

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 399 795**

51 Int. Cl.:

G05B 9/03 (2006.01)

G05B 11/42 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **08.04.2009 E 09747076 (9)**

97 Fecha y número de publicación de la concesión europea: **26.12.2012 EP 2277086**

54 Título: **Procedimiento y sistema para un conmutador de control PID sin perturbaciones**

30 Prioridad:

12.05.2008 US 119053

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

03.04.2013

73 Titular/es:

**GE INTELLIGENT PLATFORMS, INC. (100.0%)
Route 29 North & Route 606 P.O. Box 8106
Charlottesville, VA 22911, US**

72 Inventor/es:

MCNAB, KEITH, GEORGE

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 399 795 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Procedimiento y sistema para un conmutador de control PID sin perturbaciones

Antecedentes de la invención

5 La presente invención versa, en general, acerca de un control continuo de procesos (CPC) y, más en particular, acerca de la configuración dinámica de un algoritmo de CPC.

10 Un algoritmo usado comúnmente para el control continuo de procesos se denomina algoritmo "PID", refiriéndose el término "PID" a proporcional, integral y derivativo. En los sistemas de control industrial hay muchos tipos diferentes de algoritmos PID. Los tipos más comunes de algoritmos PID son o bien "no interactivos" o bien "interactivos". Con un algoritmo interactivo, los términos proporcional, integral y derivativo se combinan de una manera en la que los términos interactúan; por ejemplo, los términos son determinados en serie. Con un algoritmo no interactivo, los términos proporcional, integral y derivativo se combinan de una manera en la que los términos interactúan; por ejemplo, los términos son determinados en paralelo.

15 Cada tipo de algoritmo tiene beneficios en diferentes aplicaciones y en diferentes modos de control. En una aplicación de control puede resultar deseable conmutar una y otra vez de un tipo de algoritmo de control al otro tipo de algoritmo de control para determinar heurísticamente el mejor algoritmo para la aplicación. Además, en el curso de la operación del sistema, pueden surgir diferentes objetivos de control y condiciones operativas que harían deseable conmutar de un tipo de algoritmo de control al otro tipo de algoritmo de control. Un ejemplo sería un control de temperaturas de intervalo dividido en el que la salida PID está dividida en dos intervalos: uno para calentar y otro para enfriar. El tipo deseado de algoritmo para el intervalo de calentamiento puede ser diferente del tipo deseado de algoritmo para el intervalo de enfriamiento. En una aplicación de control continuo de procesos (CPC), detener el sistema de control para cambiar el algoritmo de control puede costar una cantidad exorbitante de tiempo y de dinero.

20 El documento US 4.587.470 da a conocer un sistema que incluye un operador PID analógico dotado de un controlador principal y un operador PID digital dotado de un controlador complementario y adaptado para llevar a cabo el cálculo PID equivalente al del operador PID analógico. Se proporciona un sumador/restador que calcula la diferencia, como error de control, entre el componente integral de la salida del operador PID analógico y el componente integral de la salida del operador PID analógico. Se usa un elemento de tiempo muerto de primer orden para evitar que el operador PID analógico haga un seguimiento del estado en circuito abierto cuando opera el relé en la unidad de conmutación.

25 En una tentativa por evitar tal pérdida de tiempo y dinero, algunos algoritmos de CPC permiten una configuración dinámica, lo que significa conmutar entre tipos de algoritmo PID sin requerir que se detenga el sistema de control. Con tal configuración dinámica, puede haber un cambio abrupto en la salida del algoritmo del sistema de control. Tal cambio abrupto puede hacer que el proceso controlado se vuelva inestable, dando posiblemente como resultado una parada de emergencia y/o la degradación del producto final.

Breve descripción de la invención

35 La presente invención proporciona un procedimiento de conmutación de un primer algoritmo de control a un segundo algoritmo de control en un sistema de control continuo de procesos según, se define en la reivindicación 1, y un controlador para un sistema de control continuo de procesos, según se define en la reivindicación 6.

Breve descripción de los dibujos

40 La Figura 1 es un diagrama esquemático de bloques de un sistema de control continuo de procesos; la Figura 2 es un diagrama de bloques de un algoritmo PID de tipo en serie; la Figura 3 es un diagrama de bloques de un algoritmo PID de tipo en paralelo; la Figura 4 es un diagrama de bloques de los tipos en serie y en paralelo de algoritmos PID y de un conmutador dinámico "sin perturbaciones"; la Figura 5 es un diagrama de bloques de un algoritmo PID de tipo en serie para su uso con el conmutador dinámico sin perturbaciones ilustrado en la Figura 4; y la Figura 6 es un diagrama de bloques de un algoritmo PID de tipo en paralelo para su uso con el conmutador dinámico sin perturbaciones ilustrado en la Figura 4.

Descripción detallada de la invención

50 La siguiente descripción detallada ilustra realizaciones de la invención a título de ejemplo y no a modo de limitación. Se contempla que la invención tenga aplicación general a sistemas de control continuo de procesos y no esté limitada a su implementación en conexión con ningún sistema particular cualquiera de ese tipo.

Tal como se usa en el presente documento, debería entenderse que un elemento o una etapa enumerados en singular y precedidos de las palabras "un" o "una" no excluyen elementos ni etapas plurales, a no ser que tal exclusión se señale de forma explícita. Además, no se busca que se interprete que las referencias a "una

realización" de la presente invención excluyan la existencia de realizaciones adicionales que también incorporen las características enumeradas.

El término procesador, tal como se usa en el presente documento, se refiere a unidades centrales de proceso, microprocesadores, microcontroladores, circuitos de conjuntos reducidos de instrucciones (RISC), circuitos integrados para aplicaciones específicas (ASIC), circuitos lógicos y cualquier otro circuito, procesador o controlador capaz de ejecutar las funciones descritas en el presente documento. Tal como se usan en el presente documento, las expresiones "soporte lógico" y "soporte lógico inalterable" son intercambiables e incluyen cualquier programa de ordenador guardado en memoria para su ejecución por un procesador, incluyendo memoria RAM, memoria ROM, memoria EPROM, memoria EEPROM y memoria RAM no volátil (NVRAM). Los anteriores tipos de memoria son únicamente ejemplares y, por ello, no son limitantes en cuanto a los tipos de memoria utilizables para el almacenamiento de un programa de ordenador.

La Figura 1 es un diagrama esquemático de bloques de un sistema de control continuo de procesos. El sistema incluye controladores o procesadores centrales acoplados a un bus de entrada/salida (I/O) y a un bus de campo. El tipo seleccionado de bus de entrada/salida y de bus de campo depende del tipo particular de sistema que esté controlándose, y puede usarse uno cualquiera de numerosos buses de entrada/salida y buses de campo conocidos disponibles en el mercado. El bus de entrada/salida y el bus de campo están acoplados a diversos sensores y controladores locales acoplado al sistema que ha de ser controlado (no mostrado). De nuevo, los sensores y los controladores locales particulares dependen del tipo particular de sistema que se esté controlando, como es bien conocido en la técnica.

Los controladores centrales también están acoplados a una red de tipo Ethernet. Hay acopladas a la red estaciones de trabajo de diseño para su uso en conexión con el diseño, la creación y la configuración del sistema de mantenimiento. También están acopladas a la red consolas de operarios para que los operarios monitoricen y controlen el proceso. Un subsistema de base de datos, a veces denominado historial, acoplado a la red, gestiona el almacenamiento y la recuperación de los datos del proceso. El subsistema de base de datos también proporciona un control de versiones para las estrategias de control del proceso, incluyendo prestaciones de registros de auditoría.

La Figura 1 ilustra una de numerosas arquitecturas ejemplares para un sistema de control continuo de procesos. Los algoritmos PID actuales no están limitados a un uso en conexión con ningún sistema particular cualquiera. Un ejemplo de un sistema de control conocido y disponible comercialmente es el sistema de procesos Proficy®, disponible comercialmente en la General Electric Company, Fairfield, Connecticut, EE. UU.

En operación, los controladores centrales reciben datos de diversos sensores situados en puntos seleccionados de datos del sistema que ha de controlarse. Los datos recibidos son guardados por el subsistema de gestión de la base de datos. además, tales datos recibidos pueden ser usados por los controladores centrales y los controladores locales para hacer ajustes a los componentes del sistema controlado.

Diversos algoritmos de control son almacenados y ejecutados por los controladores centrales. Por ejemplo, los algoritmos de control continuo de procesos son almacenados en controladores y son ejecutados por controladores en conexión con la realización de un control continuo del sistema que se esté controlando. En la Figura 2 se ilustra un ejemplo de un algoritmo tal: un algoritmo PID en serie. En la Figura 3 se ilustra otro algoritmo ejemplar: un algoritmo PID en paralelo. Los parámetros referenciados en los algoritmos ilustrados en las Figuras 2 y 3 son presentados en lo que sigue.

PV(s) es la variable de proceso que ha de controlarse

SP(s) es el valor de referencia, es decir, un valor deseado de la variable de proceso

K_D es el tiempo derivativo en minutos

N es el parámetro de filtro derivativo

P(s) es el término proporcional

I(s) es el término integral

D(s) es el término derivativo

K_P es la ganancia proporcional

PD(s) es el término proporcional y derivativo

K_I es la reposición integral en repeticiones por minuto

PID(s) es el término proporcional e integral y derivativo

Tal como se muestra en la Figura 2, la variable de proceso que ha de controlarse, PV(s), y el valor de referencia para esa variable de proceso, SP(s), se suman para obtener la diferencia Δ entre el valor de la variable y el valor de referencia. A continuación, se genera el término derivativo D(s) y luego se genera el término proporcional y derivativo PD(s). Después se suma el término proporcional y derivativo PD(s) con el término integral I(s) para generar el término proporcional e integral y derivativo PID. Tal como se muestra en la Figura 2, los términos proporcionales, integrales y derivativos se generan en serie y no tienen independencia mutua; es decir, los términos interactúan.

La Figura 3 ilustra un algoritmo PID en paralelo. Tal como se muestra en la Figura 3, la variable de proceso que ha de controlarse, PV(s), y el valor de referencia para esa variable de proceso, SP(s), se suman para obtener la diferencia Δ entre el valor de la variable y el valor de referencia. A continuación, se generan el término proporcional P(s), el término integral I(s) y el término derivativo D(s), con independencia mutua. Después se suman los términos P(s), I(s) y D(s) para generar el término proporcional e integral y derivativo (PID). Tal como se muestra en la Figura 3, los términos proporcionales, integrales y derivativos se generan en paralelo y tienen independencia mutua; es decir, los términos no interactúan.

La Figura 4 ilustra, en forma de diagrama de bloques, tipos en serie 80 y en paralelo 90 de algoritmos PID y un conmutador dinámico 100 “sin perturbaciones”. Tal como se usa en el presente documento, “sin perturbaciones” significa facilitar la evitación de cambios abruptos en las salidas del sistema de control que hagan que el proceso controlado se vuelva inestable o la degradación del producto final. Durante la operación, el controlador selecciona uno de los algoritmos PID para su ejecución, y el controlador hace que el conmutador seleccione la salida de tal algoritmo seleccionado para generar el PID de salida. Debido a los parámetros cambiantes, el controlador puede determinar conmutar de un algoritmo PID al otro algoritmo PID. Tal determinación puede realizarse, por ejemplo, con base en datos empíricos relacionados con el proceso que se esté controlando. Cuando se realiza tal determinación, el controlador hace que el conmutador seleccione la salida del otro algoritmo para generar el PID de salida.

El término integral, mostrado como I(s) tanto en la Figura 2 como en la 3, persiste entre ejecuciones del bloque de funciones. Si ocurre una conmutación entre algoritmos, el término integral, por su naturaleza, no cambiaría abruptamente. Por lo tanto, solo se compensan los términos proporcional y derivativo para garantizar una conmutación 100 “sin perturbaciones”.

Para compensar debidamente el término derivativo para una conmutación “sin perturbaciones”, debe considerarse el hecho de que el término derivativo pueda configurarse para que esté basado en el error o la variable de proceso. Los cambios escalonados en el valor de referencia podrían dar como resultado un salto derivativo, que es poco deseable y, por ende, en algunas aplicaciones de control PID, es deseable hacer que el término derivativo se base únicamente en la variable del proceso. Este hecho se suma a la complejidad de determinar la debida compensación requerida sobre el término derivativo para una conmutación sin perturbaciones en los algoritmos mostrados en las Figuras 2 y 3. Para minimizar esta complejidad en la Figura 5, que ilustra un algoritmo PID 80 en serie, y en la Figura 6, que ilustra un algoritmo PID 90 en paralelo, se aplica el principio de superposición para hacer que el término derivativo sea la suma de las respuestas individuales de las entradas PV(s) y SP(s) a la respectiva función de transferencia derivativa. Esto permite que el algoritmo haga la función de transferencia derivativa para la entrada SP(s) igual a la unidad cuando el término derivativo esté configurado para basarse únicamente en la variable de proceso.

Específicamente, con respecto a los algoritmos en serie y en paralelo ilustrados en las Figuras 5 y 6, y si el término derivativo D(s) se basa en el error, entonces:

$$A(s) = SP(s) \cdot \frac{K_D s + 1}{\frac{K_D}{N} s + 1}$$

$$C(s) = SP(s) \cdot \frac{K_D s}{\frac{K_D}{N} s + 1}$$

Si no, si el término derivativo se basa únicamente en la variable de proceso, entonces tanto A(s) como C(s) son iguales a SP(s).

Además, en las Figuras 5 y 6 se ilustra que para lograr una transferencia “sin perturbaciones” entre los tipos de algoritmo, debe compensarse PD(s) para que el valor de PID permanezca igual a su valor previo antes de la conmutación.

Específicamente, el término PD para el algoritmo en serie es:

$$PD(s)_{SERIE} = [A(s) - B(s)] \cdot K_p \tag{1}$$

siendo $A(s) = SP(s) \cdot \frac{K_D s + 1}{\frac{K_D}{N} s + 1}$ si el término derivativo se basa en el error; si no,

$$A(s) = SP(s), \quad B(s) = PV(s) \cdot \frac{K_D s + 1}{\frac{K_D}{N} s + 1}$$

El término PD para el algoritmo en paralelo es:

$$PD(s)_{PARALELO} = [C(s) - E(s)] + [SP(s) - PV(s)] \cdot K_p \quad (2)$$

siendo $C(s) = SP(s) \cdot \frac{K_D s}{\frac{K_D}{N} s + 1}$ si el término derivativo se basa en el error; si no,

$$C(s) = SP(s), \quad E(s) = PV(s) \cdot \frac{K_D s}{\frac{K_D}{N} s + 1}$$

Distribuir K_p a los términos $SP(s)$ y $PV(s)$ en las ecuaciones anteriores produce

$$PD_{SERIE} = A(s) \cdot K_p - B(s) \cdot K_p \quad (3)$$

y:

$$PD_{PARALELO} = C(s) - E(s) + SP(s) \cdot K_p - PV(s) \cdot K_p \quad (4)$$

5 Igualar entre sí las Ecuaciones (3) y (4) produce:

$$A(s) \cdot K_p - B(s) \cdot K_p = C(s) - E(s) + SP(s) \cdot K_p - PV(s) \cdot K_p \quad (5)$$

La igualdad definida en la Ecuación (5) se mantiene para realizar una conmutación 100, o transferencia, sin perturbaciones. Por lo tanto, cuando se conmuta entre los algoritmos en paralelo 90 y en serie 80, se recalculan las variables $A(s)$ y $B(s)$ de estado tras la primera ejecución del algoritmo 80 en serie después de la conmutación usando:

$$A(s) = \frac{1}{K_p} C(s) + SP(s) \quad (6)$$

$$B(s) = \frac{1}{K_p} E(s) + PV(s) \quad (7)$$

10 Asimismo, a partir de la Ecuación (5), cuando se conmuta entre los algoritmos 80 en serie y en paralelo 90, se recalculan las variables $C(s)$ y $E(s)$ de estado tras la primera ejecución del algoritmo 90 en paralelo después de la conmutación usando:

$$C(s) = K_p [A(s) - SP(s)] \quad (8)$$

$$E(s) = K_p [B(s) - PV(s)] \quad (9)$$

En una implementación digital del bloque de funciones PID avanzadas con conmutación dinámica de algoritmos, las variables $C(s)$ y $A(s)$ de estado se representan con la misma variable $H_SP(s)$, haciendo la Ecuación (6):

$$H_SP(s) = \frac{1}{K_p} H_SP(s) + SP(s) \quad (10)$$

15 y la Ecuación (8):

$$H_SP(s) = K_p [H_SP(s) - SP(s)] \quad (11)$$

5 Tal representación facilita la minimización de las variables usadas en el algoritmo, y en la Ecuación (10), tras la detección de una conmutación de paralelo 90 a serie 80, H_SP(s) es la última C(s) calculada antes de la conmutación, y en la Ecuación (11) H_SP(s) es la última A(s) calculada antes de la conmutación de serie a paralelo; por ende, las Ecuaciones (10) y (11) siguen siendo válidas. Si el término derivativo se basa únicamente en la variable de proceso, entonces H_SP(s) es igual al valor de referencia en ambos tipos de algoritmo y no es necesario ajuste alguno.

Asimismo, las variables B(s) y E(s) de estado se representan con la misma variable H_PV(s), haciendo la Ecuación (7):

$$H_PV(s) = \frac{1}{K_p} H_PV(s) + PV(s) \quad (12)$$

y la Ecuación (9):

$$H_PV(s) = K_p [H_PV(s) - PV(s)] \quad (13)$$

10 En la Ecuación (12), tras la detección de una conmutación de paralelo 90 a serie 80, H_PV(s) es la última E(s) calculada antes de la conmutación, y en la Ecuación (13) H_PV(s) es la última B(s) calculada antes de la conmutación de serie 80 a paralelo 90; por ende, las Ecuaciones (12) y (13) siguen siendo válidas.

15 Tal como se ha explicado en lo que antecede, la conmutación sin perturbaciones se logra cuando se conmuta de paralelo 90 a serie 80 ajustando dinámicamente las variables H_SP(s) y H_PV(s) de estado usando las Ecuaciones (10) y (12), respectivamente, en la primera ejecución del algoritmo 80 en serie después de la conmutación. Es decir, para la primera determinación del valor PID usando el algoritmo 80 en serie subsiguiente a la conmutación, las variables H_SP(s) y H_PV(s) de estado se ajustan usando las Ecuaciones (10) y (12). Por supuesto, después de la conmutación sin perturbaciones, las variables de estado responderán inmediatamente a los cambios en la variable de proceso y al valor de referencia según determine el algoritmo en serie.

20 La conmutación sin perturbaciones se logra cuando se conmuta de serie 80 a paralelo 90 ajustando dinámicamente las variables H_SP(s) y H_PV(s) de estado usando las Ecuaciones (11) y (13), respectivamente, en la primera ejecución del algoritmo 90 en paralelo después de la conmutación. Es decir, para la primera determinación del valor PID usando el algoritmo 90 en paralelo subsiguiente a la conmutación, las variables H_SP(s) y H_PV(s) de estado se ajustan usando las Ecuaciones (11) y (13). De nuevo, después de la conmutación sin perturbaciones, las variables de estado responderán inmediatamente a los cambios en la variable de proceso y al valor de referencia según determine el algoritmo en paralelo.

30 Aunque en lo que antecede se describen ecuaciones específicas en conexión con la conmutación sin perturbaciones, se contempla que en realizaciones alternativas puedan usarse otras técnicas para evitar cambios en la salida de los algoritmos que pueden dar como resultado inestabilidad del sistema. Por ejemplo, pueden utilizarse algoritmos conocidos de aplanamiento y de cálculos de promedios en conexión con la realización de tal conmutación, en los que, en un periodo de tiempo predeterminado (por ejemplo, 5 - 10 segundos), los valores de las salidas generadas con el algoritmo actualmente seleccionado se promedian con un conjunto de salidas previamente generadas y definidas, de modo que la transición de la salida generada con el algoritmo inicial a la salida generada con el algoritmo actual no represente un cambio de función escalonada en las salidas generadas. El inconveniente de este procedimiento es que la respuesta de la salida a las perturbaciones reales del proceso pueden llegar a ser muy lentas durante este tiempo de transición, mientras que las ecuaciones presentadas en conexión con la conmutación "sin perturbaciones" se aplican únicamente en el momento en que ocurre la conmutación.

40 Además, se cree que implementar tales algoritmos y tal conmutación de control en soporte lógico o soporte lógico inalterable proporciona los beneficios añadidos de una complejidad reducida y un uso reducido de variables en comparación con la conmutación manual o la conmutación analógica.

45 Como se apreciará con base en la descripción anterior, las realizaciones descritas en lo que antecede pueden implementarse usando programación de ordenador y/o técnicas de diseño, incluyendo soporte lógico de ordenador, soporte lógico inalterable, soporte físico o cualquier combinación o subconjunto de los mismos, siendo el efecto técnico la conmutación entre diferentes tipos de algoritmos de control (en serie y en paralelo). Cualquier programa resultante de ese tipo, que tenga código legible por ordenador, puede ser implementado o proporcionado en uno o más medios legibles por ordenador, creando con ello un producto de programa de ordenador, es decir, un artículo fabricado, según las realizaciones descritas en lo que antecede. El medio legible por ordenador puede ser, por ejemplo, sin limitación, una unidad de disco fijo (duro), un disquete, disco óptico, cinta magnética, memoria de semiconductores tal como memoria de solo lectura (ROM) y/o cualquier medio transmisor/receptor tal como Internet

u otro enlace o red de comunicaciones. El artículo fabricado que contiene el código de ordenador puede ser realizado y/o usado ejecutando el código directamente de un medio, copiando el código de un medio a otro medio o transmitiendo el código a una red.

5 Aunque la divulgación ha sido descrita en términos de diversas realizaciones específicas, se reconocerá que la divulgación puede ser puesta en práctica con modificación dentro del alcance de las reivindicaciones.

REIVINDICACIONES

1. Un procedimiento para conmutar de un primer algoritmo de control a un segundo algoritmo de control en un sistema de control continuo de procesos, comprendiendo dicho procedimiento:

5 ejecutar el primer algoritmo de control;
 determinar conmutar del primer algoritmo de control al segundo algoritmo de control;
 ejecutar el segundo algoritmo de control; y,
 para al menos una primera ejecución del segundo algoritmo de control, ajustar al menos una variable de estado usada en el segundo algoritmo de control, comprendiendo la al menos una variable de estado una variable de proceso o un valor de referencia de la variable de proceso.

- 10 2. Un procedimiento según la reivindicación 1 en el que el primer algoritmo de control es un algoritmo PID en paralelo y el segundo algoritmo de control es un algoritmo PID en serie.

3. Un procedimiento según la reivindicación 2 en el que las variables de estado $H_SP(s)$ y $H_PV(s)$ son ajustadas para al menos una primera ejecución del algoritmo PID en serie, siendo:

$$H_SP(s) = \frac{1}{K_p} H_SP(s) + SP(s),$$

y

$$H_PV(s) = \frac{1}{K_p} H_PV(s) + PV(s),$$

15 en las que:

$PV(s)$ es la variable de proceso que ha de controlarse,
 $SP(s)$ es el valor de referencia, es decir, un valor deseado de la variable de proceso, y
 K_p es la ganancia proporcional.

- 20 4. Un procedimiento según la reivindicación 1 en el que el primer algoritmo de control es un algoritmo PID en serie y el segundo algoritmo de control es un algoritmo PID en paralelo.

5. Un procedimiento según la reivindicación 4 en el que el que las variables de estado $H_SP(s)$ y $H_PV(s)$ son ajustadas para al menos una primera ejecución del algoritmo PID en paralelo, siendo:

$$H_SP(s) = K_p [H_SP(s) - SP(s)],$$

y

$$H_PV(s) = K_p [H_PV(s) - PV(s)],$$

en las que:

25 $PV(s)$ es la variable de proceso que ha de controlarse,
 $SP(s)$ es el valor de referencia, es decir, un valor deseado de la variable de proceso, y
 K_p es la ganancia proporcional.

- 30 6. Un controlador para un sistema de control continuo de procesos, comprendiendo dicho controlador un procesador acoplado a una memoria, estando programado dicho procesador para ejecutar algoritmos de control continuo de procesos para generar una salida PID, estando programado dicho controlador, además, para conmutar de un primer algoritmo de control a un segundo algoritmo de control y, para al menos una primera ejecución del segundo algoritmo de control, ajustar al menos una variable de estado usada en dicho segundo algoritmo de control, comprendiendo la al menos una variable de estado una variable de proceso o un valor de referencia de la variable de proceso.

- 35 7. Un controlador según la reivindicación 6 en el que el primer algoritmo de control es un algoritmo PID en paralelo y el segundo algoritmo de control es un algoritmo PID en serie.

8. Un controlador según la reivindicación 7 en el que las variables de estado $H_SP(s)$ y $H_PV(s)$ son ajustadas para al menos una primera ejecución del algoritmo PID en serie, siendo:

$$H_{-SP}(s) = \frac{1}{K_p} H_{-SP}(s) + SP(s),$$

y

$$H_{-PV}(s) = \frac{1}{K_p} H_{-PV}(s) + PV(s),$$

en las que:

PV(s) es la variable de proceso que ha de controlarse,
 SP(s) es el valor de referencia, es decir, un valor deseado de la variable de proceso, y
 K_p es la ganancia proporcional.

- 5
9. Un controlador según la reivindicación 6 en el que el primer algoritmo de control es un algoritmo PID en serie y el segundo algoritmo de control es un algoritmo PID en paralelo.
 10. Un controlador según la reivindicación 9 en el que el que las variables de estado H_{-SP}(s) y H_{-PV}(s) son ajustadas para al menos una primera ejecución del algoritmo PID en paralelo, siendo:

$$H_{-SP}(s) = K_p [H_{-SP}(s) - SP(s)],$$

10 y

$$H_{-PV}(s) = K_p [H_{-PV}(s) - PV(s)],$$

en las que:

PV(s) es la variable de proceso que ha de controlarse,
 SP(s) es el valor de referencia, es decir, un valor deseado de la variable de proceso, y
 K_p es la ganancia proporcional.

- 15
11. Un sistema (10) de control de procesos para controlar un proceso, comprendiendo dicho sistema:
 - una pluralidad de sensores (18) configurados para detectar parámetros del proceso que ha de controlarse;
 - una pluralidad de controladores locales (20) configurados para ajustar los parámetros del proceso; y
 - al menos un controlador central (12) según cualquiera de las reivindicaciones 6 a 10 acoplado a dichos sensores y a dichos controladores locales, estando configurado dicho controlador central para ejecutar
- 20 algoritmos de control continuo de procesos para determinar ajustes que ha de realizar dicho controlador local con base en datos de dichos sensores.

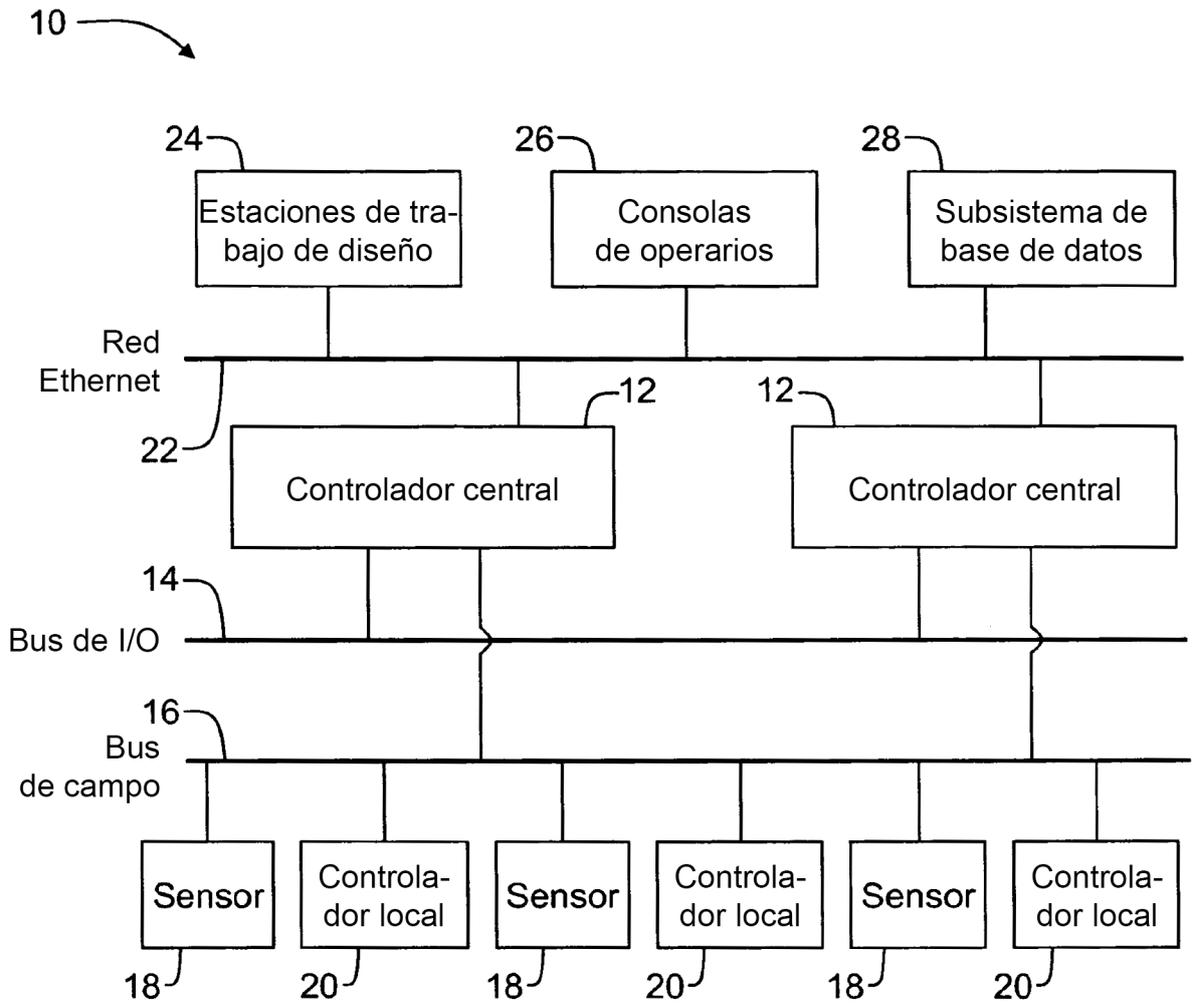


FIG. 1

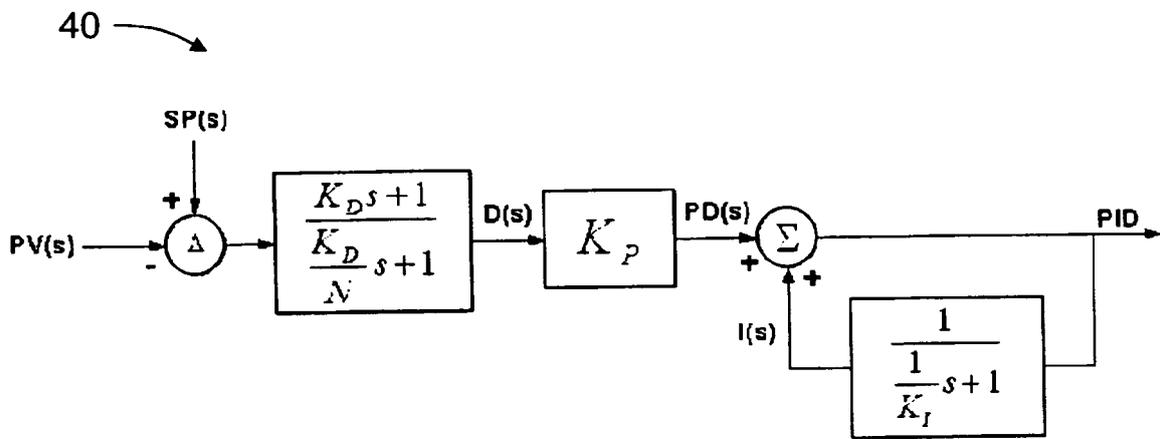


FIG. 2

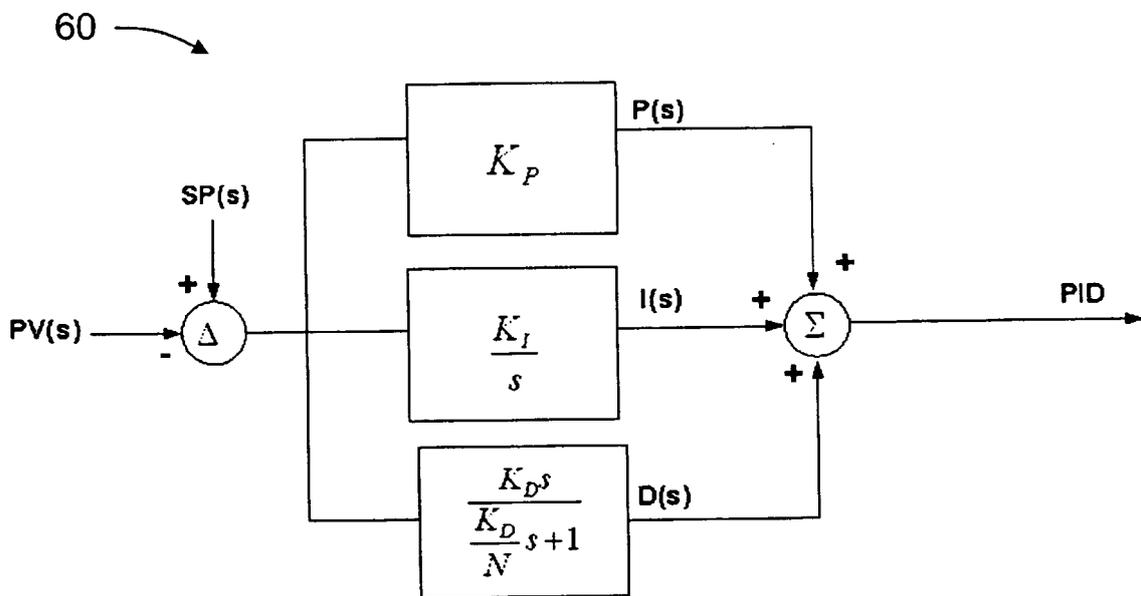


FIG. 3

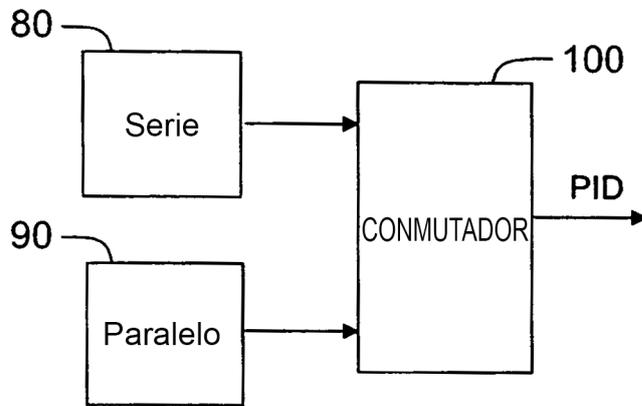


FIG. 4

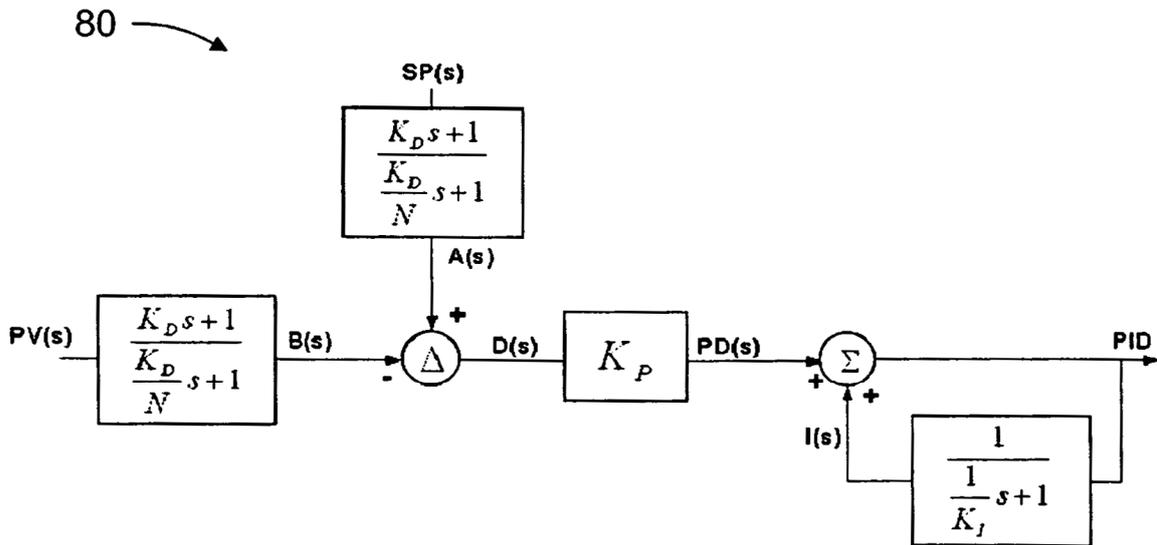


FIG. 5

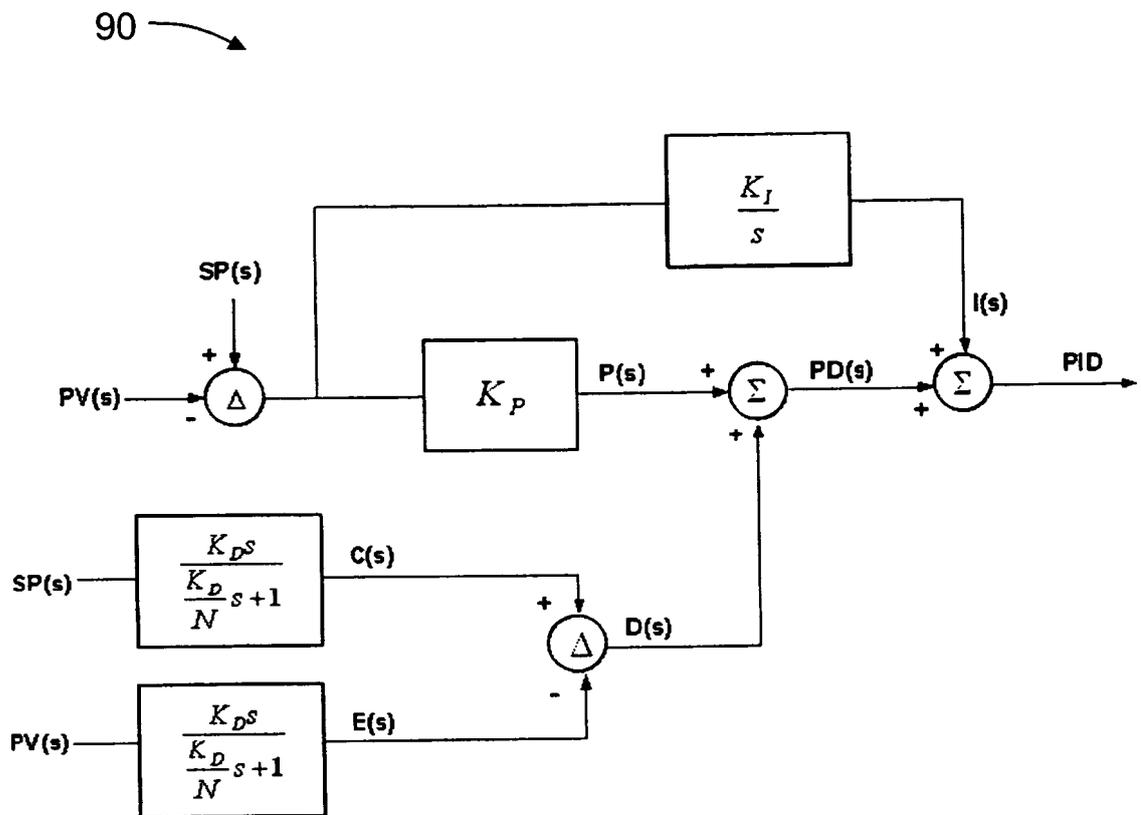


FIG. 6