

Diseño de Controlador PID Utilizando Labview

Design of PID Controller using Labview

Bruno Elio Vargas-Tamani¹

Facultad de Ingeniería Electrónica y Eléctrica, Universidad Nacional Mayor de San Marcos, Lima Perú

Resumen— Se presenta el diseño e implementación de un controlador Proporcional Integral Derivativo (PID) utilizando el lenguaje de programación LabView de National Instruments. Se muestra la secuencia de diseño del controlador PID de manera detallada y sus resultados para el control en lazo cerrado de una planta, la cual es conectada a una computadora a través de una tarjeta de adquisición de datos DAQ PCI 6024E de National Instruments. Los parámetros de control PID, así como el período de muestreo, pueden ser ingresados desde un panel de control, los resultados del desempeño se muestran gráficamente a través de datos en tiempo real que se grafican en el panel de control.

Abstract— It presents the design and implementation of a Proportional Integral Derivative controller (PID) using the programming language LabView from National Instruments. It shows the sequence of PID controller design in detail and results for the closed loop control of a plant, which is connected to a computer through a data acquisition board DAQ PCI 6024E from National Instruments. The PID control parameters, as well as the sampling period can be entered from a control panel, performance results are displayed graphically through real-time data that is plotted on the control panel.

Palabras Clave— Controlador PID, LabView, Panel de control, Diagrama de programación, Tiempo de muestreo.

Key Words— PID controller, control panel, diagram of programming, sampling time.

I. INTRODUCCIÓN

Los controladores de procesos Proporcional Integral Derivativo más conocidos con las siglas PID, si bien datan de hace mucho tiempo, siguen manteniéndose vigentes en las aplicaciones industriales y cada vez aparecen nuevas versiones que permiten mejorar su desempeño utilizándose técnicas de control actual. Por otro lado, a nivel industrial cada vez más es común utilizar interfaces hombre máquina,

que permiten al operador configurar los parámetros del controlador y/o supervisar un proceso mediante monitores, paneles, etc. que presentan en forma gráfica con indicadores numéricos y graficadores información relevante del proceso. El lenguaje de programación gráfico LabView de National Instruments es un programa que permite diseñar interfaces usuario que pueden fácilmente recibir información del proceso y presentarla ya sea gráficamente, numéricamente o de otros tipos, según convenga. Como es de esperar, también permite al usuario ingresar información que puede modificar el comportamiento del sistema; esta información puede ser procesada utilizando las diferentes funciones disponibles en LabView, para desarrollar algoritmos de control de procesos [1], [2].

El objetivo es lograr diseñar un controlador PID utilizando LabView, implementarlo y probar su desempeño en un sistema de lazo cerrado. Se debe diseñar una interface con el usuario que permita acceder al proceso a fin de optimizar su comportamiento. Se facilitará que el usuario pueda configurar los parámetros de ganancia proporcional, ganancia integral y ganancia derivativa del controlador PID, seleccionar el tiempo de muestreo para una adecuada aproximación digital del controlador analógico PID, elegir los canales analógicos de entrada y de salida, presentar resultados gráficos de la variación temporal de las señales de entrada, de salida y de control.

II. APROXIMACIÓN DIGITAL DE CONTROLADORES PID CONTINUOS

En un sistema de lazo cerrado para el control de una planta con realimentación unitaria, el controlador se conecta en serie con ella. Sea $e(t)$ la señal de error de un sistema de control de realimentación unitaria, esta señal es procesada para generar la señal de control $u(t)$ que se aplica al proceso de lazo cerrado.

La estrategia del controlador PID es obtener la señal de control $u(t)$ mediante la ecuación (1), en donde K_p es la ganancia proporcional, T_i es el tiempo integral y T_d es el tiempo derivativo.

¹Bruno Elio Vargas Tamani, e-mail: bvargast@unmsm.edu.pe.
Recepción: Marzo de 2009 / Aceptación: Abril de 2009

$$u(t) = K_p \left\{ e(t) + \frac{1}{T_i} \int_0^t e(t') dt' + T_d \frac{de(t)}{dt} \right\} \quad (1)$$

Se puede obtener un equivalente discreto del controlador PID para un tiempo de discretización T , utilizando una aproximación discreta tanto para la componente integral como para la componente derivativa que se describe mediante la ecuación (2), donde k se refiere al instante de muestreo actual, $(k-1)$ un instante de muestreo antes y $(k-2)$ dos instantes de muestreo antes [3]. Las constantes a , b y c se determinan en el proceso de diseño del controlador y están dadas por la expresión (3).

$$u(k) = u(k-1) + ae(k) + be(k-1) + ce(k-2) \quad (2)$$

$$a = K_p \left\{ 1 + \frac{T}{2T_i} + \frac{T_d}{T} \right\} \quad (3)$$

$$b = K_p \left\{ \frac{T}{2T_i} - 1 - \frac{2T_d}{T} \right\}$$

$$c = K_p \frac{T_d}{T}$$

El operador diseñará previamente el controlador PID analógico; de acuerdo al proceso por controlar, obtendrá el equivalente discreto para el periodo de muestreo elegido (T), calculará las constantes a , b y c ; con las cuales implementará el algoritmo que corresponde a la ecuación (2).

III. DISEÑO DE LA INTERFACE DE CONTROL PID CON LABVIEW

La versión de LabView que se ha utilizado es la 6.1, la cual permite comunicarse, a través de una tarjeta de adquisición de datos DAQ PCI 6024E, con un proceso analógico que constituye la planta que va ser controlada con una estrategia PID discreta.

A. Diseño del panel de control

En LabView los programas desarrollados contienen dos partes claramente diferenciadas. La primera es el panel de control que constituye la interfaz con el usuario, que se presenta cuando se abre el programa y en donde se puede ingresar datos de diferentes tipos que requiere el algoritmo programado o colocar presentadores de resultados en tiempo real, los cuales pueden ser numéricos o gráficos, entre otros.

La segunda parte, que trabaja en forma paralela al panel de control, es el diagrama de programación en donde se implementa los algoritmos del programa en base a funciones gráficas que contiene el LabView. Cada vez que se coloca algún control (nombre que da Labview a los objetos de entrada de datos) o algún indicador (nombre que da LabView a los objetos de salida de datos) en el panel de control, se genera

automáticamente su equivalente en el diagrama de programación, para poder allí realizar operaciones con los datos de entrada o conectar resultados a los indicadores.

En el panel de control mediante un indicador entero se registra el número asignado de la DAQ instalada que va a utilizarse para la lectura de los datos analógicos, y mediante otro indicador entero se registra el número asignado de la DAQ que va a entregar la señal analógica del algoritmo de control.

Para seleccionar los canales analógicos, tanto de entrada como de salida, se requieren dos indicadores del tipo I/O y dentro de ellos los del tipo DAQ Channel Name. Estos cuatro controles se muestran en la Fig. 1.



Fig. 1. Panel de control con cuatro controles iniciales.

Para la configuración de los parámetros del controlador PID discreto, el usuario debe ingresar por el panel de control tres controladores numéricos reales para las constantes a , b y c de acuerdo con la ecuación (2). También, a través de dos controles numéricos reales se indican los límites de salida analógica tanto positiva como negativa para la DAQ; esta situación previene el caso de que la operación de integración exceda valores más allá de los que pueda manejar la DAQ. Conviene agrupar a estos cinco indicadores numéricos para, luego, ser mejor tratados en el programa de control; para ese fin, se les agrupa mediante un cluster, este es un tipo de datos definido en LabView para agrupar datos de diferentes tipos. El panel de control y el diagrama de programación, ahora adopta la forma mostrada en la Fig. 2.

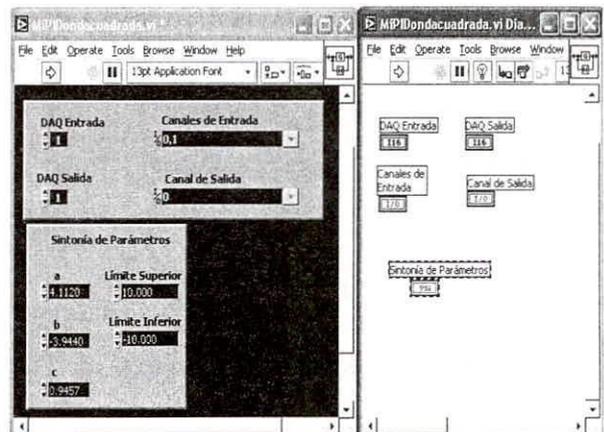


Fig. 2 Panel de control y diagrama de programación con cinco indicadores agregados.

El tiempo de muestreo que se calcula se debe ingresar mediante otro controlador numérico real. Por otro lado, para que el usuario pueda dar fin al programa se utiliza un botón del tipo booleano, específicamente del tipo Labeled Oblong Button. El nuevo panel de control y el diagrama de programación, se muestran en la Fig. 3.

El usuario necesita observar en tiempo real el valor de la señal de referencia o setpoint, también la variación temporal de la variable del proceso o variable controlada, así como la señal de control que genera el controlador PID. Para ello se utiliza tres indicadores Graph del tipo Waveform Chart. El panel de control final se muestra en la Fig. 4.

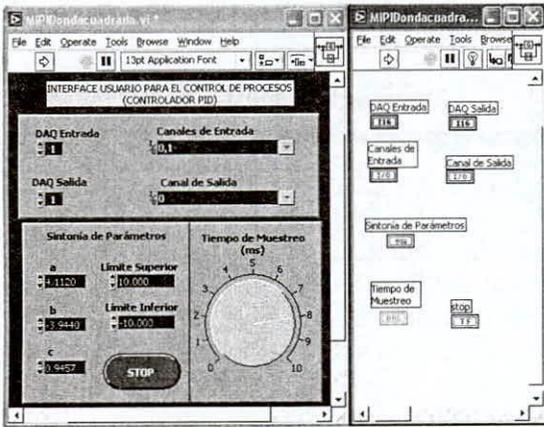


Fig. 3. Panel de control y diagrama de programación con control de muestreo y parada.

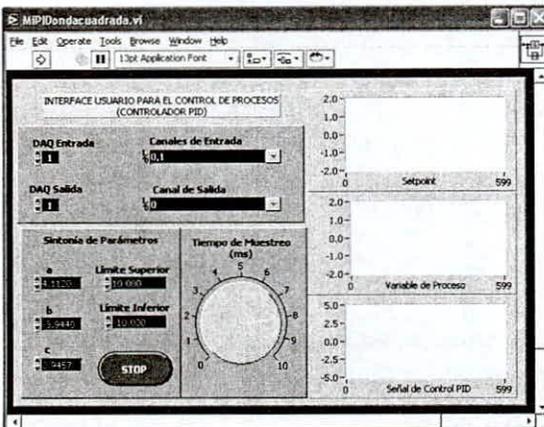


Fig. 4. Panel de control final con presentadores gráficos.

En la Fig. 5 se muestra la estructura del diagrama de programación hasta el momento; se observa los objetos equivalentes a los colocados en el panel del control.

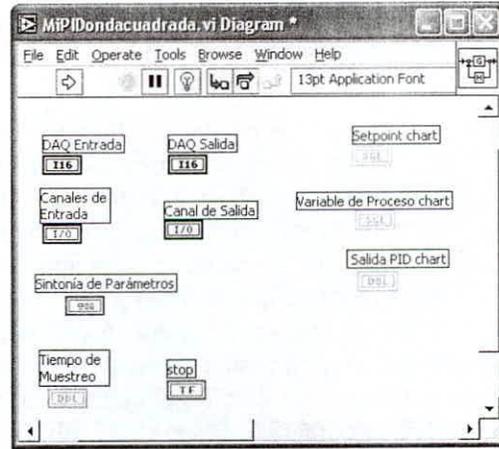


Fig. 5. Estructura del diagrama de programación.

B. Diseño del algoritmo de control

Luego de completado el diseño del panel de control, en el cual se presentan los controles (datos de entrada) y los controles (datos de salida), se pasa a diseñar el algoritmo que procesa los datos de entrada y entrega los de salida. Esta etapa se realiza completamente en el diagrama de programación.

Se necesita que el algoritmo se ejecute cada período de muestreo que debe ser ingresado en el control respectivo, la temporización se realiza mediante la función Wait Until Next ms Multiple, la cual se ubica en Time&Dialog que temporiza la cantidad de milisegundos que se ingresó. En la Fig. 6 se muestra el diagrama de programación que incluye la temporización.

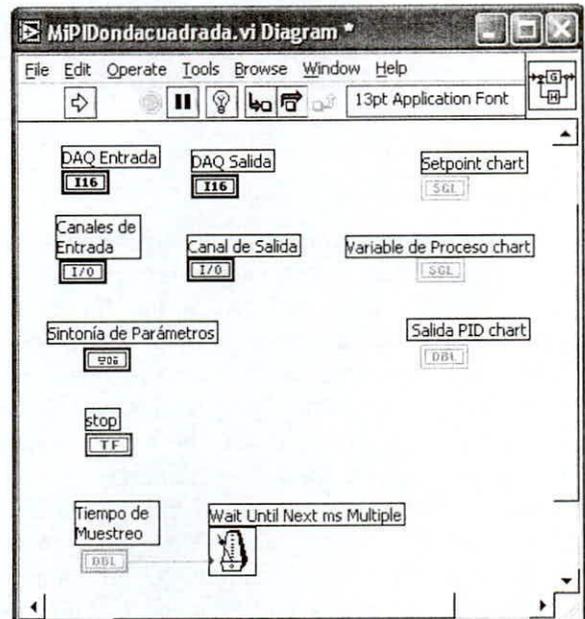


Fig. 6. Diagrama de programación del controlador PID, incluye temporización.

El algoritmo de control que se ejecuta repetidamente se programa mediante la función While Loop. Se selecciona esa función y en su interior se programan las operaciones repetitivas, se finaliza el lazo cuando se presiona el botón de parada.

Dentro del While Loop, se debe programar la lectura y escritura de datos analógicos. La lectura se realiza con la función ubicada en Data Acquisition/Analog Input/Analog Input Utilities/AI Read One Scan.vi, que permite leer datos de los canales indicados del panel de control. Los canales seleccionados se conectan a la función de lectura mediante la función Built Array que agrupa los canales en un arreglo. La escritura, que es la señal de control que calcula el algoritmo, se realiza con la función ubicada en Data Acquisition/Analog Output/Analog Output Utilities/AO Write One Update.vi, mediante ésta se permite enviar datos a los canales indicados en el panel de control. También, se utiliza otra función Built Array para conectar los canales seleccionados a la función de escritura.

En la Fig. 7 se muestra el diagrama de programación, con la estructura repetitiva While Loop, con las funciones de lectura y escritura de datos y las funciones Built Array indicadas.

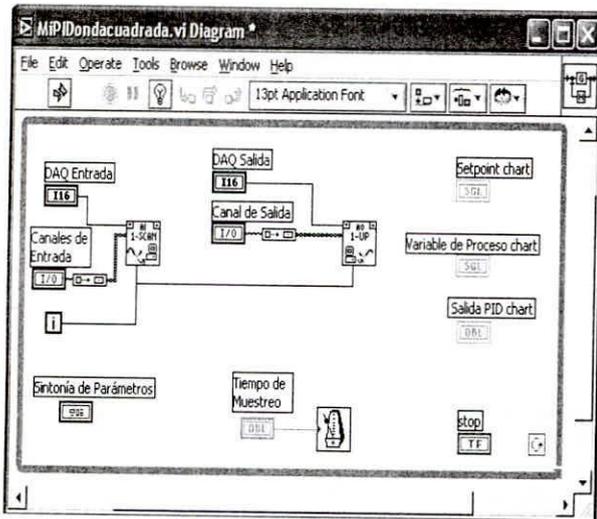


Fig. 7. Diagrama de programación incluye estructura de repetición.

Como se planteó en el diseño del panel de control, los parámetros del controlador PID discreto, agrupados en un cluster, y los límites de salida analógica para la DAQ, se utilizan para calcular la señal de control. Para implementar el controlador PID se utiliza un Subvi, en el cual se programan todas las funciones específicas para esa estrategia. Mediante la función de cluster Unbundle By Name, se separan los elementos del cluster que agrupa los parámetros del PID; así cada elemento del cluster se utiliza para programar el Subvi que corresponde al PID. La Fig. 8 muestra el Subvi en

el cual se programará el PID y las conexiones para su funcionamiento.

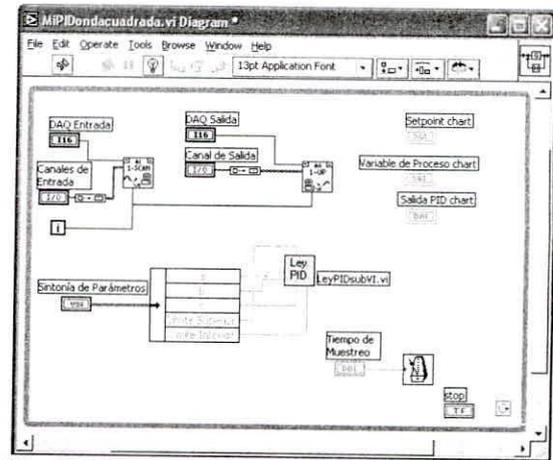


Fig. 8. Diagrama de programación con el Subvi.

El Subvi que genera la salida del PID, además de los parámetros del propio PID, debe recibir el valor del Setpoint, así como la señal de salida realimentada. Para realizar las conexiones necesarias al Subvi se utiliza la función Index Array, que separan los datos leídos por la función AI Read One Scan.vi. La salida del Subvi PID debe enviarse a la DAQ de salida, se necesita la función Build Array a partir de la salida del Subvi. La Fig. 9 muestra las conexiones del Subvi a las funciones de lectura y escritura de datos analógicos.

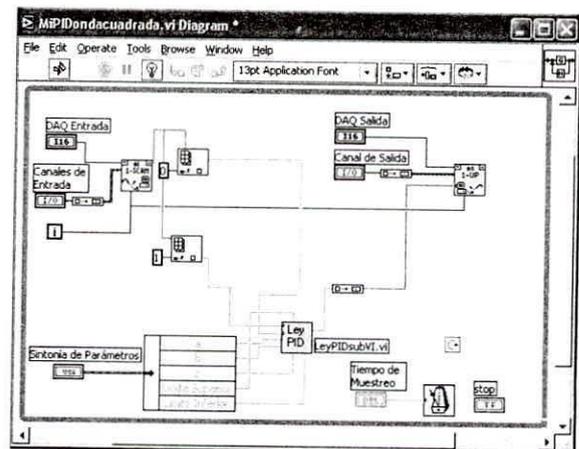


Fig. 9. Diagrama de programación con conexiones del Subvi.

Finalmente, los datos de entrada que corresponden al setpoint y a la señal realimentada de la variable del proceso deben ser vistos en los graficadores correspondientes; se realizan las conexiones en el diagrama a los indicadores del tipo Waveform Chart. Asimismo, para observar la señal de control producida por el PID, se realiza la conexión de la salida del PID al indicador Waveform Chart correspondiente. Se agregan al diagrama de programación los indicadores

de estatus tanto de lectura como de escritura de datos, que indicaran un error cuando cualquiera de ellos falle. La Fig. 10 muestra el diagrama de programación completo, donde el Subvi en el que se programará el PID está indicado como una sola función que incluirá otras funciones.

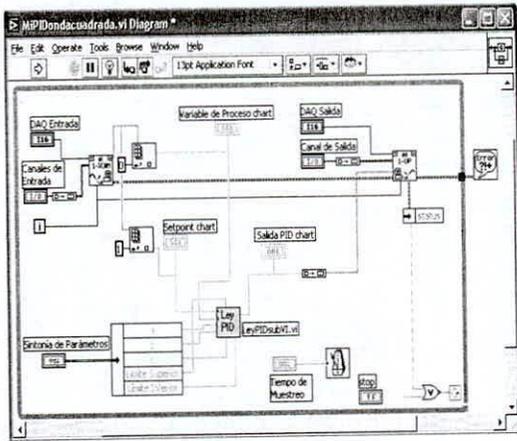


Fig. 10. Diagrama de programación completo.

C. Diseño del Subvi para el cálculo de la salida PID

La función del Subvi cuyo icono se indica como Ley PID en la Fig. 10, se encarga de calcular la ecuación en diferencias (ecuación 2) y enviar el resultado al proceso. Este Subvi se crea en otro archivo de LabView y se accede a él a través del programa principal.

A fin de implementar el algoritmo, se divide todo el programa en tres pasos secuenciales, para ello se utilizará una estructura llamada Sequence, que realiza operaciones secuenciales una detrás de otra. La estructura While Loop, dentro de la cual se programa una secuencia de tres pasos, se ejecutará sólo una vez en cada período de muestreo; eso se asegura mediante el control de la condición con una constante booleana del tipo falso. En el lazo, dentro de la estructura Sequence, la primera operación a programar es el cálculo de la señal de error actual tal como se muestra en la Fig. 11.

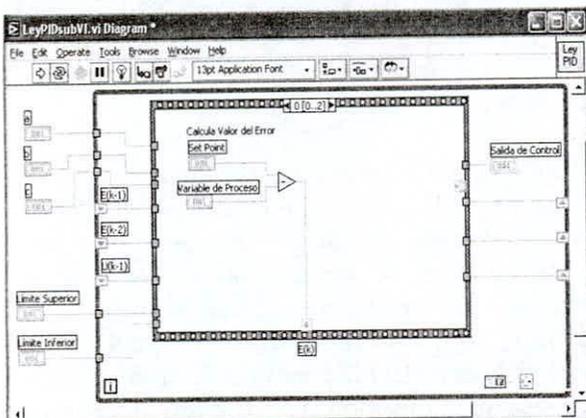


Fig. 11. Cálculo de la señal de error actual $u(k)$.

El siguiente paso es evaluar la ecuación 2, a partir de los datos conectados al Subvi. Aquí es donde se debe evitar el incremento progresivo de la función de integración, de tal manera que la señal de control exceda los límites de la DAQ, de ocurrir la salida analógica debe saturarse al valor límite positivo o negativo según sea el caso.

Para programar los casos de saturación se utiliza una estructura Case para dar el límite positivo en un caso y el negativo en otro. A la estructura Case ingresa el valor calculado por la ecuación en diferencias y allí se determina si la salida debe saturarse o no. Luego la salida del controlador se envía como salida del controlador PID a la función de salida de la DAQ y al presentador gráfico de salida. La Fig. 12 muestra el cálculo de la ecuación en diferencias y el caso de evaluación de saturación positiva.

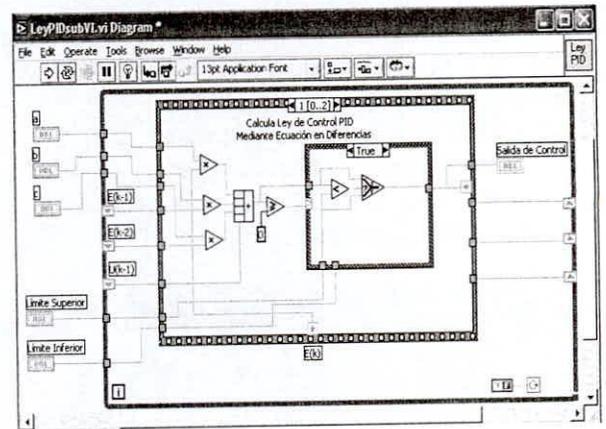


Fig. 12. Cálculo de la ecuación diferencias y evaluación saturación positiva.

La Fig. 13 muestra el cálculo de la ecuación en diferencias y el caso de evaluación de saturación negativa.

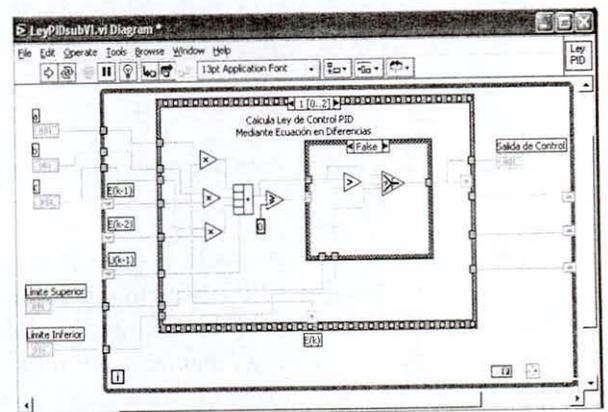


Fig. 13. Cálculo de la ecuación en diferencias y evaluación de saturación negativa.

El tercer paso, luego de calcular la salida del PID, consiste en actualizar los registros que corresponden a $e(k-1)$ y $e(k-2)$ con los valores anteriores de $e(k)$ y $e(k-1)$ respectivamente. La Fig. 14 muestra la actualización de esos registros.

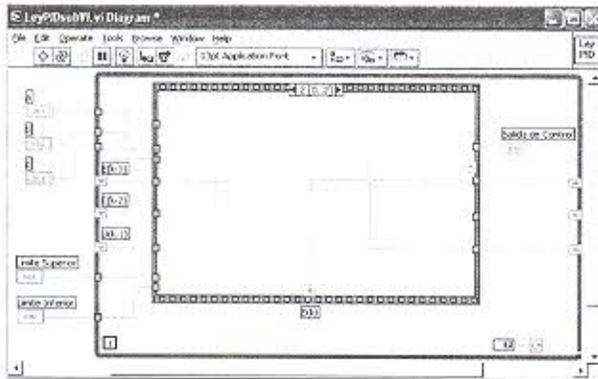


Fig. 14. Actualización de los registros $e(k-1)$ y $e(k-2)$.

IV. UTILIZACIÓN DEL CONTROLADOR PID

Se utilizó una planta con fines de ser controlada, para la cual se diseñó un controlador PID analógico, luego se eligió un período de muestreo T adecuado y se hallaron las constantes a , b y c de la ecuación en diferencias (ecuación 2). Se implementó el sistema de lazo cerrado, controlando a la planta mediante el controlador PID diseñado en LabView. Se realizaron las conexiones de la planta a la tarjeta DAQ PCI 6024E disponible, la cual por defecto se numera como uno. Se utilizan los canales cero y uno para la lectura analógica y el canal cero para la escritura analógica.

Como señal de referencia de lazo cerrado se utilizó una cuadrada que prueba la respuesta con dos niveles diferentes de referencia. En el programa de LabView de control PID se configuran las constantes necesarias como son el período de muestreo y las constantes a , b y c . El desempeño del sistema de lazo cerrado se puede comprobar en los graficadores de las señales de setpoint, variable del proceso y señal de control.

La Fig. 15 muestra el desempeño del sistema de lazo cerrado utilizando el controlador PID. Aquí se aprecia que la señal controlada sigue a la señal de referencia, con una respuesta transitoria subamortiguada y sin error de estado estacionario. La señal de control no excede los límites especificados de +10V y -10V. La señal de control llega transitoriamente a esos límites en los cambios de la señal de referencia.

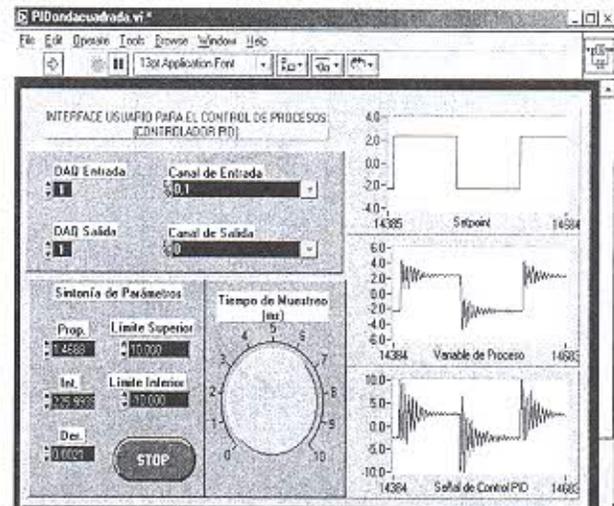


Fig. 15. Desempeño del controlador PID.

V. CONCLUSIONES

El lenguaje de programación gráfica LabView, permite crear interfaces con el usuario muy amigables, en razón de la múltiples librerías de funciones de entrada y de salida de datos, como de operaciones matemáticas que ofrece.

El controlador PID diseñado responde de acuerdo a lo esperado y permite con mucha facilidad cambiar los parámetros de control así como el tiempo de muestreo.

Se puede utilizar diferentes métodos para diseñar un controlador PID discreto y adaptar su respectiva ecuación en diferencias a la ecuación 2, con lo cual se posibilitaría implementarla utilizando la interfaz diseñada.

REFERENCIAS

- [1] Manuel Antoni, Biel Domingo, Olivé Joaquín, Prat Jordi, Sánchez Francesc. Instrumentación virtual. Alfaomega 2002.
- [2] Manuel Antoni. LabView. Programación gráfica para el control de instrumentación. ITP Paraninfo 1997.
- [3] Katsuhiko Ogata. Sistemas de control en tiempo discreto. Prentice Hall Hispanoamericana S.A. 1996.