 P-CONTROL-MAX-CERRAR. Constante = 21,9.

1 P-CONTROL-MIN-ABRIR. Constante = 12,17.

1 P-CONTROL-MAX-ABRIR. Constante = 21,2.

1 CTE-Q-ALTURA. Constante = 10,98.

1 POSICION-SOBRE-SENSOR. Constante = 17,0.

1 ESTADO-CARRERA. Es un parámetro lógico que se emplea en el algoritmo de histéresis, y que precisa el sentido de apertura o cierre en que está la carrera. Su valor se proporciona en la regla VC-R4-1. Dicho algoritmo se ejecuta si el atributo HISTÉRESIS tiene el valor "s".

1 HISTÉRESIS. Es una variable simbólica, de valores "n" o "s", y que hace posible el algoritmo al que hacemos referencia.

1 CARRERA. Es del tipo PAR-F-DO-FORW-2H. Representa la carrera de esta válvula. Tiene una historia de dos datos, que intervienen en el algoritmo de histéresis. Su valor viene dado por las reglas VC-R3-1 y VC-R4-2, que no están activas a la vez. Además es el argumento de la función tabular "caudal-carrera", que proporciona la relación entre la carrera y el porcentaje de caudal, según la tabla 2.5.

1 FLUJO. Es un parámetro PAR-F-DO-NOT-FORW-1H, que recibe su valor por la regla VC-R1, que incorpora la ecuación 2.20. Señalar aquí que se mantiene una historia de un dato, que se emplea en un gráfico, y que interviene en la fórmula de simulación que establece el valor de la altura del líquido en el depósito (ecuación 2.22).

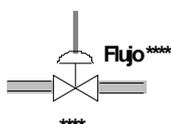


Fig. 4.9 VÁLVULA-CONTROL

En la figura 4.9 se muestra su icono, con sus conexiones, y también los valores de los atributos FLUJO y CARRERA (no se ha añadido el nombre de éste).

ROTÁMETRO

Se corresponde con el mismo dispositivo de la instalación. En nuestro caso es un elemento poco importante, al trabajar directamente con el caudal y no con el valor de conversión de éste a unidades del rotámetro. No obstante, se ha incluido debido a su importancia a la hora de realizar las pruebas experimentales que se hicieron en el capítulo segundo. Posee dos atributos:

1 CONVERSIÓN. Es un parámetro PAR-F-DO-NOT-FORW. Presenta el valor de conversión en función del caudal a través del rotámetro, dada en la ecuación 2.1 (regla R-R1).

1 FLUJO. Es del tipo PAR-F-DO-NOT-FORW-1H. Su valor se proporciona por la regla VM-R2, que se analizó cuando se explicó las ventajas de un planteamiento orientado a objetos.

Este atributo es el flujo de entrada al depósito, y mantiene un histórico de un dato. Interviene en la fórmula de simulación que establece el nivel de líquido en el depósito (ecuación 2.22).

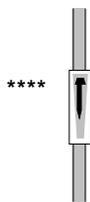


Fig. 4.10 ROTÁMETRO

En la figura adlátere se representa su icono con sus conexiones, y el valor del atributo CONVERSIÓN, aunque sin añadir el nombre de éste.

SENSOR-NIVEL

Este dispositivo se emplea para detectar la altura del líquido contenido en el depósito, y traducirlo a una señal eléctrica según la ecuación 2.3. El objeto que lo representa, simula el nexo de unión entre el proceso y el controlador, y por tanto, se ha empleado una variable para el atributo de este objeto, como ya comentamos anteriormente.

- 1 SALIDA-SENSOR. Es una variable del tipo VAR-F-DO-NOT-FORW-1H. Su valor se suministra por la regla SN-R1, que es la que incluye la ecuación 2.3, que intervendrá posteriormente en el algoritmo de control.

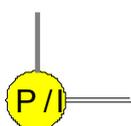


Fig. 4.11 SENSOR-NIVEL

Debemos señalar que este objeto tiene dos conexiones distintas, una SEÑAL-TRANSMISIÓN que proviene del depósito, y una SEÑAL-INSTRUM que lo relaciona con el controlador. Se muestra su icono y dichas conexiones.

TRANSDUCTOR-I-P

Este objeto representa el transductor intensidad-presión, que opera sobre el elemento final de control, de acuerdo con la señal que le llega procedente del controlador. Tiene dos atributos:

- 1 INTENSIDAD-ENTRADA. Es del tipo PAR-F-DO-NOT-FORW-0. Su valor se determina en la regla T-R1, que asigna el valor de la acción de control a este atributo. La limitación entre 4 y 20 mA se produce en el controlador.

1 PRESIÓN-SALIDA. Es un parámetro PAR-F-DO-FORW-2H. Tiene una historia de dos datos que se emplean en el algoritmo de histéresis de la válvula de control, que lanza mediante encadenamiento hacia adelante. Su valor se establece mediante la regla *whenever* T-R2, que corresponde con la ecuación (2.9).

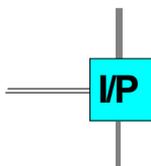


Fig. 4.12 TRANDUCTOR-I-P

Posee tres conexiones, una desde el controlador y otras dos de tipo SEÑAL-TRANSMISIÓN. De éstas, una se ha diseñado para unirlo con el compresor de aire (que en la aplicación no tiene ninguna función), y otra con la válvula de control.

DEPÓSITO

Muchas de sus características vienen dadas por constantes, como el caso de la válvula de control. Se presentan sus atributos:

- 1 NIVEL-MIN. Constante = 100,0.
- 1 NIVEL-MAX. Constante = 182,0.
- 1 NIVEL-SENSOR-MIN. Constante = 110,0.
- 1 NIVEL-SENSOR-MAX. Constante = 173,5.
- 1 POSICION-SOBRE-SENSOR. Constante = 100,0.

1 NIVEL. Es un parámetro PAR-F-DO-FORW-3S. Es quizás el atributo más importante de todos los objetos, pues de él depende la invocación de una gran cantidad de reglas que constituyen la parte fundamental de la simulación. El valor de este atributo lo proporciona una fórmula de simulación, que más adelante estudiaremos.

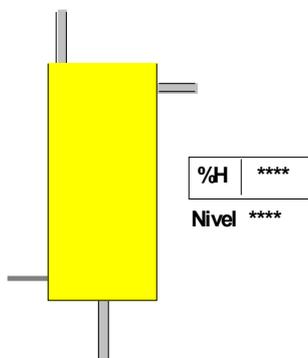


Fig. 4.13 DEPÓSITO

1 FLUJO-OUT-REB. Es de la clase PAR-F-DO-NOT-FORW. Se asigna su valor por la regla D-R2, cuando NIVEL alcanza la altura del orificio de rebosamiento. Interviene también en la fórmula de simulación del nivel.

Este elemento posee cuatro conexiones, según indica la figura 4.13. Una de ellas es una SEÑAL-TRANSMISIÓN hacia el sensor de nivel, y las demás son tuberías. También se muestra el valor del atributo NIVEL, y una tabla de lectura simple, que convierte el nivel en % de altura.

REGLAS

Existen distintos tipos de reglas que hemos empleado en nuestra aplicación. Para las reglas *if* y las reglas *incondicionales*, se ha permitido todo tipo de encadenamiento y búsqueda de datos. Para las reglas *whenever* debemos recordar que están dirigidas solamente por eventos, ejecutándose la parte consecuente cuando se producen, de modo que no pueden ser invocadas por medio de ningún tipo de encadenamiento.

En este punto debemos señalar que en un entorno industrial, las aplicaciones están dirigidas por los datos, donde un experto está continuamente monitorizando el estado de un proceso, y extrayendo conclusiones de las observaciones. Este método es precisamente el que hemos denominado encadenamiento hacia adelante, pues el razonamiento se mueve de las observaciones a las conclusiones, y la mayoría de las aplicaciones on-line, tales como monitorización, manejo de alarmas, control, y planificación se basan fundamentalmente en este tipo de encadenamiento [38]. En G2, con las reglas del tipo *whenever*, se mejora sensiblemente esta capacidad, y por ello en nuestra simulación hemos empleado muchas reglas de este tipo.

Hemos creído preferible clasificar las reglas por los elementos sobre los que se aplican, aunque existe un número reducido de ellas en las que intervienen varios objetos. Seguidamente estudiamos cada regla, los atributos sobre los que se aplican, las funciones que realizan dentro de la simulación, y sus opciones principales. Para una mayor comprensión de las mismas fijarse en el esquema de la figura 4.18 (página 80).

RED-SUMINISTRO

1 RS-R2.

```
for any red-suministro RS
  if (the flujo-in F of RS < the flujo-min FMIN of RS) or (F > the flujo-max
FMAX of RS) then
    conclude that F = max(FMIN,min(F,FMAX))
```

Establece los límites de validez del valor de FLUJO-IN de cualquier red de suministro, dado por sus atributos FLUJO-MIN y FLUJO-MAX. Se invoca vía encadenamiento hacia adelante y tiene mayor prioridad que otras que hacen referencia a dicho atributo.

1 RS-R1.

```
whenever the flujo-in F of any red-suministro RS receives a value then
  conclude that the flujo-out of RS = F
  and change the estado stripe-color of RS to (if F > 0 then the symbol blue
else the symbol grey)
  and change the center stripe-color of the tubería connected to RS to (if F>
0 then the symbol light-blue else the symbol white)
```

Se lanza cuando FLUJO-IN recibe un valor. Además de asignar el valor de FLUJO-OUT, cambia el color del icono y de la tubería conectada a este objeto.

1 RS-R3.

unconditionally
 conclude that the flujo-out FO of every red-suministro RS = $\min(\text{the flujo-max of RS, } \min(1.05 * \text{the flujo-in FI of RS, } \max(0.95 * \text{FI, } \text{FO} + \text{random}(-1.0, 1.0))))$)

Esta regla se ejecuta siempre que es invocada. En nuestro caso no se invoca nunca, sin embargo se lanza debido a que su *intervalo de exploración* (scan-interval) puede tener un valor distinto de 0 segundos. Debe resaltarse que incorpora un algoritmo que establece las fluctuaciones, en torno al valor de FLUJO-IN y con una variación máxima del 5 % en cada sentido.

1 RS-R4.

whenever the fluctuar FU of RS-1 receives a value or the flujo-in F of RS-1 receives a value or the posición-manual PM of V-1 receives a value then
 change the text of the scan-interval of RS-R3 to (if (FU \neq the symbol s) or (F = 0) or (PM = 0) then "none" else "3 seconds")
 and change the center stripe-color of every tubería connected to V-1 to (if F * PM = 0 then the symbol white else the symbol light-blue)
 and change the center stripe-color of every tubería connected to R-1 to (if F * PM = 0 then the symbol white else the symbol light-blue)

Determina la existencia de fluctuaciones en la red de suministro. Puede apreciarse que se cambia el intervalo de exploración (scan-interval) de la regla anterior cuando se satisfacen tres condiciones: FLUCTUAR 1 "s", FLUJO-IN \neq 0, POSICIÓN-MANUAL (de V-1) \neq 0. Esto se hace para mejorar la efectividad respecto del empleo de tres reglas independientes. Esta regla además cambia el color de las tuberías que llevan el caudal de entrada.

VÁLVULA-MANUAL**1 VM-R1.**

for any válvula-manual V
 if (the posición-manual PM of V < the pos-min PMIN of V) or (PM > the pos-max PMAX of V) then
 conclude that PM = $\max(\text{PMIN, } \min(\text{PMAX, PM}))$

Se lanza vía encadenamiento hacia adelante cuando POSICIÓN-MANUAL recibe un valor, chequeándose para ver si está entre sus límites. Esta regla tiene mayor prioridad que otras que puede lanzar este atributo, pues es necesario el chequeo previo antes de realizar ninguna otra tarea.

1 VM-R2.

whenever the flujo-out FO of RS-1 receives a value or the posición-manual PM of V-1 receives a value then
 conclude that the flujo of V-1 = FO * PM
 and conclude that the flujo of the equipo connected at the output of V-1 = FO * PM

Regla de tipo whenever, dirigida por el atributo FLUJO-OUT de la red-suministro RS-1 y por el atributo POSICIÓN-MANUAL de la válvula-manual V-1, y que fija su atributo FLUJO. También se asigna el valor del atributo FLUJO del equipo conectado a la salida de V-1, que es el rotámetro R-1. Esta característica ya se comentó al hablar de la jerarquía de clases.

ROTÁMETRO

1 R-R1.

whenever the flujo F of R-1 receives a value then
conclude that the conversión of R-1 = $\max(10, \min(100, (F - 7.0)/1.06))$

Esta regla específica del rotámetro R-1 precisa el valor de conversión del mismo según la ecuación 2.1. Debemos señalar que se ha limitado el mismo al ofrecido por el rotámetro real, que va desde el 10 hasta el 100 %.

SENSOR-NIVEL

1 SN-R1.

whenever the nivel N of the depósito connected at the input of SN-1 receives a value then
conclude that the salida-sensor of SN-1 = $\max(4.0, (N*16 - 1506)/63.5)$

Está dirigida por el nivel del depósito, que asigna el valor de salida del conversor de presión, (ecuación 2.3). Señalar aquí que se ha limitado la salida del mismo para su valor más bajo (4 mA).

TRANSDUCTOR-I-P

1 T-R1.

whenever the señal-control SC of the controlador C connected at the input of I-P-1 receives a value then
conclude that the intensidad-entrada of I-P-1 = SC

Asigna el valor del atributo INTENSIDAD-ENTRADA del transductor-i-p, que viene dado por la salida del controlador conectado a su entrada.

1 T-R2.

whenever the intensidad-entrada IE of I-P-1 receives a value then
conclude that the presión-salida of I-P-1 = $1.48 * IE + 0.36$

Esta regla proporciona el valor de PRESION-SALIDA, según la ecuación 2.9.

DEPÓSITO

1 D-R1.

```

unconditionally
    conclude that the nivel N of D-1 = min(the nivel-max of D-1, max(the nivel-
min of D-1, N + (the flujo of V-1 - (if the current time >=1 then the value of the flujo
of V-2 as of 1 seconds ago else 0.0) - the flujo-out-reb of D-1)/seccion-depósito
(N)))

```

Esta regla tiene un intervalo de actualización de 1 segundo, pero **no está activa**. Se muestra aquí para indicar que, puesto que empleamos una fórmula de simulación discreta para el NIVEL del depósito, también podría hacerse su simulación empleando una regla. El inconveniente principal es que el tiempo de integración en ese caso sería un valor entero en segundos, imposibilitando emplear fracciones de segundo, lo cual limitaría la aplicación.

1 D-R2.

```

if the nivel N of D-1 > 173 then
    conclude that the flujo-out-reb of D-1 = (if N < the nivel-sensor-max of D-1
then 0.0 else sqrt(N - the nivel-sensor-max of D-1)*10.98)

```

Se invoca vía encadenamiento hacia adelante cuando NIVEL recibe un valor, fijando el de FLUJO-OUT-REB. Se ha empleado para ello la misma constante que la utilizada por la válvula de control. La razón ha sido que las especificaciones de ésta son 3/4 NPT, igual que las tuberías.

1 D-R3.

```

whenever the flujo-out-reb F of D-1 receives a value then
change the center stripe-color of the tubería connected at the out2 of D-1 to (if F >
0 then the symbol light-blue else the symbol white)

```

Cambia el color de la tubería de rebosamiento (out2) cuando FLUJO-OUT-REB tiene un valor.

1 D-R4.

```

unconditionally
    move I-N to (the icon-x-position of D-1, the icon-height IH of D-1 * (N -
100.0)/82.0 + the icon-y-position of D-1 - IH/2.0)

```

Mueve un icono para representar en el espacio de trabajo de forma gráfica el nivel del líquido en el depósito. Tiene un intervalo de exploración de 25 segundos.

VALVULA-CONTROL**1 VC-R1.**

```

whenever the nivel N of D-1 receives a value then
    conclude that the flujo of V-2 = (if N > the nivel-sensor-min of D-1 then
caudal-carrera(the carrera of V-2) * the cte-q-altura of V-2 * sqrt(N - the posición-
sobre-sensor of V-2)/100.0 else 0.0)

```

Calcula el valor del FLUJO de V-2, según la ecuación 2.20. Es lanzada cuando el atributo NIVEL del depósito D-1 recibe un nuevo valor. En ella interviene una función tabular "caudal-carrera", que proporciona la relación entre la carrera y el porcentaje de caudal (tabla 2.5). Esta regla tiene mayor prioridad que otras, y se lanza inmediatamente después de que se tiene un nuevo valor de NIVEL.

1 VC-R2.

```
whenever the flujo F of V-2 receives a value then
    change the center stripe-color of every tubería connected to V-2 to (if F > 0
then the symbol light-blue else the symbol white)
```

Sirve solamente para cambiar el color de las tuberías conectadas a V-2, por eso se le ha asignado una prioridad de 7, es decir, más baja que lo general.

1 VC-R4.

```
if the histéresis of V-2= the symbol s
    then deactivate the subworkspace of NO-HISTERESIS
    and activate the subworkspace of SI-HISTERESIS
```

Esta regla hace lo contrario que la anterior. El objeto SI-HISTERESIS posee un subespacio que contiene las reglas VC-R4-1 y VC-R4-2, que se encargan de hacer la histéresis.

1 VC-R3.

```
if the histéresis of V-2 = the symbol n then
    deactivate the subworkspace of SI-HISTERESIS
    and activate the subworkspace of NO-HISTERESIS
```

Esta regla se lanza vía encadenamiento hacia delante por el valor del atributo HISTÉRESIS, que es el que decide la existencia de histéresis en la válvula. Se ha empleado una característica de G2, que permite activar subespacios de trabajo. Cuando esta variable tiene el valor "n", se desactiva el subespacio del objeto SI-HISTÉRESIS, y se activa el de NO-HISTÉRESIS, que contiene la regla VC-R3-1, y que asigna un valor para la carrera, como se explica más adelante.

1 VC-R4-1.

```
if the carrera C of V-2 /= the value of C as of 1 datapoint ago then
    conclude that the estado-carrera of V-2 = (if C - the value of C as of 1
datapoint ago < 0 then (true) else (false))
```

Se lanza vía encadenamiento hacia adelante cuando la carrera de la válvula tiene un nuevo valor, fijando el atributo ESTADO-CARRERA. Puede apreciarse que se emplea el valor de la misma en el instante anterior, por eso se implementó CARRERA con un historico de dos datos. Esta regla pone el valor lógico verdadero si la válvula está cerrando, y falso si está abriendo.

1 VC-R4-2.

```

if the presión-salida PS of I-P-1 /= the value of PS as of 1 datapoint ago then
    conclude that the carrera C of V-2 = max(0,min(11.1,
    (if the estado-carrera of V-2 then
        (if PS - the value of PS as of 1 datapoint ago > 0 then min(C, -1.18 * PS + 25.86)
        else max( C, -1.23*PS + 26.08))
    else (if PS - the value of PS as of 1 datapoint ago < 0 then max(C, -1.23*PS +
    26.08)
        else min(C, -1.18*PS + 25.86) ) ) ) )

```

Establece el algoritmo de histéresis, en función de ESTADO-CARRERA y de PRESIÓN-SALIDA del transductor-i-p, que la invoca. Se emplea el valor de este último atributo en el instante anterior, por eso tiene un histórico con dos datos. Este algoritmo selecciona una de las dos rectas de la histéresis (ecuaciones 2.10 y 2.11) dependiendo del sentido de apertura o cierre de la válvula.

1 VC-R3-1.

```

if the presión-salida PS of I-P-1 /= the value of PS as of 1 datapoint ago then
    conclude that the carrera of V-2 = max(0, min(11.1, - 1.19*PS + 25.78))

```

Fija el valor de la carrera en función de la recta media obtenida de las dos que intervienen en la histéresis, haciendo que esta no se produzca.

FÓRMULAS DE SIMULACIÓN

Solamente se ha empleado una, que suministra el valor del NIVEL del depósito en función de los caudales de entrada (de R-1) y salida (el que pasa por V-2 y el que sale por el orificio de rebosamiento) y que corresponde a la discretización de la ecuación 2.22.

```

state variable: next value of the nivel N of D-1 = min(the nivel-max of D-1, max( the
nivel-min od D-1, N + DT*(the flujo of R-1 - (if the current time >= 1 seconds then
the value of the flujo of V-2 as of 1 seconds ago else 0.0) - the flujo-out-reb of D-1)/
seccion-depósito (N))), with initial value 100.0, and with time increment DT

```

Debemos comentar que hemos utilizado una fórmula de simulación discreta por la facilidad que supone establecer los límites del NIVEL, y el empleo de una función tabular, que tiene como argumento el propio atributo, y que recoge la característica que se comentó para la tabla 2.1, en la que se estudió la no uniformidad de la sección del depósito.

Además se ha empleado el parámetro DT, que es del tipo PAR-F-DO-NOT-FORW, y que regula el incremento temporal para la simulación. Inicialmente tiene un valor por defecto de 0,5 s, aunque se puede cambiar por medio de un botón de entrada de datos, situado en el espacio de trabajo "MISCELÁNEA". Señalamos también que se ha incluido, de una forma hábil, la característica que se comentó para el retardo de la tubería de desagüe (1 segundo).

VERIFICACIÓN DE LA SIMULACIÓN

Vamos a realizar las pruebas en lazo abierto sobre la simulación, y que básicamente han sido tres:

- 1 Medida del tiempo de vaciado del depósito para distintas aperturas de la válvula de control.
- 1 Respuesta temporal del sistema en lazo abierto para un salto escalón en el caudal de entrada, manteniendo la carrera de la válvula en su valor máximo.
- 1 Respuesta temporal del sistema en lazo abierto para un salto escalón en la carrera de la válvula, manteniendo el caudal de entrada constante.

Los ensayos correspondientes ya se realizaron en el proceso real, y están descritos en el capítulo segundo. El primero cuando se determinó la característica de caudal de la válvula de control; y los otros dos cuando se analizó la respuesta del proceso para variaciones escalón en las señales de entrada. Lo que pretendemos ver es el grado de adecuación que posee la simulación que hemos descrito, con esos ensayos. A partir de unos buenos resultados podremos realizar distintas pruebas para implementar diferentes tipos de controladores sobre la simulación, lo cual puede dar lugar a que posteriormente puedan ser incorporados al control del proceso real.

Todas estas pruebas se realizaron en lazo abierto, de modo que, en nuestro caso, debemos romper el lazo de realimentación. Esto se ha conseguido prohibiendo la actuación de la regla T-R1, que fijaba el valor de la intensidad de entrada al transductor intensidad-presión, sustituyéndola por la regla siguiente:

whenever prueba receives a value then
conclude that the presión-salida of I-P-1 = 1.48 * prueba + 0.36

Se ha creado un parámetro cuantitativo llamado PRUEBA, cuyo valor se determina en un botón de entrada de datos, y que fija, igual que para los casos del proceso real, dicha intensidad de entrada.

1 *Vaciado del depósito*

Las gráficas de la figura 4.15 se corresponden con bastante exactitud a las presentadas en la figura 2.15. Las pequeñas desviaciones se dan precisamente porque hemos utilizado la tabla 2.5 para establecer la relación entre la carrera y el porcentaje de caudal, y no la 2.6, que es la que representa la figura 2.15. Por otra parte, esto pone de manifiesto que la ecuación 2.20 se ajusta muy bien a la situación real.

1 *Escalón en el caudal de entrada*

En la figura 4.16 se aprecia la respuesta del sistema al salto escalón en el caudal de entrada, y que es prácticamente la misma que la de la figura 2.19. Debemos comentar aquí que la realización se ha llevado a cabo con los siguientes datos:

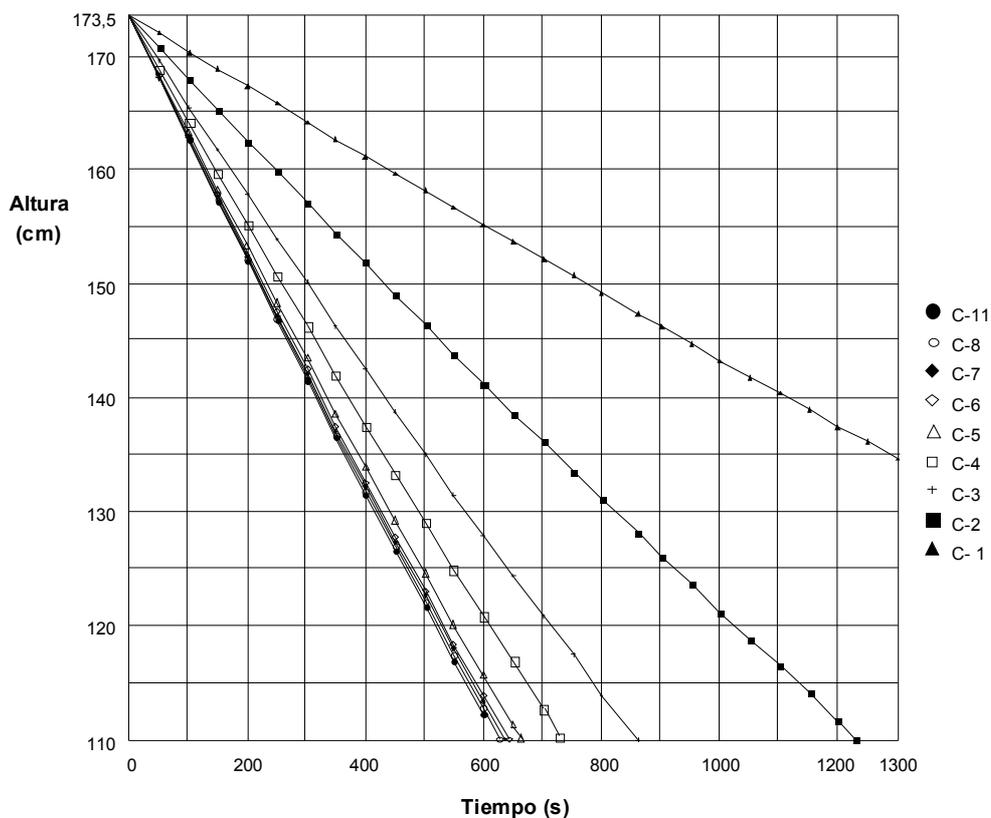


Fig. 4.15 Vaciado del depósito para distintas aperturas de la válvula de control.

- Carrera constante correspondiente a válvula totalmente abierta.
- Caudal de entrada inicial: $111,1 \text{ cm}^3/\text{s}$ Caudal de entrada final: $118,6 \text{ cm}^3/\text{s}$
- Altura inicial: $119,4 \text{ cm}$ Altura final: $133,6 \text{ cm}$

1 Escalón en la carrera de la válvula

La figura 4.17 representa la respuesta para el salto en la carrera, y que encaja con la de la figura 2.20. La prueba se ha realizado con los siguientes datos:

- Caudal de entrada constante de valor $97,2 \text{ cm}^3/\text{s}$
- Carrera inicial: 4 mm Carrera final: 3 mm
- Altura inicial: $126,0 \text{ cm}$ Altura final: $165,2 \text{ cm}$

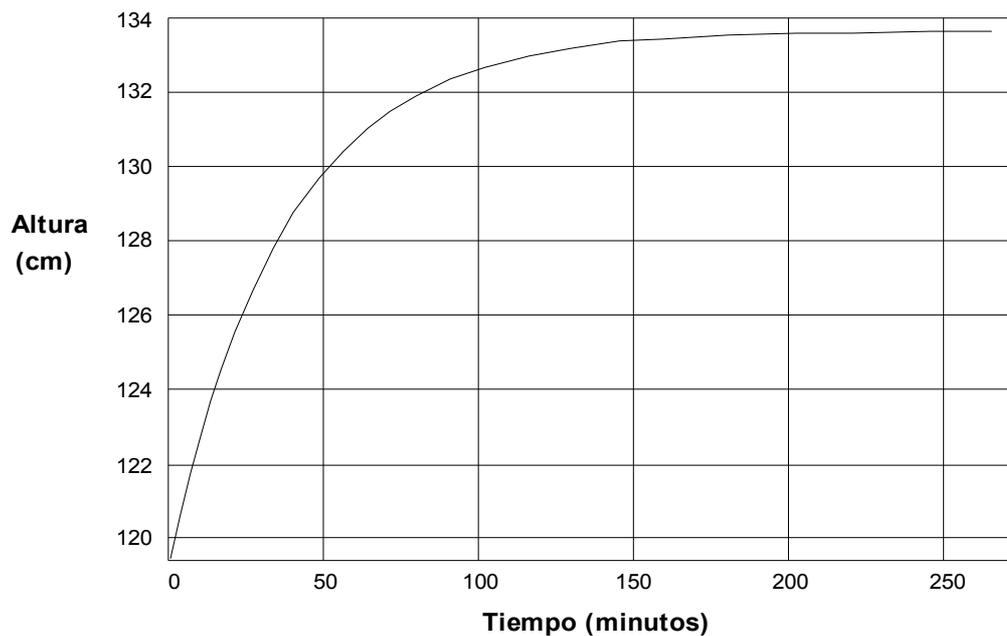


Fig. 4.16 Respuesta del sistema a un escalón en el caudal de entrada, para carrera constante.

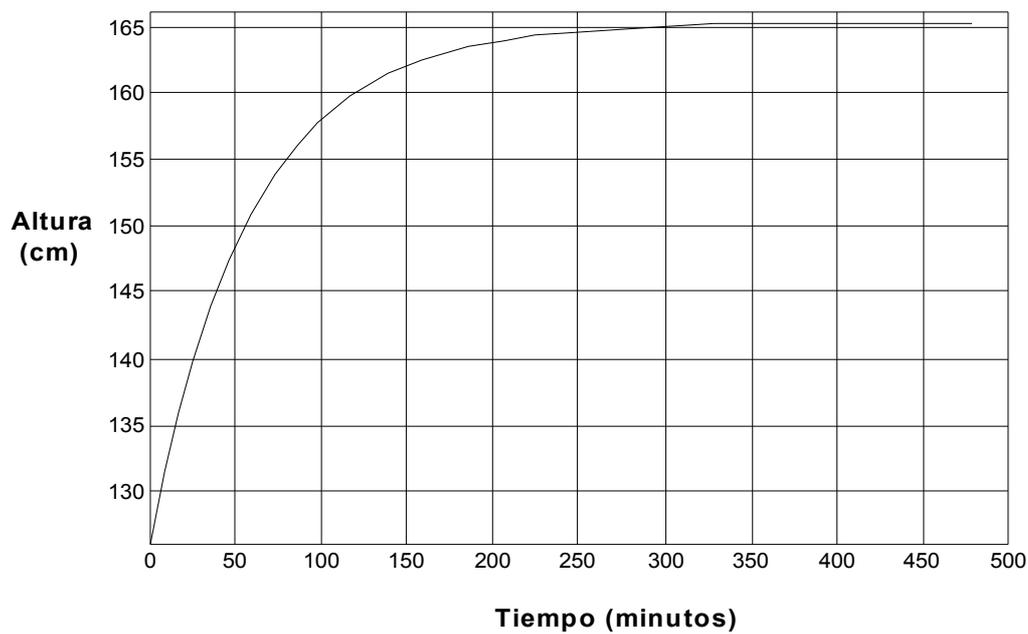


Fig. 4.17 Respuesta del sistema a un escalón la carrera, para caudal de entrada constante.

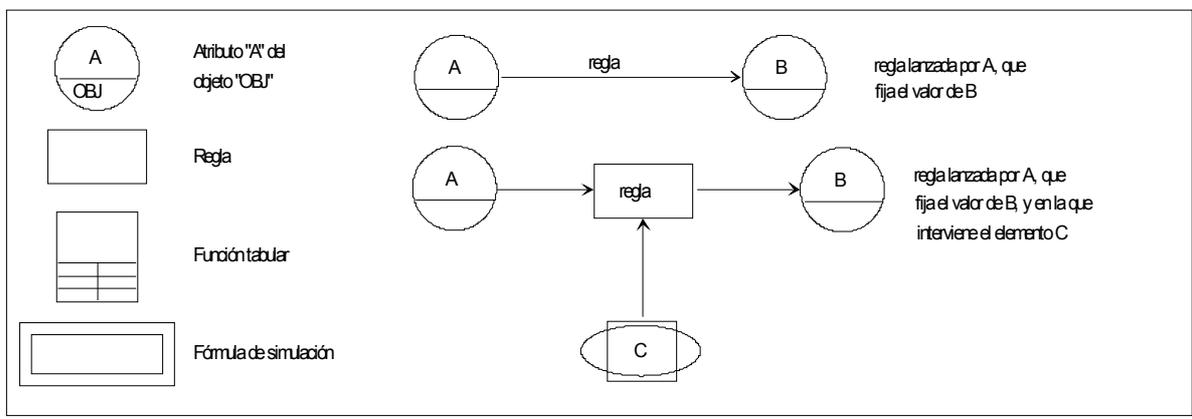
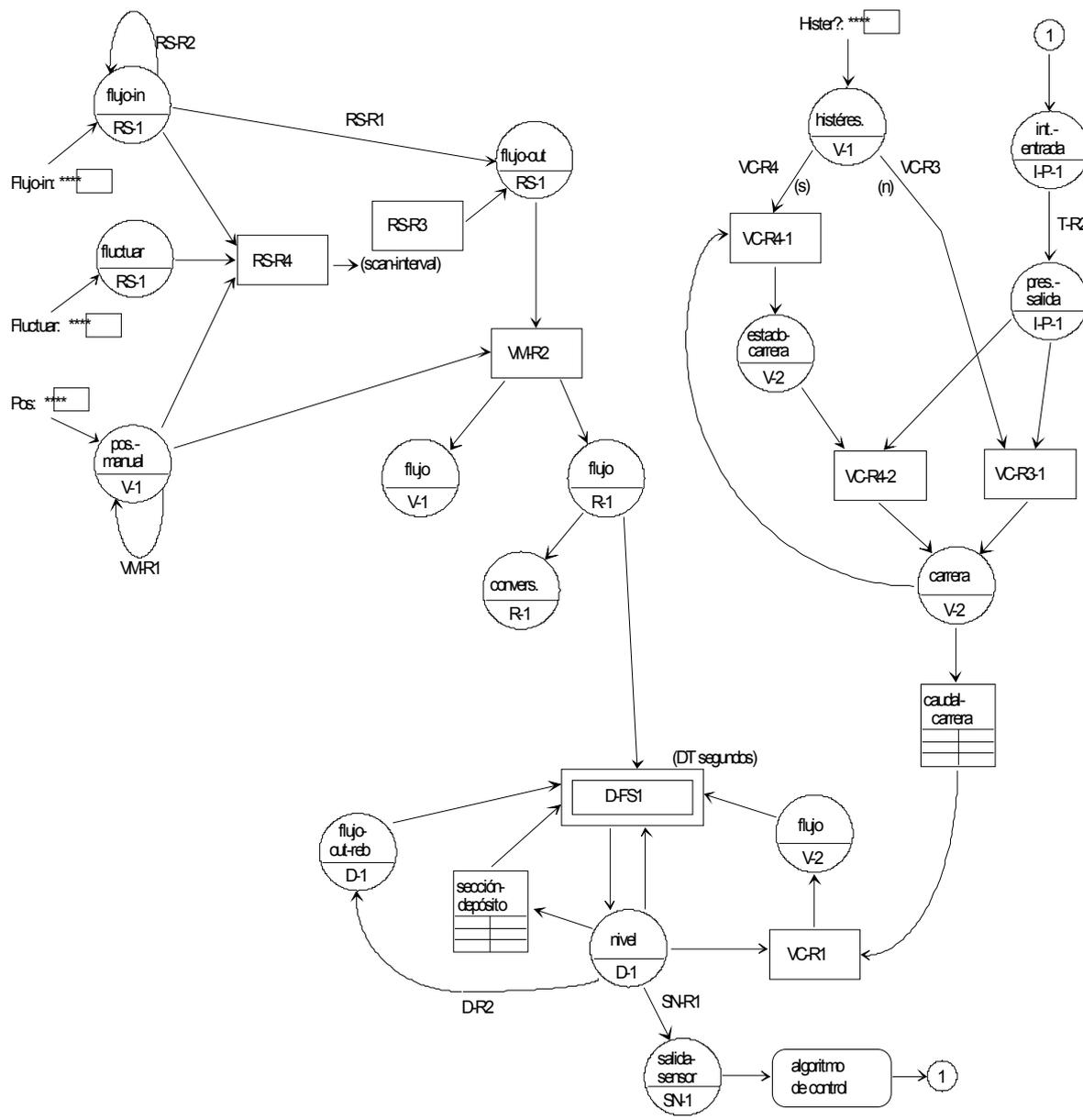


Figura 4.18. Esquema de la simulación.

5

CONTROL BORROSO

INTRODUCCIÓN

Ya se mencionó en el capítulo primero que existe un volumen considerable de conocimiento sobre el control de procesos que suele expresarse en forma imprecisa. Esto hace que la incertidumbre en la información de la base de conocimiento produzca incertidumbre en la validez de las conclusiones. Por otra parte, la mayor parte del conocimiento humano es vago e impreciso y por tanto el pensamiento y razonamiento incluyen información inexacta [46]. Seguidamente se exponen las técnicas principales que se han empleado para tratar con conocimiento inexacto:

1 **Enfoque Bayesiano.** Basada en teoría de probabilidad, trata solamente con incertidumbre [30]. Aunque es el más riguroso, es difícil de emplear en dominios expertos, debido al gran volumen de combinaciones que deben considerarse (algunos autores [20, 7], empleando la suposición de independencia condicional, han llegado a reducir el número de probabilidades estimadas) y a la dificultad de implementar las funciones de densidad de probabilidad cuando los ingenieros y técnicos trasladan su conocimiento [13]. Por último, señalar lo poco adaptadas que están estas técnicas para el tratamiento de problemas dinámicos.

1 **Factores de certeza.** Un factor de certeza $CF(h,e)$ es un valor numérico entre 0 y 1 que indica el grado de confirmación de la hipótesis h basada en la evidencia e . Estos factores se emplean en Mycin [32] para tratamiento de incertidumbre y evidencia (hechos y reglas). Una ventaja sobre el enfoque anterior es que no requiere probabilidades previas y entonces no precisa un gran volumen de datos estadísticos.

1 **Teoría de la evidencia de Dempster-Shafer.** Esta teoría calcula funciones de evidencia. Su mayor problema está en que necesita gran número de procesos computacionales, y en muchos casos la función de evidencia para la inferencia puede ser muy compleja [35].

1 **Lógica borrosa o difusa.** Es un enfoque posibilístico, en el que la fuente de imprecisión se debe a la naturaleza vaga o no bien definida de sentencias tales como:

"Temperatura es medianamente alta".

"Si la temperatura es bastante alta y su incremento es muy grande, entonces disminución grande del caudal".

Estas sentencias involucran una gran cantidad de información. En este modelo, la imprecisión asociada a esos enunciados se trata en la teoría de los conjuntos borrosos, desarrollada por el ruso Lofti A. Zadeh, profesor de Ingeniería Eléctrica en la Universidad de Carolina en Berkeley (EE.UU.), a mediados de los años sesenta. La lógica borrosa surge como un intento de formalización del razonamiento con incertidumbre; en particular, y al contrario que otras formas de razonamiento, intenta abordar problemas definidos en términos lingüísticos, y por tanto imprecisos, donde los datos están expresados en términos cualitativos.

LÓGICAS Y LÓGICA BORROSA

Procedente del inglés, el término *fuzzy* se ha venido traduciendo por difuso o borroso según las preferencias de cada autor. Sin embargo, es una práctica aconsejable no traducir expresiones que aparecen frecuentemente en la literatura, como *fuzzyficación* (borrosificación), *defuzzyficación* (que consecuentemente, deberíamos traducirlo por desborrosificación), etc. Nosotros emplearemos indistintamente las expresiones anteriores.

En un primer acercamiento, la lógica borrosa puede ser contemplada en contraposición a la lógica clásica, empleada en las técnicas de computación usuales [15], donde un elemento de un conjunto se define por su total pertenencia o su no pertenencia. En este sentido, cualquier enunciado o proposición puede tomar un valor lógico VERDADERO o FALSO, en definitiva, 0 ó 1 (parece evidente que una puerta está **abierta** o está **cerrada**). Sin embargo, en la teoría de los conjuntos borrosos, se introduce el concepto de grado de pertenencia, que representa la *posibilidad*, expresada por un valor entre 0 y 1, de que un determinado elemento pertenezca a un conjunto. Según esto, una puerta no tiene porqué estar necesariamente **abierta** (1) o **cerrada** (0), sino que puede estar **medianamente abierta** (0,5), **muy abierta** (0,8), **casi cerrada** (0,1), etc.

La principal ventaja de utilizar términos lingüísticos como los anteriores, está en que permite plantear el problema en los mismo términos en los que lo haría un experto humano. De hecho, el propio Zadeh justifica el inexplicable éxito de sus trabajos con la frase: *El mundo es fuzzy*. En otras palabras, no tiene sentido buscar la solución a un problema no perfectamente definido, por medio de un planteamiento matemático muy exacto, cuando el ser humano es el primero que razona empleando la inexactitud [46].

Lógica

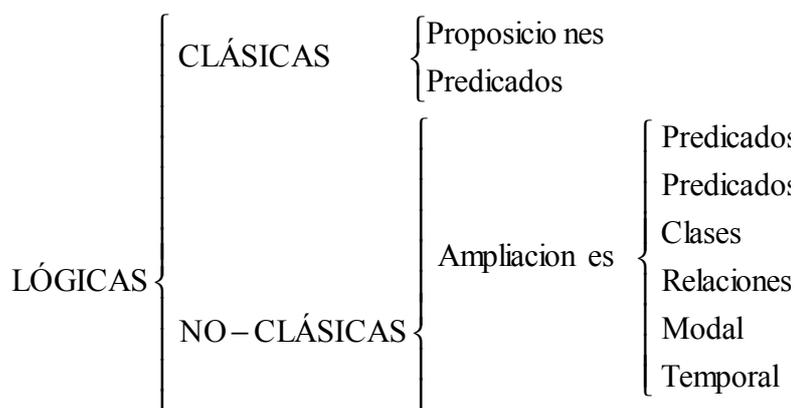
La Lógica es la ciencia que estudia las condiciones formales de validez de una inferencia y, en general, de una argumentación cualquiera [32]. Para definir una lógica hay que empezar por determinar las *variables* (enunciados elementales) y las *conectivas* que pueden unir a éstas para dar lugar a *sentencias* (enunciados compuestos). En un segundo paso hay que especificar *axiomas* como enunciados básicos de la lógica que estemos definiendo y *reglas operativas* que permitan derivar un enunciado de otro. En este punto, se pueden enunciar *teoremas*, que serán sentencias que, mediante la aplicación consecutiva de reglas operativas, pueden obtenerse a partir de los axiomas. Se dice asimismo que una sentencia es una *tesis* si es un axioma o un teorema.

El siguiente paso es la *semántica*, es decir, dar significado o interpretar las sentencias. Para ello hay que definir un conjunto de valores semánticos, con un mínimo de dos elementos, y un conjunto de *operaciones* cerradas con los elementos del conjunto, de modo que a cada variable le corresponda un valor semántico y a cada conectiva una operación. De este modo será posible calcular el valor semántico de cualquier sentencia.

Debemos resaltar, por las implicaciones en el tema que nos ocupa, que aunque el conjunto de valores semánticos ha sido tradicionalmente de dos elementos {verdadero, falso}, desde un punto de vista general no se imponen restricciones al número de elementos, ni siquiera que sea finito.

Cuando en el conjunto de valores semánticos está el valor "verdadero", se define *tautología* como aquella sentencia que para cualquier interpretación de sus variables, su interpretación siempre es verdadera. Cuando la interpretación es siempre falsa se llama *contradicción*. La relación entre sentencias válidas, o tesis, con su valor semántico es lo que lleva a los conceptos de *completitud* y *consistencia* de las lógicas. Si en una lógica toda tautología es una tesis se dice que es completa, y se dice que es consistente cuando toda tesis es una tautología.

Con estos conceptos esenciales se construyen las diferentes lógicas según se definan sus variables y conectivas, sus reglas operativas y su semántica. Podemos seguir el siguiente esquema para agrupar las lógicas más conocidas. Se ha hecho una primera división en lógicas clásicas y no clásicas, agrupando en las primeras las conocidas en la antigüedad y formalizadas en el siglo pasado y en las no clásicas las formalizadas en el presente siglo como derivación de las anteriores.



Revisaremos brevemente las lógicas clásicas, ya que por una parte constituyen la base de las demás, y por otra nos permitirá destacar las diferencias con la lógica borrosa.

Lógica de proposiciones

1 Variables proposicionales: son simples enunciados declarativos ("hoy es martes").

1 Conectivas:

negación	2
conjunción	2
disyunción	2
condicional	1
bicondicional	1

1 Axiomas: Los más conocidos son cuatro:

$(p \supset p) \supset p$
 $q \supset (p \supset q)$
 $(p \supset q) \supset (q \supset p)$
 $(q \supset p) \supset [(r \supset p) \supset (r \supset q)]$

1 Reglas operativas: la fundamental es la operación de *sustitución* por la que una variable proposicional se sustituye por una sentencia. Se podrían definir otras reglas, como la *unión* (si A y B son tesis, entonces $A \supset B$ es tesis) y la de *separación* (si A es tesis y $A \supset B$ es tesis, entonces B es tesis).

1 Valores semánticos: {verdadero, falso} o {1, 0}.

1 Operaciones semánticas: son las del álgebra de Boole [32].

1 Comentarios: esta lógica es completa y consistente, destacando dos teoremas:

T1: $\supset p \supset p$ (tercio exclusivo)
T2: $\supset (p \supset q) \supset (\supset p \supset \supset q)$ (ley de De Morgan)

Lógica de predicados

La idea básica de la lógica de predicados es formalizar el concepto de propiedad ("Alberto es un ser vivo") y el de relación ("Juan vive en Jerez"), ampliando elementos de la lógica de proposiciones.

1 Variables: se dividen en dos tipos,

- *colectivas* ("los peces"), empleándose la notación usual x,y,z para miembros genéricos del colectivo, denominadas variables propiamente dichas. Se denomina universo de discurso al conjunto de posibles valores particulares que pueden tomar dichas variables.
- *miembros* ("Juan"), con la notación usual a,b,c , denominadas constantes.

1 Conectivas: se definen tres nuevas respecto de las anteriores,

- *propiedad* (predicado monádico), por ejemplo, "Alberto es un ser vivo": $S(a)$
- *relación* (predicado poliádico), por ejemplo, "Juan vive en Madrid": $V(j,m)$
- *cuantificador*, que se dividen en universal (para todo elemento ... se verifica ...) y existencial (existe un elemento ... tal que ...).

1 Axiomas: se añaden dos nuevos:

$$\begin{aligned} & \exists x (P(x)) \wedge P(a) \\ & \exists x (p \wedge P(x)) \wedge (p \wedge \exists x (P(x))) \end{aligned}$$

1 Semántica: se conserva la anterior, extendiéndose a los cuantificadores. La interpretación del cuantificador \exists corresponde a la interpretación de la conjunción de todos los posibles miembros, mientras que la interpretación de \wedge es el de la disyunción.

1 Comentarios: la lógica de predicados también es completa y consistente.

LÓGICA BORROSA

La lógica borrosa surge como un intento de formalizar el razonamiento con incertidumbre, y abordar problemas definidos en términos lingüísticos, y por tanto imprecisos. El primer punto que debe afrontarse es el de la representación matemática de dichas expresiones. Podemos asegurar que un hombre es **alto** si mide 1,90 m. Una estatura de 1,75 m no correspondería a un varón alto, sino a uno de estatura **normal**. De esta forma, "1,90 es alto" es cierto al 100 %, "1,75 es alto" es cierto al 0 %, pero "1,75 es normal" es cierto al 100 %. Pues bien, la lógica borrosa consiste en asignar a estaturas comprendidas entre 1,75 y 1,90, valores borrosos entre 0 y 1.

La lógica borrosa tiene por una parte el mismo aspecto sintáctico que la lógica de predicados y por otra su semántica se basa en el concepto de borrosidad que se formaliza en la teoría de conjuntos borrosos.

Conjuntos borrosos

Para abordar la teoría de los conjuntos borroso es preferible olvidarse, por lo menos durante un tiempo, de nuestros conocimientos de la teoría clásica de conjuntos, ya que la idea de partida es completamente distinta. En la teoría clásica un elemento cualquiera o bien pertenece a un conjunto o bien no pertenece al mismo. En teoría borrosa un elemento siempre pertenece en cierto grado a un conjunto y nunca pertenece del todo al mismo. Una definición podría ser la siguiente:

Un subconjunto borroso A de un universo $X=\{x\}$ es un conjunto de pares ordenados $A = \{(x, 1_A(x)) \mid x \in X\}$ donde $1_{A:X} : [0,1]$ es la función de pertenencia asociada a A .

El grado de certeza no tiene un sentido probabilístico, como en otras lógicas, sino más bien representa un grado de compatibilidad o posibilidad de que un predicado sea cierto. Siguiendo con el ejemplo de estaturas, la función de pertenencia del conjunto **muy bajo** (MB) se podría tomar como:

$$\begin{aligned} \text{si } x < 1,50 & \quad 1_{MB}(x) = 1 \\ \text{si } 1,50 \leq x < 1,60 & \quad 1_{MB}(x) = 1 - 10(x - 1,5) \\ \text{si } 1,60 \leq x & \quad 1_{MB}(x) = 0 \end{aligned}$$

En la figura 5.1 se han definido una gama completa de conjuntos borrosos de estaturas en base a los siguientes términos: **muy bajo**, **bajo**, **normal**, **alto** y **muy alto**.

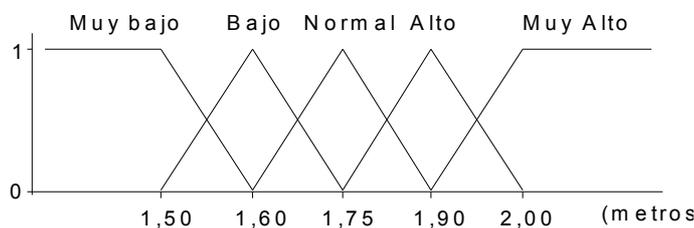


Fig. 5.1 Funciones de pertenencia.

Las operaciones básicas que se definen normalmente con conjuntos borrosos son las mismas que con los conjuntos clásicos.

igualdad	$A = B \iff \mu_A(x) = \mu_B(x)$
inclusión	$A \subseteq B \iff \mu_A(x) \leq \mu_B(x)$
unión	$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$
intersección	$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$
complemento	$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$

Observando estas operaciones se concluye que los conjuntos clásicos se pueden considerar un caso particular de los conjuntos borrosos, en los que la función de pertenencia toma exclusivamente valores 0 ó 1. En particular, el conjunto vacío, \emptyset , tendrá función de pertenencia constante e igual a cero, mientras el conjunto completo, el universo X , también la tendrá constante, pero igual a uno.

Debemos advertir, en relación a las lógicas clásicas, que la operación de complemento, por la propia definición de conjunto borroso, no da en general un conjunto disjuncto ni complementario en el sentido clásico:

$$\begin{aligned} A \cap \bar{A} &= \emptyset \\ A \cup \bar{A} &= X \end{aligned}$$

Un concepto a destacar es el de *relación borrosa* entre conjuntos borrosos. Para lo que se va a definir en primer lugar el producto cartesiano: dados dos subconjuntos borrosos A y B de universos X e Y respectivamente, se define producto cartesiano como

$$A \times B = \{(x, y), \min(\mu_A(x), \mu_B(y)) \mid \forall x \in X \quad \forall y \in Y\}$$

Entonces se llama *relación borrosa* a cualquier subconjunto del producto cartesiano.

Por último faltaría definir la composición de relaciones. Dados tres subconjuntos A , B y C de universos X , Y y Z entre los que se tienen definidas las siguientes relaciones:

$$\begin{aligned} A R_1 B &= \{(x, y), \mu_{R_1}(x, y)\} \\ B R_2 C &= \{(y, z), \mu_{R_2}(y, z)\} \equiv \end{aligned}$$

Se define la relación compuesta de R1 y R2 como

$$A(R_1 R_2)C = \{(x, y), \max_{\forall y \in Y} \{\min(\mu_{R_1}(x, y), \mu_{R_2}(y, z))\}\}$$

Semántica borrosa

En la lógica borrosa los *valores semánticos* serán subconjuntos borrosos del intervalo real $[0,1]$, siendo necesario definir para cada predicado los correspondientes subconjuntos. Llegados a este punto conviene precisar algunos conceptos que se emplean con asiduidad.

1 Se denomina *universo de discurso* al conjunto de posibles valores particulares que pueden tomar las variables que intervienen en el predicado. En la figura 5.1 se representa el universo de discurso para la variable estatura, que se ha elegido entre 1,50 y 2,00 m.

1 Se llaman *etiquetas lingüísticas* a los valores semánticos correspondientes a un predicado. En el ejemplo se han tomado cinco etiquetas. Estas etiquetas se corresponden con el lenguaje natural, y habitualmente se toman entre 5 y 9, ya que menos no definen bien el problema, y más se salen fuera de los límites que sería capaz de discernir una persona.

1 Las *funciones de pertenencia* son las que aparecen en la figura 5.1. Cada término lingüístico corresponde a un subconjunto borroso que lleva asociada una función de pertenencia. Esta representa el grado de asociación de un valor numérico (una estatura) con ese término. Por ejemplo, "1.80 es ALTO" toma el valor 0.33 para la función de pertenencia de **alto**.

Dado que la interpretación de un predicado (asignación de un valor semántico) es un conjunto borroso, las operaciones semánticas para interpretar una sentencia serán las correspondientes a los conjuntos borrosos:

$$\begin{aligned} I(\neg A) &= \overline{I(A)} \\ I(A \wedge B) &= I(A) \cap I(B) \\ I(A \vee B) &= I(A) \cup I(B) \end{aligned}$$

Como conclusión de estas operaciones se puede decir que, dado que la operación de complemento no da conjuntos complementarios en el sentido clásico, tautologías de la lógica de proposiciones, como $2A \ 2 \ A$, dejan de serlo en la borrosa, e incluso resulta difícil hablar de tautologías. La conclusión, por tanto, es que la lógica borrosa no será ni completa ni consistente.

Si no hay unanimidad de criterio en cuanto a las operaciones borrosas anteriores, la disparidad en la interpretación del condicional se hace francamente notoria. Como interpretación más sencilla de $A \rightarrow B$ se puede tomar como la de una relación borrosa entre las interpretaciones de A y B. En particular como el propio producto cartesiano

$$I(A \rightarrow B) = I(A) \times I(B)$$

Una segunda interpretación, quizás no demasiado afortunada, es considerar equivalente $A \rightarrow B$ y $\neg A \vee B$, por lo que sería

$$I(A \rightarrow B) = \overline{I(A)} \cup I(B)$$

La tercera versión, posiblemente la más aceptada, es interpretar primero el caso "si A entonces B, si no A entonces C" como

$$I[(A \rightarrow B) \vee (\overline{A} \rightarrow C)] = (I(A) \times I(B)) \cup (\overline{I(A)} \cup I(C))$$

y considerar entonces A 1 B como un caso particular del anterior, donde C coincide con su universo

$$I(A \rightarrow B) = \{(x, y), \max[\min(\mu_A(x), \mu_B(y)), 1 - \mu_A(x)] \quad \forall x \in X \quad \forall y \in Y\}$$

CONTROL BORROSO DIRECTO

CONTROLADOR PID DIGITAL

Resulta interesante, antes de estudiar el controlador borroso, repasar algunas ideas básicas empleadas en el control clásico. Un PID tradicional proporciona una acción que es suma de tres efectos, uno Proporcional a la señal de error, otro proporcional a su Integral y otro a su Derivada. La acción proporcional produce una señal de control, mayor cuanto más grande es el error, que tiende a contrarrestarlo, estableciendo la ganancia. La integral elimina el error que, en la mayoría de los sistemas, se produce en régimen estacionario cuando hay un cambio en el punto de consigna, con el inconveniente de que aumenta la inestabilidad del proceso. La acción derivativa permite acelerar la respuesta del sistema, es decir, mejora el transitorio, pero su problema es que amplifica las señales de ruido. La expresión continua de la actuación de control es:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}$$

donde las tres constantes k_p , k_i y k_d se denominan ganancia proporcional, integral y derivativa, respectivamente. También suele representarse de la forma:

$$u(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right]$$

en función de la ganancia global K , del tiempo integral T_i y del tiempo derivativo T_d .

La versión digital correspondiente será el resultado de discretizar esta ecuación, considerando sólo los valores de las funciones en instantes de muestreo múltiplos de un cierto periodo T . La derivada se sustituye por una diferencia de primer orden y la integral por una suma. Se obtiene así una aproximación válida sólo si dicho periodo de muestreo es suficientemente pequeño.

Dependiendo del método numérico que se utilice para aproximar la integral se tendrán expresiones con los parámetros ligeramente diferentes, pero con la misma estructura. En el caso concreto de aproximación trapezoidal se tiene:

$$u(nT) \equiv u(n) = k_p e(n) + k_i T \left[\frac{e(0) + e(n)}{2} + \sum_{m=1}^{n-1} e(m) \right] + k_d \frac{e(n) - e(n-1)}{T}$$

Para implementar en la computadora esta ley, o cualquier otra, es preferible obtener su forma recursiva, ya que, además de ser más sencilla, evita problemas de saturación. Restándole por tanto a la expresión de la señal de control en un instante la correspondiente al instante anterior, se obtiene:

$$u(n) = u(n-1) + k_p [e(n) - e(n-1)] + k_i T \frac{e(n) - e(n-1)}{2} + k_d \frac{e(n) - 2e(n-1) + e(n-2)}{T}$$

y reordenándolo para que quede de forma más compacta:

$$u(n) = u(n-1) + q_0 e(n) + q_1 e(n-1) + q_2 e(n-2)$$

siendo los coeficientes q_0 , q_1 y q_2 funciones de los parámetros del PID continuo k_p , k_i y k_d , y del tiempo de muestreo T (recordar lo comentado para la figura 1.2), con una forma concreta que depende del método utilizado para aproximar la integral y la derivada. Para nuestro caso:

$$\begin{aligned} q_0 &= k_p + \frac{k_i T}{2} + \frac{k_d}{T} \\ q_1 &= -k_p + \frac{k_i T}{2} - 2 \frac{k_d}{T} \\ q_2 &= \frac{k_d}{T} \end{aligned}$$

Como puede apreciarse, el algoritmo consiste en una simple ecuación matemática *lineal*. A pesar de su sencillez este controlador tiene en general un buen comportamiento, razón que justifica su gran utilización. El principal problema que presenta en cualquiera de sus formas es la obtención de sus parámetros. Aunque existen varios métodos y reglas para estimarlos [28], los resultados que proporcionan no son totalmente satisfactorios y suele ser necesario realizar sucesivas pruebas y ajustes hasta llegar a los valores óptimos.

Nosotros hemos querido comparar la respuesta de un PID con el controlador borroso que más adelante describiremos. Para ello ha sido necesario concebir sobre la simulación establecida de antemano todos los elementos necesarios para su implantación. Primero hemos creado una clase de objetos, llamada CONTROLADOR, con unos atributos, que serán genéricos a todos ellos:

- 1 ENTRADA-CONTROL (entrada en intensidad al controlador, procedente del sensor de nivel), SEÑAL-CONTROL (salida de control del mismo, también en intensidad), SET-POINT (valor de consigna en mA), HREF (valor de consigna en altura) y DAEMON (se

emplea para lanzar la regla que contiene el algoritmo de control, de acuerdo con el tiempo de muestreo MT). Todas estos son del tipo PAR-F-DO-NOT-FORW.

1 REFERENCIA. Es un parámetro del tipo PAR-F-DO-FORW, que representa el porcentaje de altura deseado.

1 MT. Es el tiempo de muestreo del controlador. Es del tipo CTE-CONTROLADOR.

Las reglas que atañen a todo CONTROLADOR hacen alusión a estos parámetros, y son las siguientes:

1 C-R1.

for any controlador C if the referencia R of C > 100 or R < 0 then
conclude that $R = \min(100, \max(0, R))$

Ésta limita el valor de la referencia de cualquier controlador a un valor entre 0 y 100. Tiene mayor prioridad, como se ha visto en ocasiones precedentes.

1 C-R2.

whenever the referencia R of any controlador C receives a value then
conclude that the set-point of C = $0.16 * R + 4$
and conclude that the href of C = $R * 63,5 / 100,0 + 110,0$

Fija el valor de consigna del controlador y la altura correspondiente.

1 C-R3.

whenever the salida-sensor SS of the sensor nivel SN connected at the output out-senal of D-1 recives a value then
conclude that the entrada-control of the controlador connected at the output of SN = SS

Asigna el valor de ENTRADA-CONTROL según el valor del sensor de nivel conectado a la entrada del controlador.

1 C-R4.

whenever the mt MM of the controlador connected at the output of SN-1 recieves a value then
change de text of C-FS0 to "the daemon of any controlador = 0, with time increment [MM] seconds"

Esta regla cambia el intervalo de muestreo del controlador, según la fórmula de simulación siguiente.

1 C-FS0.

the daemon of any controlador = 0 , with time increment 1 second

Se utiliza para asignar un nuevo valor (es siempre el mismo) al atributo DAEMON cada MT segundos, que lanzará, vía encadenamiento hacia adelante, las reglas que activan el algoritmo de control (ver reglas PID-R2, en la página siguiente, y FF-R1, en la página 106).

Además hemos creado otra clase de objetos, llamada PID, que deriva de la anterior y que posee los siguientes atributos:

1 ERROR, ERROR-1, ERROR-2, Q0, Q1 y Q2. Son del tipo PAR-F-DO-NOR-FORW y se emplean dentro del algoritmo de control.

1 KP, KI y KD. Pertenecen al tipo CTE-CONTROLADOR. Representan los parámetros del controlador.

Las reglas que operan sobre los controladores PID son las siguientes:

1 PID-R1.

whenever any cte-controlador receives a value then
 conclude that the q0 of every pid P = the kp of P + 0.5*the mt of P*the ki of P + the kd of P/the mt of P
 and conclude that the q1 of every PID P2 = 0.5*the mt of P2*the ki of P2 - the kp of P2 - 2.0*(the kd of P2/the mt of P2)
 and conclude that the q2 of every PID P3 = the kd of P3/the mt of P3

Esta regla asigna el valor a los coeficientes del controlador, como estudiamos antes.

1 PID-R2.

whenever the daemon of any pid P receives a value then in order
 conclude that the error-2 of P = the error-1 of P
 and conclude that the error-1 of P = the error of P
 and conclude that the error of P = the set-point of P - the entrada-control of P
 and conclude that the señal-control SC of P = min(16.4,max(7.9, SC + the q0 of P*the error of P + the q1 of P*the error-1 of P + the q2 of P*the error-2 of P))

Se ejecuta cuando DAEMON recibe un nuevo valor, es decir, cada tiempo de muestreo. Fija los valores de los distintos errores, y es el algoritmo de control del controlador. Comentar este hecho: la señal de control se establece en una regla. Señalar además que la salida de ésta se ha limitado entre 16,4 y 7,9 en vez de hacerlo entre 20 y 4 (mA), debido a las razones comentadas cuando se estudio el transductor intensidad-presión (fig. 2.7).

Una vez creados todos los elementos necesarios para incorporar un PID al sistema, y ajustados los parámetros del mismo (después de varios intentos, como suele ser habitual [28]) y tiempo de muestreo, a los siguientes valores:

1 Tiempo de integración para la simulación del proceso: 1 segundo (se obtienen prácticamente los mismo resultados que empleando tiempos menores).

1 Tiempo de muestreo del controlador: 2 segundo (muchos autores indican que para un proceso de nivel el tiempo de muestreo suele tomarse entre 5 y 10 segundos, sin embargo, nosotros hemos decidido emplear éste por las mejoras que supone tener activado durante más tiempo el controlador).

1 kd = 23.0; ki = 1.0; kp = 5.0

se presentará la respuesta del sistema con un controlador de este tipo. Se han tomado tres alturas características (25, 50 y 75 %) y tres caudales de entrada frecuentes (50, 82 y 113 cm³/s), que nos permiten abarcar casi todas las posibilidades. Para el caso en que el caudal de entrada es de 113 cm³/s, no se ha tomado la altura de referencia 25 %, pues esta altura es en la que aproximadamente se establece el equilibrio dinámico para válvula totalmente abierta.

Las pruebas se han realizado sin fluctuaciones en la red de suministro, es decir, el caudal de entrada a la instalación, para cada caso, ha permanecido en un valor fijo, lo cual permite ver claramente la respuesta del controlador sin encubrir posibles resultados derivados de la existencia de perturbaciones. Sin embargo, no podemos eliminar el comportamiento no lineal de la válvula de control debido a la histéresis en la carrera, pues es una característica inherente de la misma, que no podemos evitar.

El estudio se ha realizado en situaciones en que el nivel de líquido ha sido superior e inferior al valor deseado, pues la no linealidad del sistema y la característica de apertura rápida de la válvula, dan indicios de que las respuestas deben ser apreciablemente distintas. De hecho, los parámetros anteriores dependen mucho de la situación elegida; así, cuando el valor del nivel del depósito es menor que el de referencia, se realiza un buen ajuste del controlador con los siguientes:

$$1 \text{ kd} = 27.0; \text{ ki} = 0.6; \text{ kp} = 6.0$$

y cuando la situación es la contraria, con:

$$1 \text{ kd} = 19.0; \text{ ki} = 1.35; \text{ kp} = 4.0$$

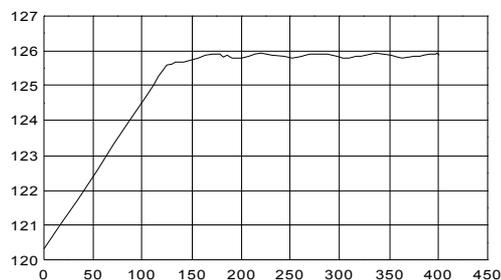
Hemos preferido no privilegiar una sobre la otra, de modo que se ha tomado aproximadamente la media de cada parámetro en cada situación. A continuación se presenta la respuesta del sistema, tomando la *altura* de salida del depósito, y la *carrera* de la válvula. Parece claro pensar que ésta es un elemento importante, ya que el nivel puede mantenerse cercano al de referencia, pero la carrera puede sufrir variaciones bruscas que pueden llevar a la válvula a un estado inestable entre sus dos posiciones extremas, que en ningún momento sería aconsejable para la vida útil de la misma, y sobre todo porque el control en esta situación sería muy pobre.

En las páginas siguientes se muestran las gráficas que hemos obtenido, en las que se representa la altura y la carrera frente al tiempo. Todas ellas tienen su eje de abscisas en segundos; para la altura, el de ordenadas viene dado en cm; y para la carrera, en mm.

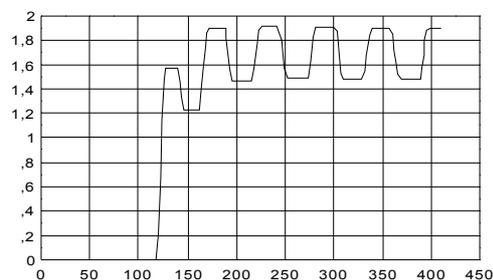
1) Caudal de entrada 50 cm³/s.

1 Altura de referencia 25 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

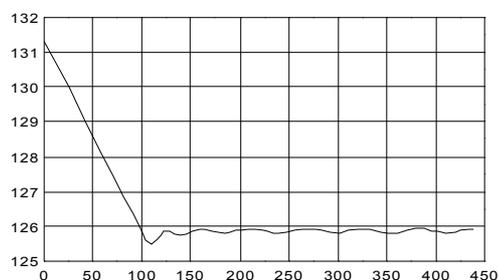


Altura

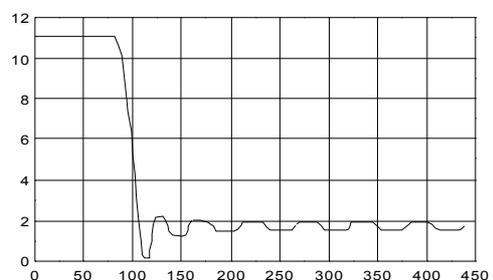


Carrera

b) Situación contraria.



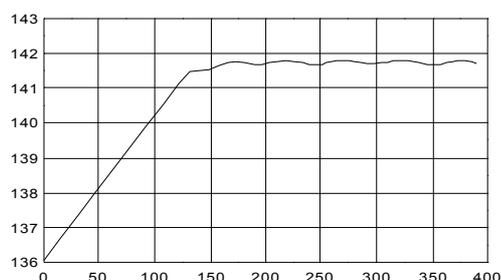
Altura



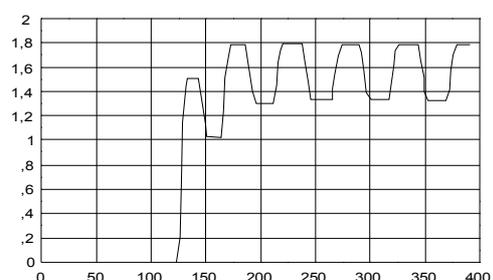
Carrera

1 Altura de referencia 50 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

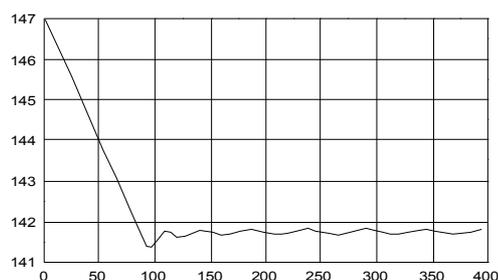


Altura

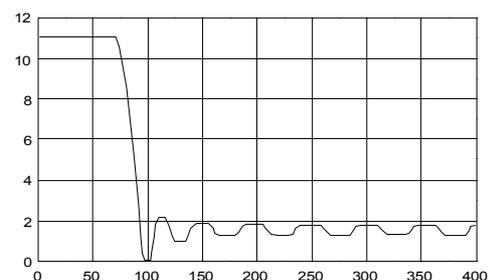


Carrera

b) Situación contraria.



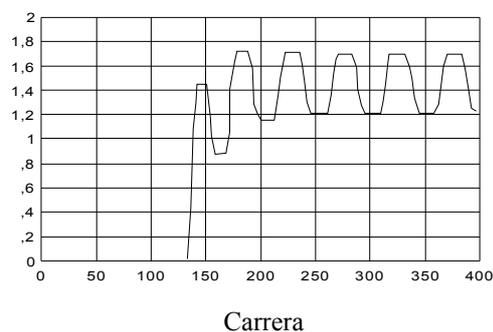
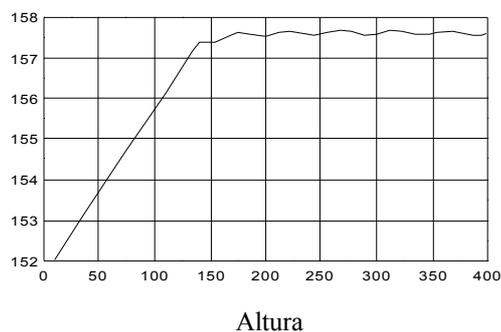
Altura



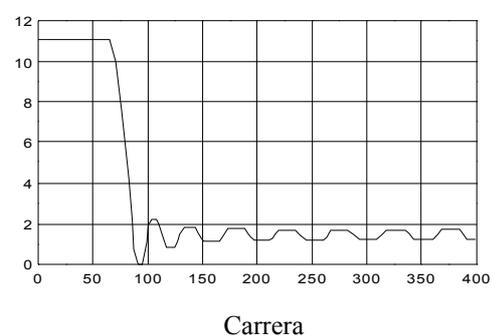
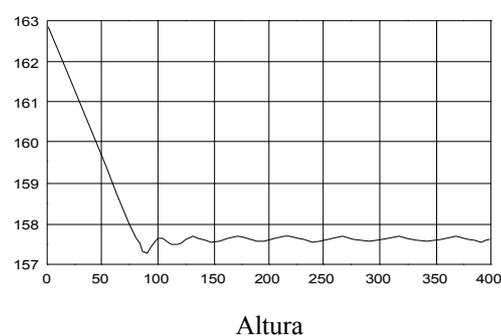
Carrera

1 Altura de referencia 75 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:



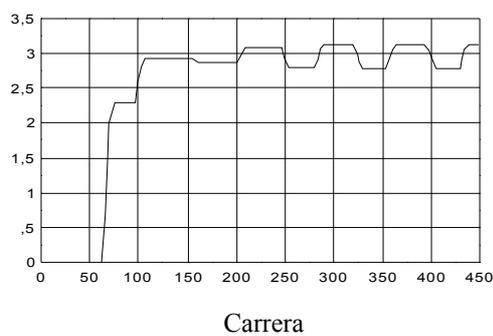
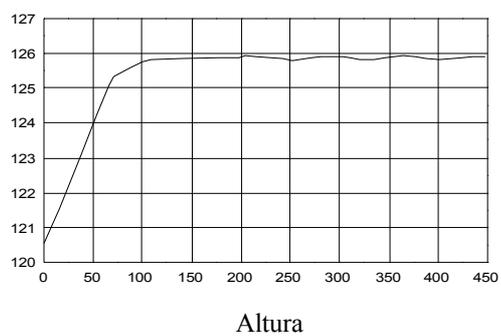
b) Situación contraria:



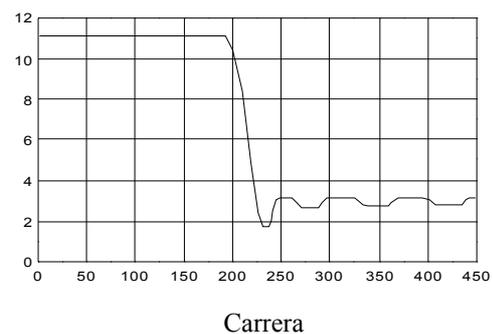
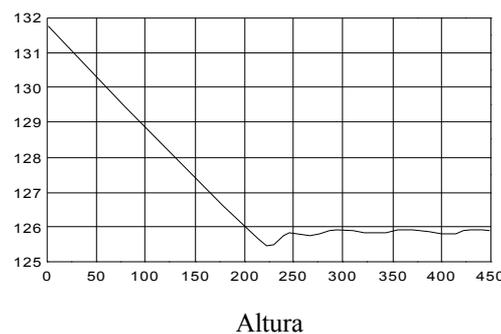
2) Caudal de entrada 82 cm³/s.

1 Altura de referencia 25 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

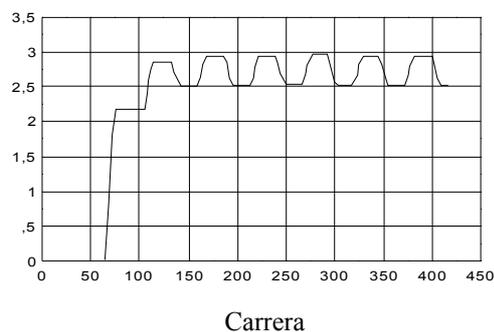
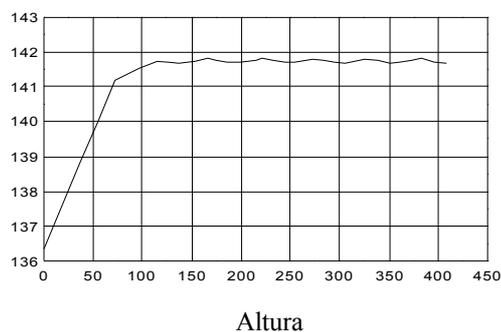


b) Situación contraria:

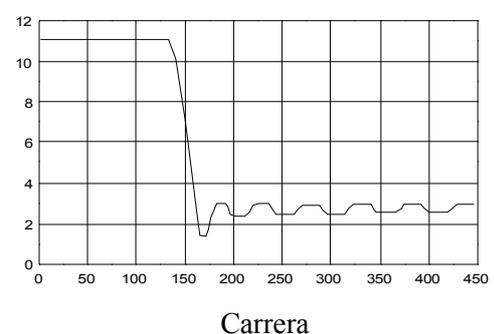
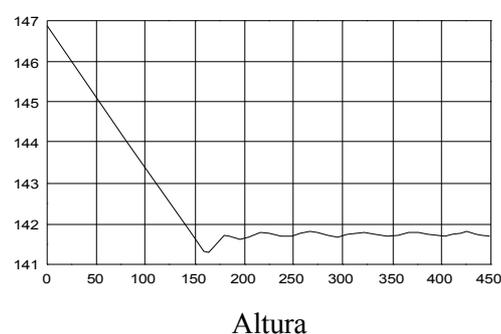


1 Altura de referencia 50 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

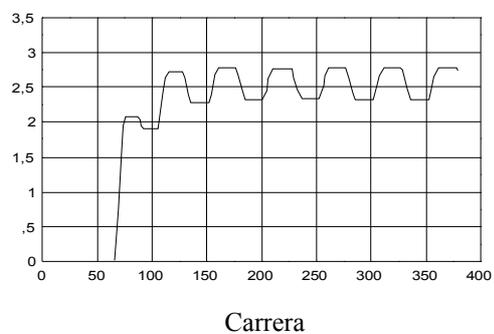
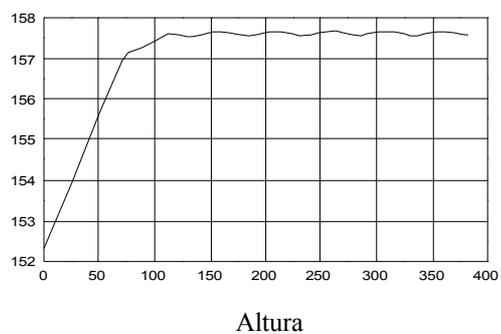


b) Situación contraria:

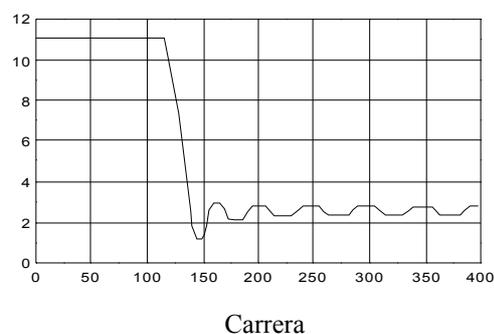
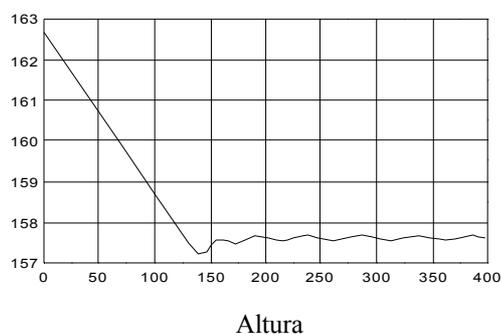


1 Altura de referencia 75 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:



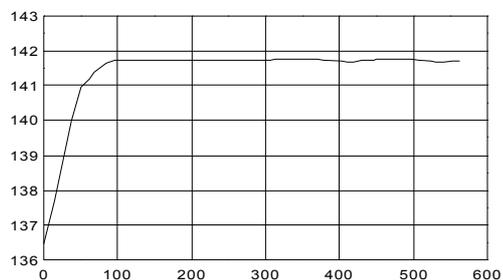
b) Situación contraria:



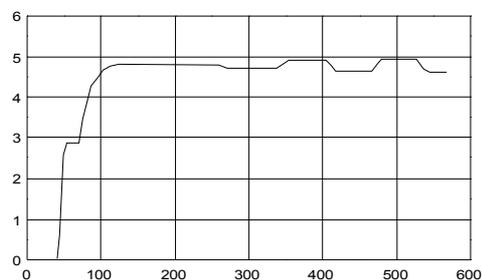
3) Caudal de entrada 113 cm³/s.

1 Altura de referencia 50 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

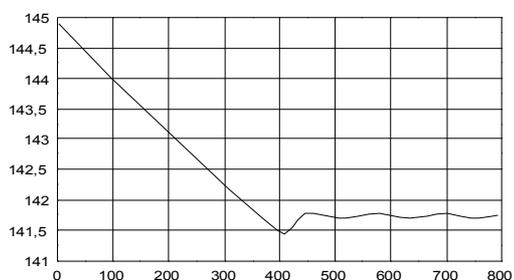


Altura

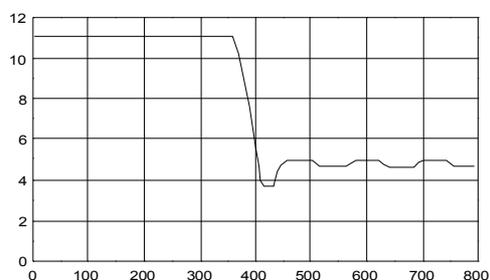


Carrera

b) Situación contraria:



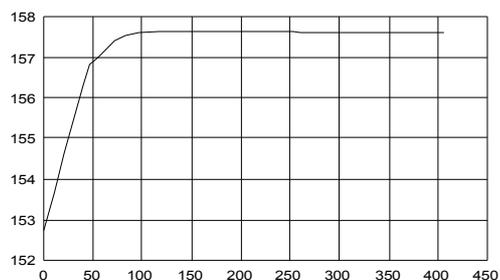
Altura



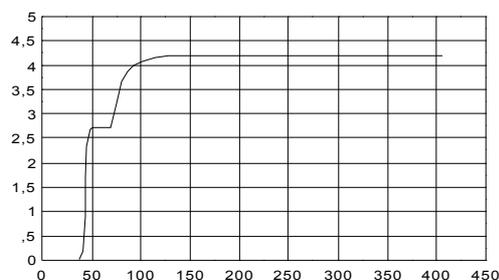
Carrera

1 Altura de referencia 75 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

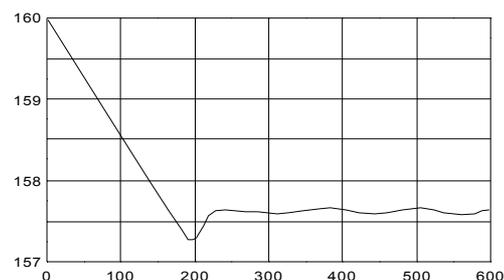


Altura

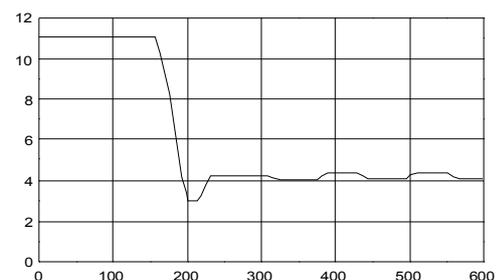


Carrera

b) Situación contraria:



Altura



Carrera

Del análisis de todas estas gráficas se obtiene un resultado esperado: aunque el nivel se mantiene en torno al valor de la referencia (la desviación máxima es muy pequeña, de unos pocos milímetros), la carrera de la válvula sufre las consecuencias de su histéresis. En el proceso real también se aprecia este hecho, e igual que en la simulación, las variaciones de la carrera no superan los dos milímetros en la mayoría de los casos. Las válvulas de control están diseñadas para soportar adecuadamente esta variación continuada de la carrera, siempre que se realice un mantenimiento periódico adecuado de la misma, aunque una variación menor siempre puede alargar su vida útil.

En el apartado siguiente nos proponemos realizar un controlador a la medida del proceso, de modo que el control sea más efectivo, y posibilitando la reducción en las variaciones en la carrera de la válvula debida a su histéresis. Puesto que la histéresis se produce cuando se pretende retornar a una situación anterior (es decir, siempre que se intenta corregir la posición), debemos procurar que el sistema no se comporte como uno típicamente subamortiguado, sino que lo haga como uno sobreamortiguado, en el que la posición se alcance paulatinamente sin cambiar el sentido de la acción de control, y esto para cada situación específica del proceso. Aunque no es sencillo de elaborar, y prácticamente imposible con un controlador lineal como el PID, con el controlador borroso que se describe seguidamente pueden establecerse relaciones heurísticas entre distintas situaciones del proceso, que pueden proporcionar un control más refinado sobre el proceso que estamos describiendo.

CONTROLADOR BORROSO

Como diferencia respecto al tipo de control que vamos a estudiar, resaltar que emplearemos sólo dos de las componentes P+I+D anteriores, y falta de linealidad.

La idea básica del controlador fuzzy aparece reflejada en la figura 5.2. En ella puede apreciarse como, a partir de la consigna y de la variable procedente del sensor, se calcula el error (E) y el cambio en el error (CE), ponderándose mediante las constantes **K1** y **K2**. Tomando estas variables de partida, el controlador produce como salida el cambio en la acción de control a aplicar (CU), o bien una acción de control absoluta (U).

Internamente, el controlador borroso se compone de un conjunto de reglas lingüísticas, que tienen como condiciones los valores posibles de las variables de entrada, y que concluyen una acción de control, o su variación, en términos también lingüísticos (que luego será transformada en un valor determinista). Las reglas, de carácter heurístico, se obtienen de la experiencia sobre el proceso, de los conocimientos prácticos de aplicación en general, y de los resultados registrados sobre el proceso real o sobre prototipos del mismo. La facilidad de introducir nuevas variables y nuevas reglas permite realizar controladores muy complejos, pero simples de implantar.

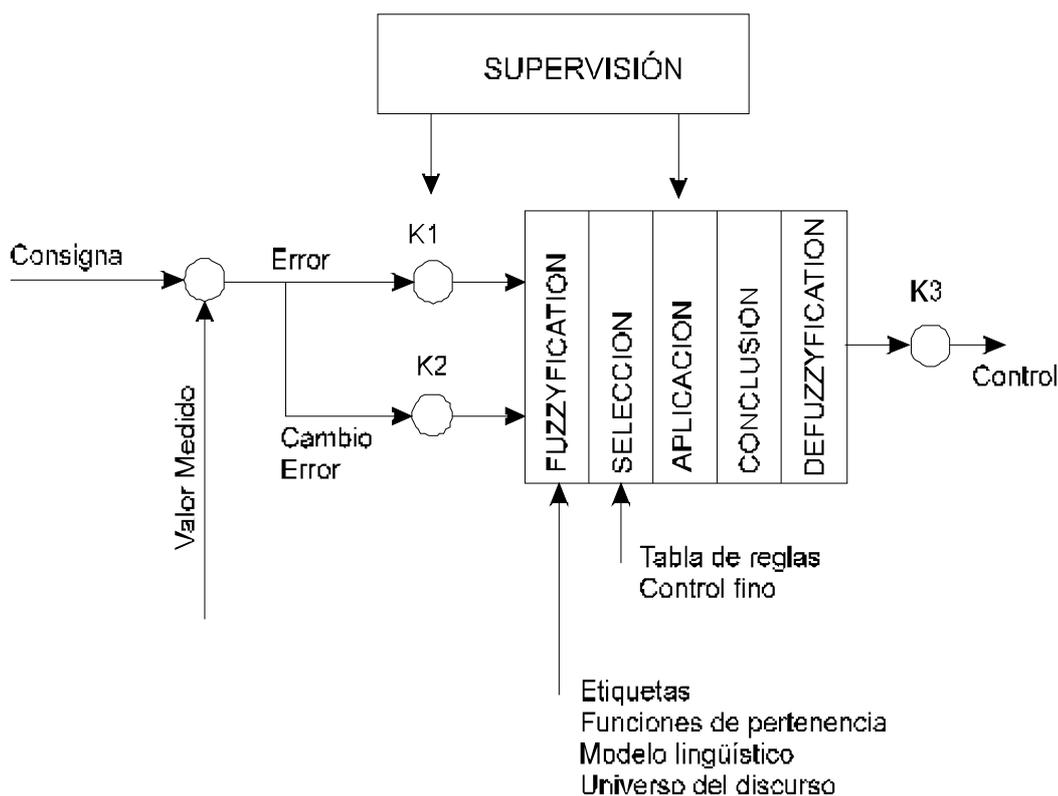


Fig. 5.2 Controlador fuzzy

Mencionar también que opcionalmente puede existir un módulo de supervisión que ajuste los parámetros del regulador on-line.

El control borroso consta de las siguientes etapas:

- 1 Fuzzyfication (borrosificación)
- 1 Planteamiento y selección de reglas
- 1 Aplicación de reglas y conclusión borrosa
- 1 Defuzzyfication y conclusión numérica.

Fuzzyfication (borrosificación)

La primera acción que hay que realizar es obtener, a partir de los valores deterministas de E y CE, sus equivalentes valores borrosos. Para ello es preciso tener definido el universo del discurso y las etiquetas lingüísticas que se van a emplear, así como la función de pertenencia asociada a cada una de ellas, como se muestra en la figura 5.3.

El eje de abscisas representa el *universo de discurso*, es decir, el rango de variación de la variable de entrada (E y CE), que en nuestro caso se ha elegido entre 13 para ambas dentro del controlador (ver apéndice I), si bien, las constantes K1 y K2 van a ponderar el valor real de este

intervalo. Además debemos señalar que, debido a las características del proceso, se ha tomado distinto valor de estas constantes dependiendo tanto del signo de E como del de CE.

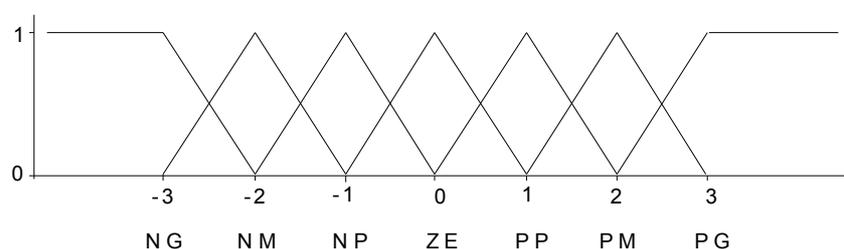


Fig. 5.3 Funciones de pertenencia.

Como *etiquetas lingüísticas* se han tomado siete (para las dos, E y CE):

- PG:** positivo grande, centrado en +3
- PM:** positivo mediano, centrado en +2
- PP:** positivo pequeño, centrado en +1
- ZE:** cero, centrado en 0
- NP:** negativo pequeño, centrado en -1
- NM:** negativo mediano, centrado en -2
- NG:** negativo grande, centrado en -3

En cuanto a la *función de pertenencia* de un término lingüístico, ésta representa el grado de asociación del valor numérico de la señal de error, $e(t)$, o de la señal cambio en el error, $c(t)$, con ese término, como ya se dijo anteriormente. Por ejemplo, 1 sería PP con un grado de certeza de 1, y tendría un grado de certeza 0 para las restantes etiquetas, como puede verse en la figura 5.3, ya que no corta a ninguna de sus funciones de pertenencia; mientras que -0.4 sería ZE con un grado de certeza de 0,4 y sería NP con un grado de certeza de 0,6. Las funciones de pertenencia básicamente pueden ser de cinco tipos, clasificándose por su forma:

- *Trapezoidales*. Pueden derivar en rectangulares o en triangulares.
 - *Rectangulares*. No tienen significado físico, pues una variable borrosa no puede pasar bruscamente de valer 1 a valer 0, o a la inversa.
 - *Triangulares*. Son funciones lineales por tramos, pero presentan una discontinuidad en la primera derivada, que hereda la acción de control.
- *Exponenciales*. Muestran un comportamiento muy adecuado y no presentan discontinuidad en la derivada, aunque tiene el inconveniente de su lentitud de cálculo.
- *Polinómicas*. Tienen una forma similar a la de las funciones de densidad normales, siendo más rápidas de calcular que éstas.

En nuestro trabajo hemos empleado funciones de pertenencia triangulares (fig. 5.3), debido fundamentalmente a que G2 proporciona una forma rápida y sencilla de construir estas funciones,

como vimos en el capítulo tercero de esta memoria. Las dos etiquetas extremas tienen funciones de pertenencia unidad desde el centro de cada una de ellas hacia 11, con el fin de truncar aquellos valores de las variables de entrada que no pertenezcan al universo de discurso.

La borrosificación consiste en calcular el grado de pertenencia de las variables de entrada a cada una de las etiquetas lingüísticas mediante las funciones de pertenencia. Éste será un número comprendido entre 0 y 1 para cada etiqueta. Recordemos que en G2 las variables lógicas fuzzy se representan empleando el rango de 41 a 41, por lo que debemos hacer la conversión lineal al intervalo [0,1] al realizar la defuzzyficación.

En los apartados que siguen emplearemos la notación siguiente:

n_e : número de términos lingüísticos asociados a la función de error

n_c : número de términos lingüísticos asociados a la función de cambio en el error

n_u : número de términos lingüísticos asociados a la función de cambio en la acción de control

$1_i(\mathbf{e}(t))$: función de pertenencia asociada a la etiqueta lingüística i de la función de error, $i=[1..n_e]$

$1_j(\mathbf{c}(t))$: función de pertenencia asociada a la etiqueta lingüística j de la función cambio en el error, $j=[1..n_c]$

$1_k(\mathbf{u}(t))$: función de pertenencia asociada a la etiqueta lingüística k de la función cambio en la acción de control $k=[1..n_u]$

Planteamiento y selección de reglas

Tras haber traducido los valores deterministas de las variables de entrada a valores borrosos, se pasa a aplicar aquellas reglas que puedan ser disparadas. Para ello se dispone de una tabla de reglas con dos entradas (E y CE) por cada variable de entrada al regulador (tabla 5.1)

		CE						
		NG	NM	NP	ZE	PP	PM	PG
E	NG	NG	NG	NG	NPP	NPP	NPP	NPP
	NM	NG	NM	NM	NPP	PM	PG	PG
	NP	NM	NM	NP	NPP	PP	PM	PG
	ZE	NG	NG	NP	ZE	PP	PM	PG
	PP	NG	NG	NP	PPP	PP	PG	PG
	PM	NG	NG	NM	PPP	PM	PG	PG
	PG	PPP	PP	PM	PG	PG	PG	PG

Tabla 5.1 Reglas del controlador fuzzy.

La tabla contiene reglas a aplicar por el regulador y son de la forma:

IF E=ZE AND CE=ZE THEN CU=ZE

IF E=PP AND CE=NP THEN CU=NP

Se han empleado 9 etiquetas lingüísticas para el valor difuso de la conclusión, y son las siguientes:

- PG:** positivo grande, centrado en +3,6
- PM:** positivo mediano, centrado en +2,0
- PP:** positivo pequeño, centrado en +1,2
- PPP:** positivo pequeño-pequeño, centrado en 0,5
- ZE:** cero, centrado en 0
- NPP:** negativo pequeño-pequeño, centrado en -0,5
- NP:** negativo pequeño, centrado en -1,1
- NM:** negativo mediano, centrado en -1,8
- NG:** negativo grande, centrado en -3,8

Después de la borrosificación, para cada regla de la tabla tenemos un valor numérico de entrada de E y otro de CE. Podemos construir una tabla adicional, o *matriz de inferencia*, que represente el peso que tendrá cada una de las reglas en la conclusión final. Al ser reglas del tipo:

IF E=... AND CE=... THEN ...

el grado de cumplimiento de la premisa será el menor de cada una de sus condiciones, tomándose este grado como peso de la conclusión. Por lo tanto, cada término de la matriz de inferencia tiene como peso o masa m_{ij} el menor de los respectivos grados de pertenencia de E y CE a las etiquetas correspondientes a esa fila y columna.

$$m_{ij} = \min(1_i(e(t)), 1_j(c(t)))$$

A partir de la matriz de inferencia, resulta posible hacer una selección de las reglas que van a ser aplicadas, eliminando las que tengan un grado de pertenencia demasiado pequeño. Además, de no ser eliminadas, implicaría una acción de control contraria a la deseada. Esta selección de reglas equivale a truncar las funciones de pertenencia. Con funciones polinómicas, trapezoidales o triangulares (nuestro caso) no es preciso.

Aplicación de reglas. Conclusión borrosa

Hasta aquí disponemos de un conjunto de reglas y de un peso para la conclusión final de cada una de ellas. La acción de control que concluye cada regla es un conjunto borroso al que se le ha asignado un peso m_{ij} . La interpretación usual de este conjunto con peso es otro conjunto borroso en el que la función de pertenencia se calcula como producto de la función de pertenencia primitiva por el peso:

$$1'_{ij}(u(t)) = m_{ij} \cdot 1_k(u(t))$$

Por tanto, el resultado final de la aplicación de todas las reglas es una serie de conjuntos borrosos con sus respectivas funciones de pertenencia, como puede apreciarse en la figura 5.4.

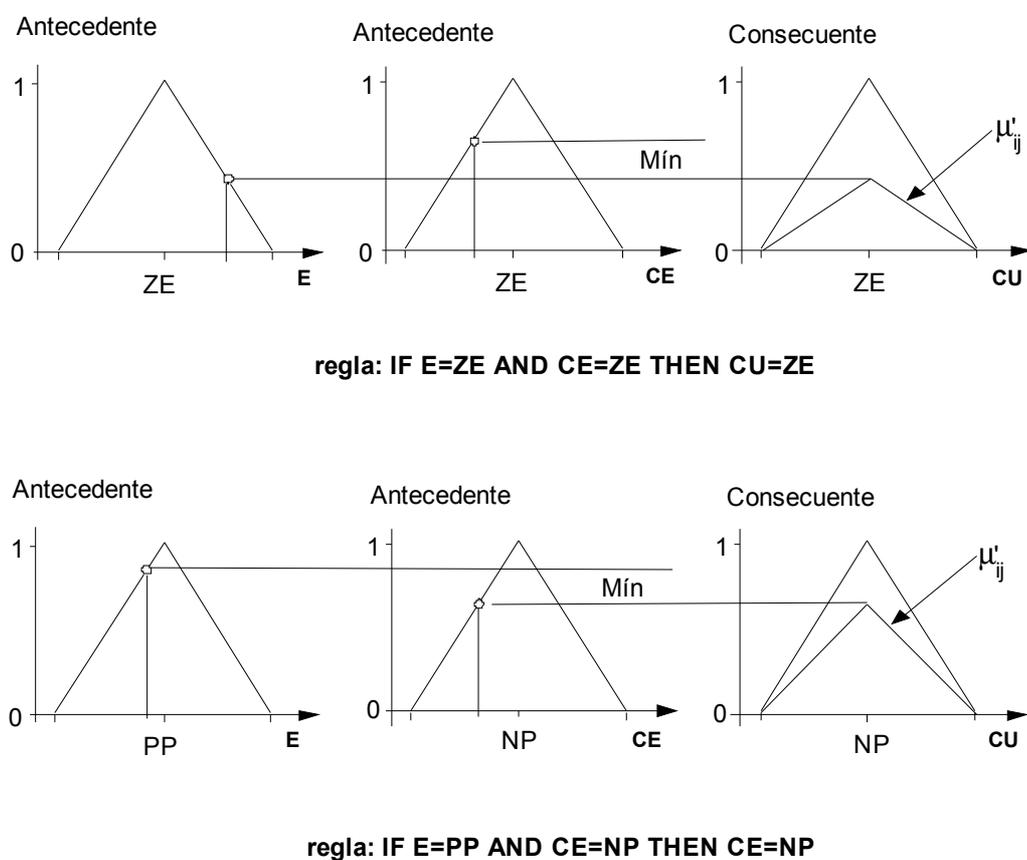


Fig. 5.4 Aplicación de reglas y conclusión borrosa.

Defuzzyficación y conclusión numérica

Finalmente, a partir del conjunto de curvas de salida, se procede al cálculo del valor numérico de la conclusión. Para ello existen esencialmente dos métodos:

- Tomar como conclusión el máximo de la curva suma de todas las anteriores (fig. 5.5 a).
- Calcular el centro de gravedad del área de la curva suma, o punto que deja el mismo área a ambos lados del mismo (fig. 5.5 b).

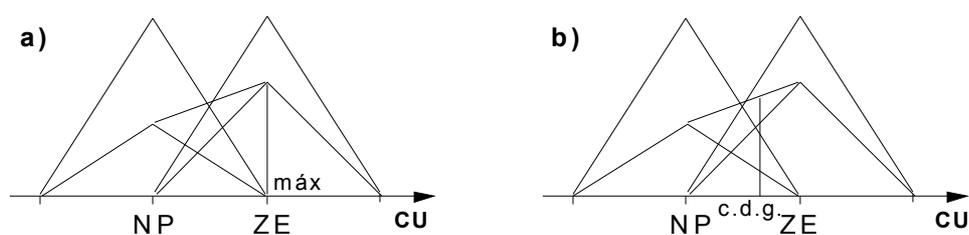


Fig. 5.5 a) Máximo de la curva suma. b) Centro de gravedad.

La última de las opciones es la que tiene más información en cuenta, y es la que nosotros hemos empleado dentro del controlador borroso. El centro de gravedad se calcula como:

$$c.d.g. = \frac{\sum_{i=1}^{n_e} \sum_{j=1}^{n_c} c_{ij} \cdot m_{ij} \cdot a_{ij}}{\sum_{i=1}^{n_e} \sum_{j=1}^{n_c} m_{ij} \cdot a_{ij}}$$

siendo c_{ij} el valor central, m_{ij} el peso y a_{ij} el área de la función de pertenencia asociada a la conclusión de cada regla. Como observación interesante hacer notar que en el resultado final sólo influye el área y no la forma de la curva.

Una vez obtenido el valor numérico, éste puede ser multiplicado por un factor de escala (**K3**), antes de ser enviado como acción de control. Pueden hacerse modificaciones en este valor para proporcionar un ajuste fino, empleando la misma tabla de reglas, o distinta, como se comenta más adelante.

Para terminar con la descripción del controlador borroso debemos hacer consideraciones sobre la construcción de la tabla de reglas, y la configuración de los mismos.

Construcción de la tabla de reglas

La tabla viene determinada por las características del proceso y es configurada por el experto. Además debe cumplir unas normas elementales que garanticen las siguientes propiedades:

- 1 **Completitud.** Las reglas han de cubrir todas las combinaciones posibles de entradas al controlador, con el fin de que no queden huecos o fisuras ante las que no se tomarían acciones coherentes.
- 1 **Consistencia.** Las reglas han de ser consistentes, es decir, no pueden coexistir dos acciones de control para la misma situación, lo que daría lugar a contradicciones.
- 1 **Interacción** de cada una de las reglas con el controlador, es decir, el peso propio que tiene una regla entre las restantes. Se puede tener una idea de su valor analizando en qué medida se ve afectada la acción de control al ir eliminando cada una de las reglas.

Configuración de reguladores fuzzy

La configuración de un regulador fuzzy se lleva a cabo dando valores adecuados a determinados parámetros de los que intervienen en él. Pueden distinguirse dos tipos de parámetros: de configuración y de ajuste:

- 1 Parámetros de configuración:

- (1) Número de variables de entrada y salida.

- (2) Errores y cambios en el error a considerar de entre todos los posibles de las variables de entrada (variables útiles). Importante en control multivariable.
- (3) Número de términos lingüísticos a emplear para cada variable útil anterior.
- (4) Número de términos lingüísticos a emplear para cada variable de salida.
- (5) Dimensión de la tabla de reglas (viene definida por los parámetros anteriores).

1 Parámetros de ajuste:

- (6) Media o centro de cada función de pertenencia de las variables de entrada.
- (7) Amplitud, ancho o desviación típica de cada función de pertenencia (según el tipo de curvas que se empleen) de las variables de entrada.
- (8) Media o centro de cada etiqueta lingüística de las variables de salida o acción de control.
- (9) Amplitud, ancho o desviación típica de cada función de pertenencia (según el tipo de curvas que se empleen) de la variable de salida.

En primer lugar se fijan los parámetros de configuración, y se da un valor inicial adecuado a los parámetros de ajuste. Los parámetros (6) y (7) vienen determinados en gran medida por las características del sistema a controlar (error máximo, transitorio, etc.), por lo que una vez determinados ya no es necesario volver a calibrarlos. El ajuste, por tanto, se basará esencialmente en el establecimiento de (8) y (9), es decir, acomodando las funciones de pertenencia de las variables de salida. Seguidamente hacemos unas reflexiones sobre la elección de los parámetros (6) y (7):

- (6): comprimiendo las medias se reduce el universo de discurso, es decir, tiene en cuenta valores más pequeños para la entrada del regulador. Si no se varía ningún otro parámetro, la acción de control es más rápida y brusca. Esto puede utilizarse en el mismo regulador para aumentar la resolución (control fino), si se combina con una disminución de escala en las etiquetas de salida. Las expansiones/compresiones pueden hacerse aumentando algo más los c.d.g. asociados a las etiquetas extremas, es decir, no tiene porqué hacerse de forma proporcional (según se comprueba empíricamente).
- (7): Al aumentar la amplitud, las funciones de pertenencia son más planas, por lo que cualquier valor numérico aumentará su grado de pertenencia a todas las etiquetas lingüísticas. El resultado es una acción de control más suave, ya que todas las reglas toman más peso (incluso las que son contrarias a la acción necesaria). Si la amplitud aumenta demasiado, la acción de control se va haciendo más lenta, hasta anularse en el límite, lo que sería equivalente a aplicar todas las reglas con grado de certeza igual a uno. Al disminuir la amplitud, las acciones de control son más precisas y rápidas, pues el regulador resulta más

selectivo con las reglas que aplica. Pero si se disminuye demasiado, las acciones de control se vuelven irregulares, ya que quedan espacios muertos entre las funciones de pertenencia de las etiquetas lingüísticas.

La propia tabla de reglas es susceptible de ser modificada, si las conclusiones de sus reglas se expresan en forma de valores numéricos y no mediante etiquetas lingüísticas. De esta forma, la conclusión sería directamente el centro de gravedad de cada función de pertenencia asociada a la salida del controlador. Al modificar las reglas, debe tomarse la precaución de seguir cumpliendo las condiciones de consistencia y estabilidad de partida.

Control fino

Con el fin de disminuir el error en régimen permanente es factible utilizar una tabla de control fino, similar a las descritas (o incluso con otras reglas), con mayor escala de resoluciones y acciones de control proporcionales a esa escala. De esta forma, un error en régimen permanente que pudiera parecer pequeño con la tabla inicial, con esta última tendría mayor resolución, y el regulador seguiría actuando, en vez de detenerse por haber alcanzado ya la zona muerta.

También es perfectamente posible disponer de una sola tabla para ambos casos y variar únicamente las escalas K_1 , K_2 y K_3 .

1 **K_1 y K_2** : son las constante que multiplican al error y variación del error antes de pasarlos a sus equivalente borrosos. Variarlas es análogo a variar simultáneamente las medias y las amplitudes de las etiquetas lingüísticas, por lo que su influencia se limita a la ya comentada para el caso (6) del apartado anterior.

1 **K_3** : es la constante que multiplica el valor determinista de la conclusión. Su variación tiene un efecto directo sobre la acción de control. A veces es preciso modificarla de acuerdo con las otras dos para obtener una acción de control más precisa.

Señalamos aquí que nosotros modificamos solamente el valor de las escalas K_1 y K_2 dependiendo del signo del error y del cambio en el error, respectivamente, pues las características de la válvula de control (recordemos que es de apertura rápida) obligan a hacer distintas consideraciones, implementadas de forma no lineal en el controlador borroso, para obtener un comportamiento adecuado del proceso.

Desarrollo sobre G2

Al igual que hicimos con el PID, debemos implementar el controlador borroso sobre G2. Para ello se ha definido una clase llamada FUZZY, que deriva de CONTROLADOR, y que tiene los siguientes atributos:

1 ERROR, ERROR-1 y CAMBIO-ERROR. Son parámetros PAR-F-DO-NOT-FORW, que viene fijados por la siguiente regla:

1 FF-R1:

whenever the daemon of any fuzzy FF receives a value then in order
 conclude that the error-1 of FF = the error of FF and
 conclude that the error of FF = the set-point of FF-the entrada-control of FF
 and conclude that the cambio-error of FF = the error of FF-the error-1 of FF
 and start proc-fuzzy(the error of FF, the cambio-error of FF, the señal-
 control of FF)

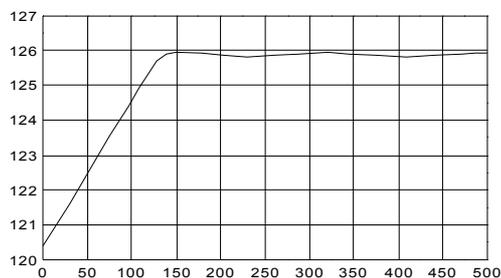
Esta regla, además de fijar el valor de los atributos del controlador fuzzy, lanza el procedimiento que calcula la señal de control, y que se muestra en el apéndice I. Una diferencia respecto del cálculo de la acción de control con el PID, es que en aquel caso se realizaba dentro de una regla, mientras que aquí, debido a su mayor complejidad se tiene que realizar dentro de un procedimiento.

Hemos realizado las pruebas que hacíamos en el caso del controlador PID, y las mismas consideraciones a la hora de desarrollarlas, cuyos resultados se muestran en las gráficas de las páginas siguientes:

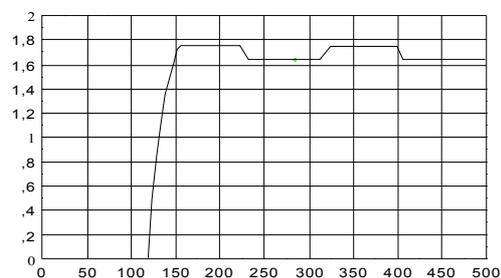
1) Caudal de entrada 50 cm³/s.

1 Altura de referencia 25 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

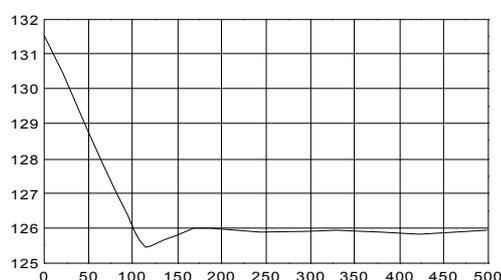


Altura

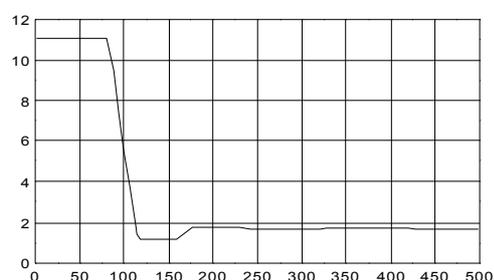


Carrera

b) Situación contraria.



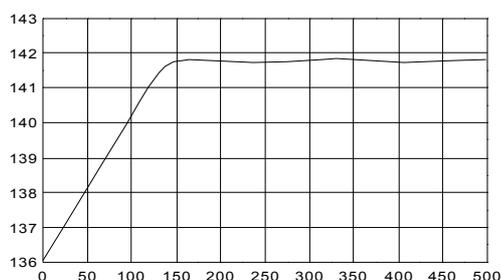
Altura



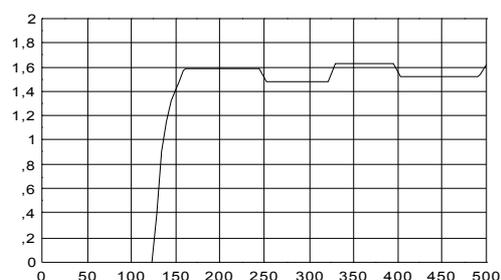
Carrera

1 Altura de referencia 50 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

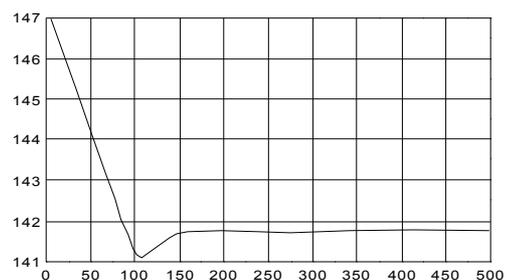


Altura

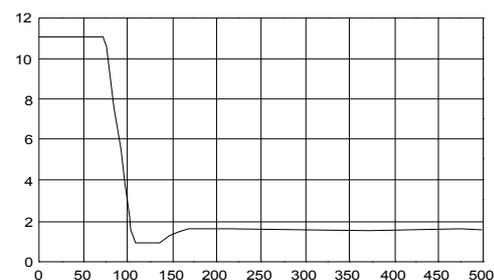


Carrera

b) Situación contraria.



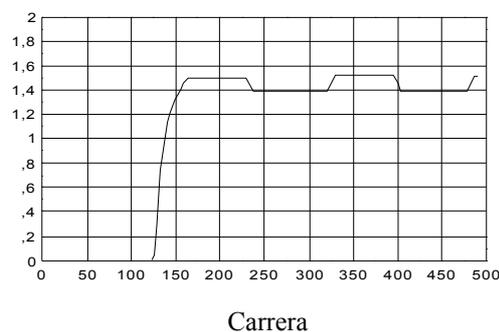
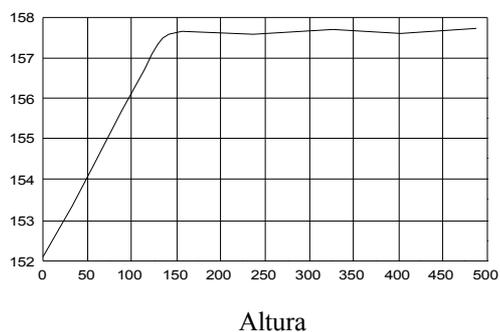
Altura



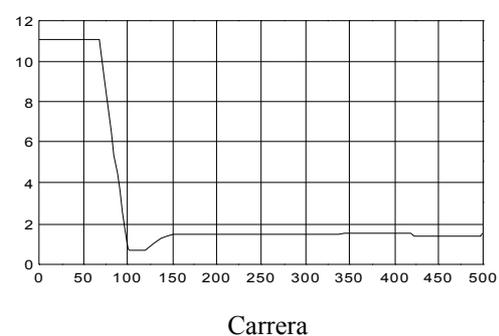
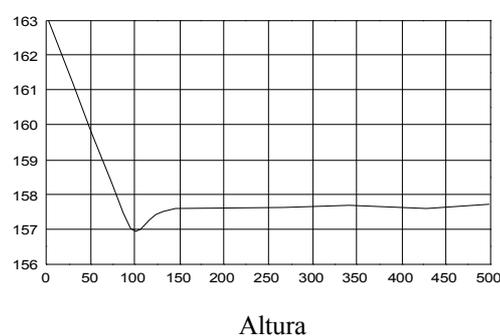
Carrera

1 Altura de referencia 75 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:



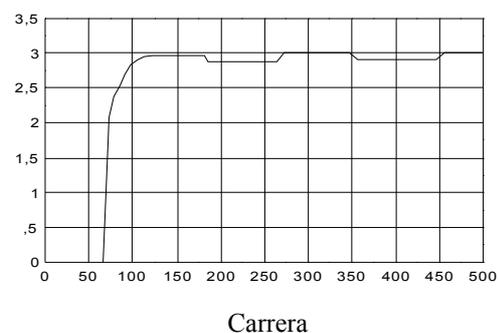
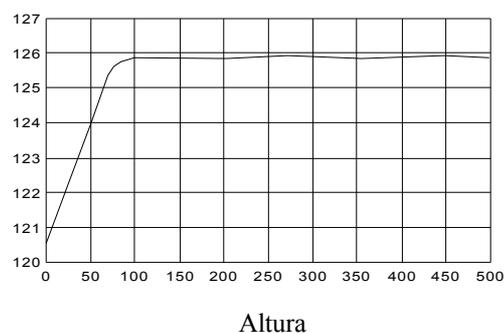
b) Situación contraria:



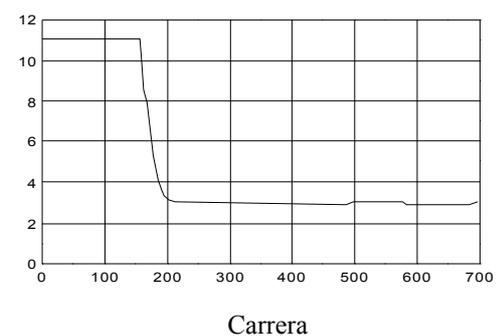
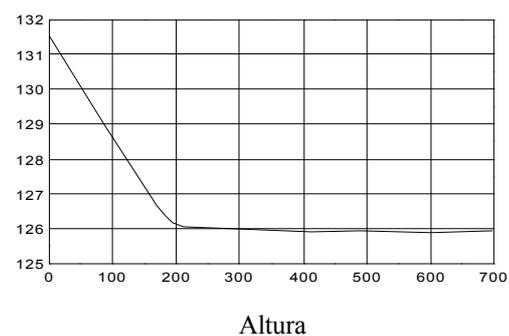
2) Caudal de entrada 82 cm³/s.

1 Altura de referencia 25 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

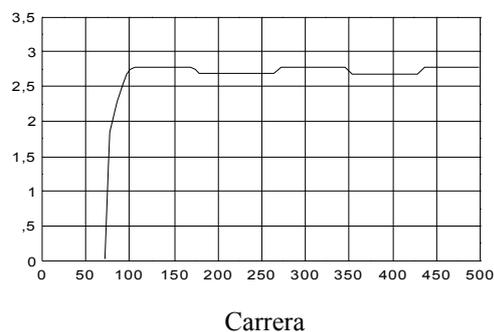
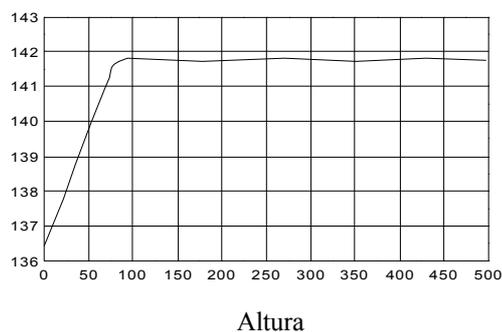


b) Situación contraria:

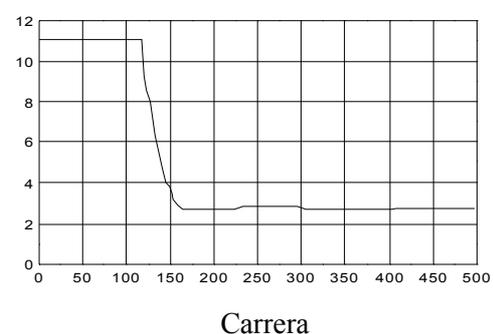
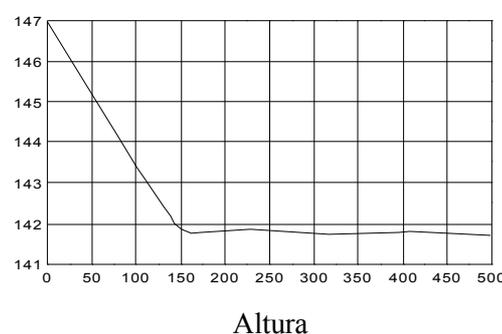


1 Altura de referencia 50 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

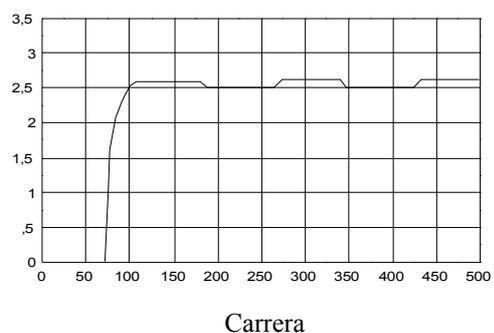
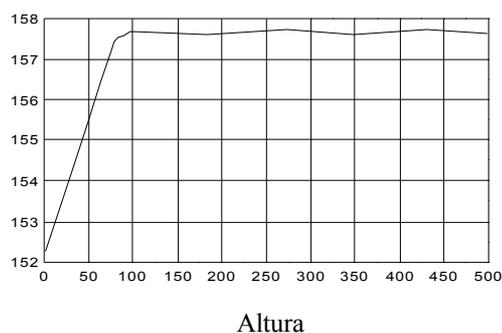


b) Situación contraria:

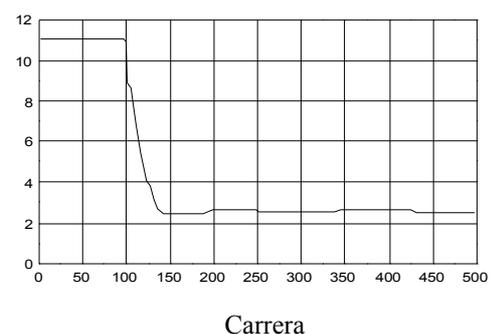
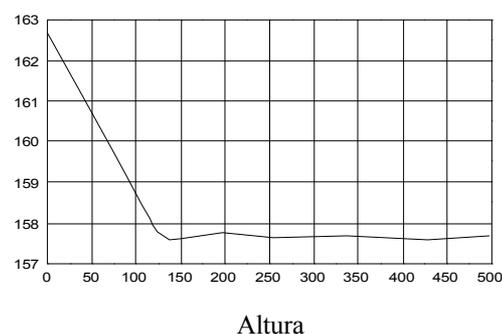


1 Altura de referencia 75 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:



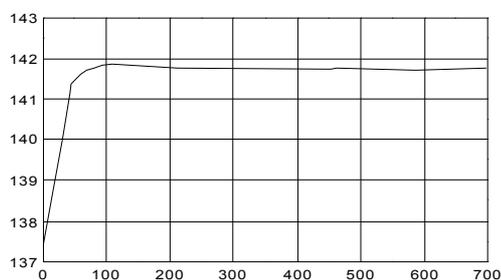
b) Situación contraria:



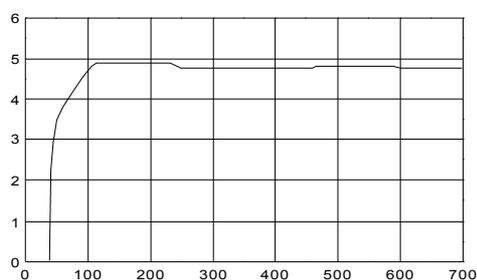
3) Caudal de entrada $113 \text{ cm}^3/\text{s}$.

1 Altura de referencia 50 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

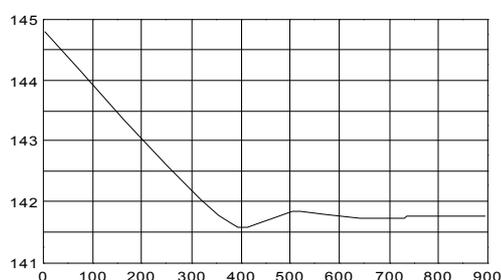


Altura

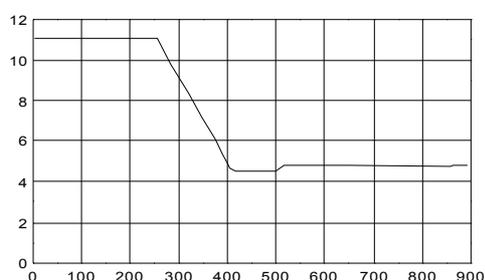


Carrera

b) Situación contraria:



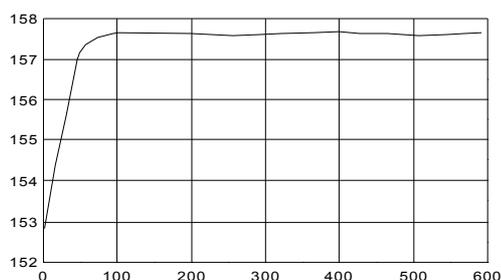
Altura



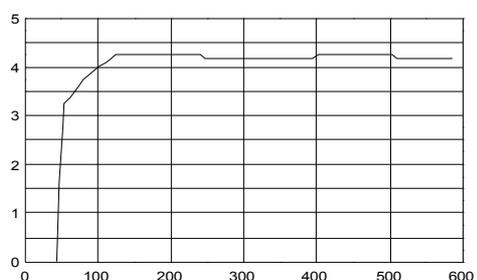
Carrera

1 Altura de referencia 75 %.

a) Situación en que el nivel del líquido tiene que aumentar para alcanzar al deseado:

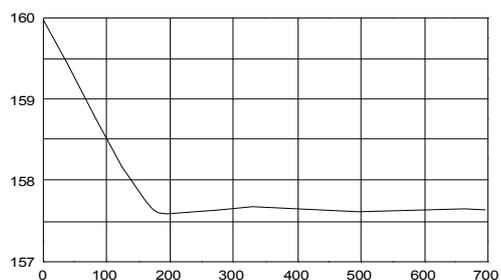


Altura

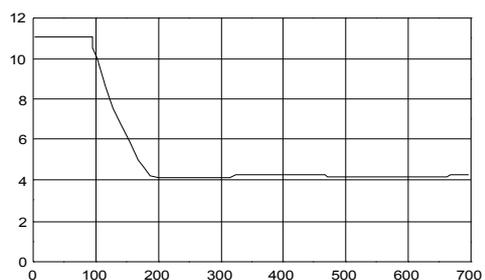


Carrera

b) Situación contraria:



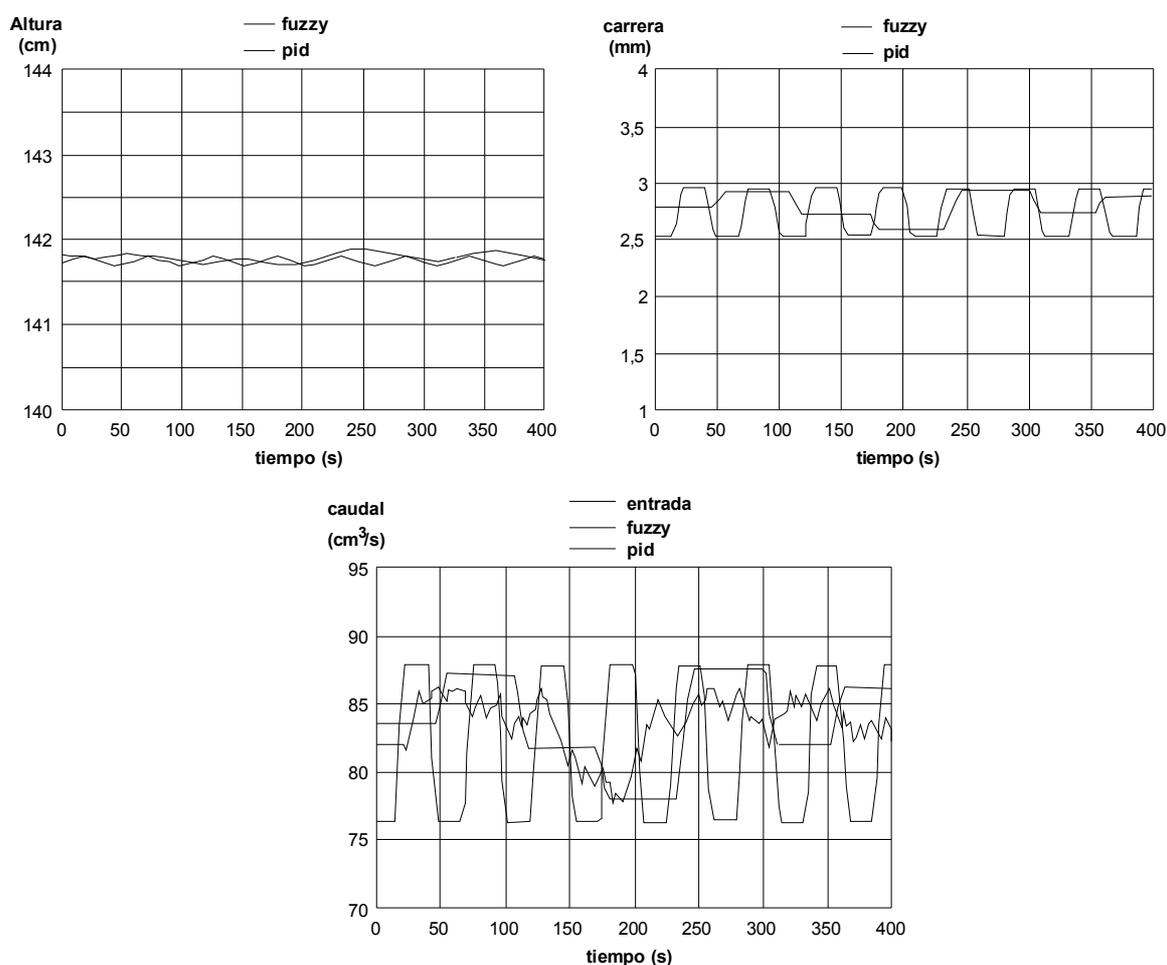
Altura



Carrera

Podemos observar, respecto de las gráficas obtenidas para el PID (pág. 92 a 95), que se mejora sensiblemente el comportamiento del sistema, no en cuanto al mantenimiento del nivel el líquido en el depósito, que prácticamente se mantiene en torno al de referencia, sino en la disminución de las variaciones de la carrera. Lo que realmente ha sucedido es que se ha conseguido que la respuesta del sistema completo responda de una forma más eficiente que en aquel caso. Por otra parte, al emplear un controlador no lineal, hecho a la medida del proceso, y de acuerdo con la experiencia sobre el mismo, se han distinguido claramente las dos situaciones características: cuando el nivel tiene que aumentar para alcanzar la referencia, y cuando tiene que disminuir.

También se ha estudiado el caso de la regulación del nivel de líquido cuando existen fluctuaciones en el caudal de entrada al depósito. También se observa una ligera mejoría respecto del PID, provocada al obligar al sistema a responder de forma más precisa con el controlador borroso. Como ejemplo, se presenta el caso en que el valor en torno al cual se producen las fluctuaciones en el caudal de entrada es de $82 \text{ cm}^3/\text{s}$, y una altura de referencia del 50 %. Puede apreciarse que la regulación del nivel en ambos casos es muy parecida, mientras que las variaciones en la carrera de la válvula son mayores para el PID.



Sin embargo, debemos señalar que el trabajo que supone ajustar los parámetros de un PID son mucho más rápidos y menos laboriosos que para ajustar los del controlador borroso. En efecto, mientras que un PID viene caracterizado por tres valores (constantes proporcional, integral y derivativa), en un controlador borroso deben considerarse todos los elementos que intervienen en su construcción: universo del discurso, número de términos lingüísticos para las variables de entrada y salida, centro y forma de las funciones de pertenencia para cada uno de ellos, dimensión de la tabla de reglas, valores de K_1 , K_2 y K_3 , etc. Además, existen algunas dificultades para su perfecta implementación, como son la presencia imprescindible de un experto que suministre el conocimiento necesario, o el hecho de que una modificación en los parámetros del controlador obliga a una revisión de todo el conjunto de reglas para detectar posibles apariciones de inconsistencias o tendencias a la inestabilidad. Por otra parte, prácticamente no existe una teoría que nos indique los pasos a seguir para el diseño del controlador.

A pesar de todo ello, existen algunas ventajas que debemos señalar:

- 1 No es necesario un modelo preciso del sistema a controlar. Ya estudiamos en el capítulo primero, que el diseño clásico de controladores se basa en un conocimiento exhaustivo del proceso, a través de la descripción del mismo mediante ecuaciones matemáticas. Cuando el proceso es complejo, encontrar un algoritmo de control puede llegar a ser una tarea imposible.
- 1 Se implementan fácilmente los conocimientos del operador, mediante reglas expresadas en términos lingüísticos.
- 1 Resulta posible alcanzar con facilidad las especificaciones de tiempo y transitorios prefijadas.
- 1 El controlador fuzzy es poco sensible a cambios de los parámetros del sistema, al ser no lineal.
- 1 Presenta un alto rechazo al ruido.
- 1 Permiten contemplar situaciones excepcionales del estado del proceso, gracias a su forma de representar el conocimiento.

Por todo ello, el control fuzzy tiene algunas particularidades que lo hacen claramente susceptible de ser utilizado en el control de procesos, en particular cuando se reúnen algunas de las siguiente características:

- Procesos difíciles de automatizar, o de dificultosa estimación de sus parámetros, y que, paradójicamente, son controlados fácilmente por operadores humanos.
- Procesos con incertidumbre, complejos, no lineales, de orden elevado, variantes con el tiempo.
- Situaciones en las que resulta difícil la medición del valor de las variables a controlar, etc.

6

CONCLUSIONES

Después de realizar un profundo estudio del proceso real, y de los elementos y dispositivos que incluye, se ha pasado a su simulación empleando técnicas de Inteligencia Artificial, en concreto de Sistemas Expertos. Las ventajas que supone mantener la base de conocimiento separada totalmente de los mecanismos de inferencia y control, permiten ampliar y modificar rápidamente, y de forma muy sencilla, este conocimiento. Aunque existen lenguajes de simulación específicos, orientados a bloques, etc., y más efectivos que el que nosotros hemos empleado, se han intentado mostrar las cualidades del empleo de técnicas de este tipo para dirigir la simulación. Comentar además que esencialmente se ha empleado el encadenamiento hacia adelante, pues el razonamiento en sistemas de control se mueve de las observaciones a las conclusiones. Debemos señalar también que con muy pocas reglas se consiguen resultados ampliamente satisfactorios, que en parte derivan del amplio estudio de la aplicación.

Por otra parte, se ha llevado a cabo un estudio pormenorizado del controlador borroso, haciendo hincapié en que las acciones de control están guiadas por los términos lingüísticos empleados por el propio experto del proceso, estableciendo relaciones cualitativas entre las variables del mismo, y utilizando un algoritmo de control altamente no lineal. De esta forma pueden resolverse problemas de control donde la aproximación tradicional no es del todo satisfactoria, sobre todo en aquellas situaciones en que los procesos son difíciles de caracterizar. En nuestro caso concreto, se ha implementado un controlador borroso, y se ha comparado con un PID convencional. La utilización de un sencillo PID suele ser suficiente cuando no se requieren respuestas muy precisas, e incluso podemos decir que con él se controla relativamente bien el proceso de nivel. Pero cuando se requieren respuestas más precisas, la utilización de un controlador borroso puede sernos de gran ayuda, obteniendo resultados mejores, precisamente de la característica no lineal del mismo.

Finalmente, debe comentarse que es muchos casos un control convencional bien ajustado, o un control adaptativo es más satisfactorio que intentar sustituirlo por un controlador inteligente muy sofisticado.

APÉNDICE I

Este procedimiento es el que realiza la acción de control del controlador borroso:

```

proc-fuzzy(e:float, ce:float, sc:float)
nge,nme,npe,zee,ppe,pme,pge,
ngc,nmc,npc,zec,ppc,pmc,pgc: truth-value;
a1,a2,a3,a4,a5,a6,a7,b1,b2,b3,b4,b5,b6,b7: truth-value;

{9 etiquetas lingüísticas variable salida}
ng:float=-3.8;
nm:float=-1.8;
np:float=-1.1;
npp:float=-0.5;
ze:float=0.0;
ppp:float=0.5;
pp:float=1.3;
pm:float=1.7;
pg:float=3.7;

x0:float=0.0;      {Numerador del c.d.g.}
x1:float=0.0;      {Denominador del c.d.g.}
x,k3:float;
zero:truth-value = truth-value(-1);

begin
  if e>0 then e=e*12.5 else e=e*10.0;
  if ce < 0 then ce=57.0*ce else ce=120.0*ce;
  if abs(e)>2.5 then k3=1.0 else k3=0.1;

{Defuzzyfication: 7 etiquetas, tanto para E como para CE}
{e}
  nge = (e < -2.5 (+- 0.5));  a1 = nge is more true than zero;
  nme = (e = -2.0 (+- 1.0));  a2 = nme is more true than zero;
  npe = (e = -1.0 (+- 1.0));  a3 = npe is more true than zero;
  zee = (e = 0.0 (+- 1.0));   a4 = zee is more true than zero;
  ppe = (e = 1.0 (+- 1.0));   a5 = ppe is more true than zero;
  pme = (e = 2.0 (+- 1.0));   a6 = pme is more true than zero;
  pge = (e > 2.5 (+- 0.5));   a7 = pge is more true than zero;

{ce}
  ngc = (ce < -2.5 (+- 0.5));  b1 = ngc is more true than zero;
  nmc = (ce = -2.5 (+- 1.0));  b2 = nmc is more true than zero;
  npc = (ce = -1.0 (+- 1.0));  b3 = npc is more true than zero;
  zec = (ce = 0.0 (+- 1.0));   b4 = zec is more true than zero;
  ppc = (ce = 1.0 (+- 1.0));   b5 = ppc is more true than zero;
  pmc = (ce = 2.0 (+- 1.0));   b6 = pmc is more true than zero;
  pgc = (ce > 2.5 (+- 0.5));   b7 = pgc is more true than zero;

{Tabla de reglas y Defuzzyfication}
if a1 then begin
  if b1 then begin x=.5*(quantity(nge and ngc)+1); x0 = x0 + ng*x; x1=x1+x; end;
  if b2 then begin x=.5*(quantity(nge and nmc)+1); x0 = x0 + ng*x; x1=x1+x; end;
  if b3 then begin x=.5*(quantity(nge and npc)+1); x0 = x0 + ng*x; x1=x1+x; end;
  if b4 then begin x=.5*(quantity(nge and zec)+1); x0 = x0 + npp*x; x1=x1+x; end;
  if b5 then begin x=.5*(quantity(nge and ppc)+1); x0 = x0 + npp*x; x1=x1+x; end;
  if b6 then begin x=.5*(quantity(nge and pmc)+1); x0 = x0 + npp*x; x1=x1+x; end;
  if b7 then begin x=.5*(quantity(nge and pgc)+1); x0 = x0 + npp*x; x1=x1+x; end;
end;
if a2 then begin
  if b1 then begin x=.5*(quantity(nme and ngc)+1); x0 = x0 + ng*x; x1=x1+x; end;
  if b2 then begin x=.5*(quantity(nme and nmc)+1); x0 = x0 + nm*x; x1=x1+x; end;
  if b3 then begin x=.5*(quantity(nme and npc)+1); x0 = x0 + nm*x; x1=x1+x; end;
  if b4 then begin x=.5*(quantity(nme and zec)+1); x0 = x0 + npp*x; x1=x1+x; end;
  if b5 then begin x=.5*(quantity(nme and ppc)+1); x0 = x0 + pm*x; x1=x1+x; end;

```

```

        if b6 then begin x=.5*(quantity(nme and pmc)+1); x0 = x0 + pg*x; x1=x1+x; end;
        if b7 then begin x=.5*(quantity(nme and pgc)+1); x0 = x0 + pg*x; x1=x1+x; end;
    end;
    if a3 then begin
        if b1 then begin x=.5*(quantity(npe and ngc)+1); x0 = x0 + nm*x; x1=x1+x; end;
        if b2 then begin x=.5*(quantity(npe and nmc)+1); x0 = x0 + nm*x; x1=x1+x; end;
        if b3 then begin x=.5*(quantity(npe and npc)+1); x0 = x0 + np*x; x1=x1+x; end;
        if b4 then begin x=.5*(quantity(npe and zec)+1); x0 = x0 + npp*x; x1=x1+x; end;
        if b5 then begin x=.5*(quantity(npe and ppc)+1); x0 = x0 + pp*x; x1=x1+x; end;
        if b6 then begin x=.5*(quantity(npe and pmc)+1); x0 = x0 + pm*x; x1=x1+x; end;
        if b7 then begin x=.5*(quantity(npe and pgc)+1); x0 = x0 + pg*x;x1=x1+x; end;
    end;
    if a4 then begin
        if b1 then begin x=.5*(quantity(zee and ngc)+1); x0 = x0 + ng*x; x1=x1+x; end;
        if b2 then begin x=.5*(quantity(zee and nmc)+1); x0 = x0 + ng*x; x1=x1+x; end;
        if b3 then begin x=.5*(quantity(zee and npc)+1); x0 = x0 + np*x; x1=x1+x; end;
        if b4 then begin x=.5*(quantity(zee and zec)+1); x0=x0 + ze*x; x1=x1+x; end;
        if b5 then begin x=.5*(quantity(zee and ppc)+1); x0 = x0 + pp*x; x1=x1+x; end;
        if b6 then begin x=.5*(quantity(zee and pmc)+1); x0 = x0 + pm*x; x1=x1+x; end;
        if b7 then begin x=.5*(quantity(zee and pgc)+1); x0 = x0 + pg*x; x1=x1+x; end;
    end;
    if a5 then begin
        if b1 then begin x=.5*(quantity(ppe and ngc)+1); x0 = x0 + ng*x; x1=x1+x;end;
        if b2 then begin x=.5*(quantity(ppe and nmc)+1); x0 = x0 + ng*x; x1=x1+x; end;
        if b3 then begin x=.5*(quantity(ppe and npc)+1); x0 = x0 + np*x; x1=x1+x; end;
        if b4 then begin x=.5*(quantity(ppe and zec)+1); x0 = x0 + ppp*x; x1=x1+x; end;
        if b5 then begin x=.5*(quantity(ppe and ppc)+1); x0 = x0 + pp*x; x1=x1+x; end;
        if b6 then begin x=.5*(quantity(ppe and pmc)+1); x0 = x0 + pg*x; x1=x1+x; end;
        if b7 then begin x=.5*(quantity(ppe and pgc)+1); x0 = x0 + pg*x; x1=x1+x; end;
    end;
    if a6 then begin
        if b1 then begin x=.5*(quantity(pme and ngc)+1); x0 = x0 + ng*x; x1=x1+x; end;
        if b2 then begin x=.5*(quantity(pme and nmc)+1); x0 = x0 + ng*x; x1=x1+x; end;
        if b3 then begin x=.5*(quantity(pme and npc)+1); x0 = x0 + nm*x; x1=x1+x; end;
        if b4 then begin x=.5*(quantity(pme and zec)+1); x0 = x0 + ppp*x; x1=x1+x; end;
        if b5 then begin x=.5*(quantity(pme and ppc)+1); x0 = x0 + pm*x; x1=x1+x; end;
        if b6 then begin x=.5*(quantity(pme and pmc)+1); x0 = x0 + pg*x; x1=x1+x; end;
        if b7 then begin x=.5*(quantity(pme and pgc)+1); x0 = x0 + pg*x; x1=x1+x; end;
    end;
    if a7 then begin
        if b1 then begin x=.5*(quantity(pge and ngc)+1); x0 = x0 + ppp*x; x1=x1+x; end;
        if b2 then begin x=.5*(quantity(pge and nmc)+1); x0 = x0 + pp*x; x1=x1+x; end;
        if b3 then begin x=.5*(quantity(pge and npc)+1); x0 = x0 + pm*x; x1=x1+x; end;
        if b4 then begin x=.5*(quantity(pge and zec)+1); x0 = x0 + pg*x; x1=x1+x; end;
        if b5 then begin x=.5*(quantity(pge and ppc)+1); x0 = x0 + pg*x; x1=x1+x; end;
        if b6 then begin x=.5*(quantity(pge and pmc)+1); x0 = x0 + pg*x; x1=x1+x; end;
        if b7 then begin x=.5*(quantity(pge and pgc)+1); x0 = x0 + pg*x; x1=x1+x; end;
    end;

    conclude that the señal-control of the controlador connected at the output of SN-1=
        min(14.6,max(7.9, sc + k3*(x0/x1)))

    end
    {Fin del procedimiento}

```

APÉNDICE II

- whenever any (OBJETO) receives a value then (...) = cuando cualquier OBJETO recibe un nuevo valor, entonces (...)
- conclude that (...) = concluir que (...)
- the (ATRIBUTO) of (OBJETO) = el (ATRIBUTO) de (OBJETO)
- the (ATRIBUTO) of every (OBJETO) = el (ATRIBUTO) de todos los (OBJETO)
- connected at the input/output of (...) = conectado a la entrada/salida de (...)
- for any (OBJETO) = para todo (OBJETO).
- change the (REGION) stripe-color of (OBJETO) to (COLOR) = cambiar el color de la (REGION) de (OBJETO) a (COLOR).
- the symbol (...) = el símbolo (...)
- deactivate/activate the subworkspace of (OBJETO) = desactivar/activar el subespacio de (OBJETO).
- the value of the (ATRIBUTO) of (OBJETO) as of (NUMERO) seconds ago = el valor del (ATRIBUTO) de (OBJETO) hace (NUMERO) segundos.
- move = mover
- state variable : next value of (...) = variable de estado: valor siguiente de (...)
- with initial value (NUMERO) = con valor inicial.
- with time increment (NUMERO) seconds = con incremento temporal de (NUMERO) segundos.

BIBLIOGRAFÍA

- [1] AGUILAR, J. *Representación Simbólico-Numérica para Sistemas Expertos de Control en Tiempo Real*, en "La Inteligencia Artificial y el Control en Tiempo Real", Ed. IIC y Repsol, pp. 241-259. 1992.
- [2] ALAMAN, X. *Introducción a la Inteligencia Artificial para la Asistencia al Control en Tiempo Real*, en "La Inteligencia Artificial y el Control en Tiempo Real", Ed. IIC y Repsol, pp. 15-55. 1992.
- [3] ALEIXANDRE, V. et al. *Automática I*, Tomos I y II. Universidad Nacional de Educación a Distancia, 1987.
- [4] ALVAREZ, R. *A la búsqueda de un Control Industrial Más Inteligente*. *Automática e Instrumentación*, nº. 217, pp. 59-62. Octubre, 1991.
- [5] CHANTLER, M.J. *Real Time Aspects of Expert Systems in Process Control*. Herriot-Watt Univ. 1987.
- [6] CHIEN, Y. y LIEBOWITZ, J. *Expert Systems in the SDI Environment*. *IEEE Computer*, pp. 115-121. Julio, 1986.
- [7] COWELL, R.G. et al. *Sequential Model Criticism in Probabilistic Expert Systems*. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, V. 15, nº 3. pp. 209-219. Marzo, 1993.
- [8] CREUS, A. *Instrumentación Industrial*. Ed. Marcombo-Boixareu, 1979.
- [9] CREUS, A. *Válvulas de Control de Caudal*. *Automática e Instrumentación*, informes 87-88, pp. 169-186. 1988.
- [10] CUBEIRO, A.M. *Control de Procesos: Regulador de Nivel Experimental*. Trabajo de Grado, Universidad de Salamanca, 1984.
- [11] FAUGHT, N.S. *Applications of AI in Engineering*. *IEEE Computer*, pp. 17-27. Julio, 1986.
- [12] FINK, P.K. et al. *A General Expert System Design for Diagnostic Problem Solving*. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, V. PAMI-7, nº. 5, pp. 553-560. Septiembre, 1985.
- [13] *Fuzzy Sets and Decision Analysis*. Ed. Zimmermann H, Zadeh L.A y Gaines, B.R., Elsevier Science Publisher B.V. 1984.
- [14] GALAN, R. *La Inteligencia Artificial desde la Perspectiva del Control Automático*, en "La Inteligencia Artificial y el Control en Tiempo Real", Ed. IIC y Repsol, pp. 97-113. 1992.
- [15] GARCIA-CEREZO, A.J. *Aplicaciones Actuales de la Lógica Borrosa*. *Automática e Instrumentación*, nº. 216, pp. 113-119. Septiembre, 1991.
- [16] GARCIA-CEREZO, A.J. y OLLERO, A. *Diseño e Implementación de Controladores Heurísticos Mediante Razonamiento Aproximado*. *Actas del VI Congreso de Informática y Automática*, pp. 153-163. Madrid, Octubre, 1985.
- [17] GENSYM CO. *Reference Manual for G2 Version 3.0*, Gensym Corporation, 1992.
- [18] HONG, S.J. *Introducción del editor*, en *IEEE Computer*, pp. 12-15. Julio, 1986.
- [19] JIMENEZ, A. y MATIA, F. *Aplicaciones de Fuzzy Logic en Tiempo Real*, en "La Inteligencia Artificial y el Control en Tiempo Real", Ed. IIC y Repsol, pp. 189-239. 1992.
- [20] KESHAVAN, H.K. et al. *Introduction to the Special Section on Probabilistic Reasoning*. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, V. 15, nº. 3, pp. 193-195. Marzo, 1993.
- [21] LEUNG, K.S y WONG, M.H. *An Expert-System Shell Using Structured Knowledge*. *IEEE Computer*, V. 23, nº. 3, pp. 38-47. Marzo, 1990.
- [22] LEUNG, K.S. y LAM, W. *Fuzzy Concepts in Expert Systems*. *IEEE Computer*, V. 21, nº. 9, pp. 43-56. Septiembre, 1988.

- [23] MITCHELL, T. et al. *A Knowledge-Based Approach to Design*. IEEE Trans. on Pattern Analysis and Machine Intelligence, V. PAMI-7, nº. 5, PP. 502-510. Septiembre, 1985.
- [24] MITTAL, S. et al. *PRIDE: An Expert System for the Design of Paper Handling Systems*. IEEE Computer, pp. 102-114. Julio, 1986.
- [25] OGATA, K. *Ingeniería de Control Moderna*. Ed. Prentice-Hall, 1974.
- [26] OLLERO, A. *Los Sistemas Expertos en Control de Procesos*. Automática e Instrumentación, nº. 193, pp. 171-180. Julio, 1989.
- [27] OLLERO, A. y GARCIA-CEREZO, A. *Inteligencia Artificial: Aplicaciones Recientes en Control y Supervisión de Procesos*. Automática e Instrumentación, nº. 188, pp. 153-164. Febrero, 1989.
- [28] PEREZ, B. *Diseño de un Sistema de Control de Procesos: Controladores PID y Deadbeat*. Trabajo de Grado, Universidad de Salamanca, 1991.
- [29] *Practical Process Instrumentation and Control*. Ed. The Staff of Chemical Engineering, New York, McGraw-Hill, 1980.
- [30] PRADÉ, H. *A Computational Approach to Approximate and Plausible Reasoning with Applications to Expert Systems*. IEEE Trans. on Pattern Analysis and Machine Intelligence, V. PAMI-7, nº. 3. Mayo, 1985.
- [31] PUIG, F. *La investigación en Automática y Robótica en Universidades y Centros Públicos de Investigación*. Automática e Instrumentación, nº 179, pp.127-144. Abril, 1988.
- [32] ROLSTON, D.W. *Principios de Inteligencia Artificial y Sistemas Expertos*. Ed. McGraw-Hill, 1990.
- [33] SANZ, R. et al. *Introducción al Control Inteligente de Procesos*. Informática y Automática, Vol. 24-1, pp. 5-17. 1991.
- [34] SELF, K. *Designing with Fuzzy Logic*. IEEE Spectrum, pp. 42-44. Noviembre, 1990.
- [35] SMETS, P. *The Combination of Evidence in the Transferable Belief Model*. IEEE Trans. on Pattern Analysis and Machine Intelligence, V. 12, nº. 5, pp. 447-458. Mayo, 1990.
- [36] STOCK M. *AI in Process Control*. Ed. McGraw-Hill, 1989.
- [37] SUGENO, M. *Industrial Applications of Fuzzy Control*. Ed. M. SUGENO, Elsevier Science Publisher B.V. 1985.
- [38] TAUNTON, J.C. *Empleo de Sistemas Expertos en Línea en Control de Procesos*. Automática e Instrumentación, nº. 178, pp.181-185. Marzo, 1988.
- [39] TZAFESTAS, S.G. et al. *Expert-system Methodology in Process Supervision and Control*. Univ. Zagrau, Atenas, 1988.
- [40] VENKATASUBRAMANIAS, V. y RICH, S.H. *An Object-Oriented Two-Tier Architecture for Integrating Compiled and Deep-level Knowledge for Process Diagnosis*. Computer Chemical Engineering, Vol. 12, nº 9-10, pp. 903-921. 1988.
- [41] VERBRUGGEN H.B. y ÅSTRÖM, K.J. *Artificial Intelligence and Feedback Control*. IFAC Workshop on Artificial Intelligence in Real Time Control, Septiembre, 1989.
- [42] VERDEJO, M.F. *Aplicaciones de los Sistemas Expertos en la Industria*. Automática e Instrumentación, pp. 197-201. Marzo, 1986.
- [43] VIÑA, A. *El Control Automático desde la Perspectiva de la Inteligencia Artificial*, en "La Inteligencia Artificial y el Control en Tiempo Real", Ed. IIC y Repsol, pp. 115-155. 1992.
- [44] WHITE, F.M. *Mecánica de Fluidos*. Ed. McGraw-Hill, 1983.
- [45] ZACCAGNINI, J.L. *La Interfaz de Usuario en los Sistemas de Control: Aspectos Ergonómicos*, en "La Inteligencia Artificial y el Control en Tiempo Real", Ed. IIC y Repsol, pp. 359-399. 1992.
- [46] ZADEH, L.A. *Commonsense Knowledge Representation Based on Fuzzy Logic*. IEEE Computer, V. 16, nº. 10, pp.61-65. Octubre, 1983.