

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica

Creación de un objeto ActiveX que determine la función de transferencia de una planta y proponga los parámetros de sintonización PID a partir de la respuesta al escalón de la planta

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura

Andrea Montenegro Guillén

Cartago Agosto, 2005

Hoja del tribunal evaluador

INSTITUTO TECNOLÓGICO DE COSTA RICA
ESCUELA DE INGENIERIA ELECTRONICA
PROYECTO DE GRADUACIÓN
TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

Firma


Ing. Sergio Morales Hernández

Profesor lector

Firma


Ing. Néstor Hernández Hostaller

Profesor lector

Firma


Ing. Eduardo Interiano Salguero

Profesor asesor



Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, 5 de agosto de 2005

Declaración de autenticidad

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 11 de agosto de 2005

Andrea Montenegro Guillén

Céd: 1-1061-0924

Resumen

Existe en la industria la necesidad de que los procesos productivos se realicen en forma automática. Ahí es donde entra la labor del ingeniero, quién busca la mejor solución para cada caso que se presenta.

Realizar estos trabajos sin contar con las herramientas adecuadas, hace que el tiempo requerido para desarrollarlo sea mayor y que el estudio del sistema a controlar sea escaso o inclusive nulo.

Por lo anterior es que en GNE se quiere disponer de la herramienta que mejor se adapte a los trabajos que allí se realizan, siendo una de sus actividades principales la automatización de procesos industriales.

La solución implementada consiste en un objeto ActiveX, que lleva el nombre de PIDX, que calcula las constantes de sintonización PID de la planta en estudio. Este programa también encuentra el modelo del sistema, se puede especificar si se quiere hacer una aproximación a un modelo de primer orden o de segundo orden, esto le sirve al programa para hacer una sintonización adecuada.

Como parte de la solución, el PIDX crea un archivo con los datos leídos de la entrada y la salida de la planta, cuando se le aplica a ésta un estímulo de tipo escalón. Esto último junto con el modelo encontrado sirve para analizar el comportamiento del proceso a controlar.

El PIDX se usa en simulaciones creadas en los programas InTocuh e InControl, en donde se simula el comportamiento real de la planta y de donde el programa implementado obtiene los datos que necesita.

Palabras claves: Automatización; Objeto ActiveX; PID; Respuesta al Escalón; Modelo del Sistema.

Abstract

In the industry, exist the necessity that the productive processes are made in automatic form. There it is where the work of the engineer enters, who looks for the best solution for each case that appears.

To make these works without telling on the suitable tools, causes that the required time to develop it is greater and than the study of the system to control becomes scarce or including null.

By the previous thing it is that in GNE it is wanted to have the tool that better adapts to the works that are made there, being the one of its main activities automatization of industrial processes.

The implemented solution consists of a ActiveX object, that takes the name of PIDX, that calculate the constants of tuning PID of the plant in study. This program also finds the model of the system, can be specified if it is wanted to make an approach to a model of first order or second order, this serves the program to him to make a tuning suitable.

Like part of the solution, the PIDX creates a file with the readed data of the entrance and the exit of the system, when a step type stimulus is applied. The last, along with found model serves to analyze the behavior of the process to control.

The PIDX is used in simulations created in the programs InTouch and InControl, in where the real behavior of the system is simulated and from where the implemented program collects the data that need.

Key words: Automatization; ActiveX Object; PID; Answer to the Step; Model of the System.

Dedicatoria

A mi mamá por su sacrificio y amor, por enseñarme a vivir y a luchar.

A mis hermanos Diego, Mariana y Sebastián porque cada uno de ellos me ha dado la fortaleza para seguir adelante.

A mi papá por ser mi ejemplo.

A Tony porque siempre ha creído en mí.

A Ileana porque siempre me ha tendido su mano.

A mis abuelos, por todo el cariño y apoyo.

Gracias a Leo, Luis Carlos y Sol.

A todos los que han estado conmigo, especialmente a los que me han ayudado a terminar y hoy están a mi lado.

ÍNDICE GENERAL

Capítulo 1: Introducción	1
1.1 Problema existente e importancia de su solución	1
1.2 Solución seleccionada	3
Capítulo 2: Meta y objetivos	6
2.1 Meta	6
2.2 Objetivo general	6
2.3 Objetivos específicos	6
2.3.1 Objetivos de software	6
2.3.2 Objetivos de implementación	7
Capítulo 3: Marco teórico	8
3.1 Descripción del sistema o proceso a mejorar	8
3.2 Antecedentes bibliográficos	9
3.3 Descripción de los principales principios físicos y/o electrónicos relacionados con la solución del problema	10
Capítulo 4: Procedimiento metodológico	29
4.1 Reconocimiento y definición del problema	29
4.2 Obtención y análisis de información	30
4.3 Evaluación de las alternativas y síntesis de una solución	31
4.4 Implementación de la solución	32
4.5 Reevaluación y rediseño	34
Capítulo 5: Descripción detallada de la solución	36
5.1 Análisis de soluciones y solución final	36
5.1.1 Simulación	36

5.1.2	Modelo de la planta	37
5.1.3	Sintonización PID	44
5.2	Descripción del software	50
Capítulo 6: Análisis de Resultados		52
Capítulo 7: Conclusiones y recomendaciones		73
7.1	Conclusiones	73
7.2	Recomendaciones	74
Bibliografía		76
Apéndices		78
A.1	Glosario, abreviaturas y simbología	78
A.2	Información sobre la empresa	79
B.1	Resultados parciales	81

ÍNDICE DE FIGURAS

Figura 1.1	Diagrama de bloques de un proceso industrial controlado con un PID	2
Figura 1.2	Entradas y salidas del PIDX	4
Figura 1.3	Flujo de información entre programas	5
Figura 3.1	Diagrama de flujo del proceso actual	8
Figura 3.2	Respuesta al escalón unitario	11
Figura 3.3	Valores usados para el cálculo del modelo de la planta	12
Figura 3.4	Respuesta al escalón para diferentes valores de ζ	16
Figura 3.5	Curva de respuesta al escalón de plantas altamente subamortiguadas	17
Figura 3.6	Gráfica normalizada η vs λ	20
Figura 3.7	Respuesta al escalón de una planta de segundo orden subamortiguada	21
Figura 3.8	Gráfica normalizada ξ vs λ	22
Figura 3.9	Ganancia de Fertik para un control PID	26
Figura 3.10	Tiempo integral de Fertik para un control PID	27
Figura 3.11	Tiempo derivativo de Fertik para un control PID	27
Figura 3.12	Ganancia de Fertik para un control PI	28
Figura 3.13	Tiempo integral de Fertik para un control	28
Figura 4.1	Secuencia de pasos para la implementación de la solución	32
Figura 5.1	División del algoritmo en primer y segundo orden para calcular el modelo de la planta	38
Figura 5.2	Diagrama de flujo de la solución para la sintonización PID	48
Figura 5.3	Diagrama de flujo de la solución final	49
Figura 6.1	Ventana principal de la simulación en InTocuh	53

Figura 6.2	Hoja de propiedades del objeto Lead Lag	53
Figura 6.3	Hoja de propiedades del objeto PID Control	54
Figura 6.4	Ventana gráfico histórico	55
Figura 6.5	Ventana gráfica de temperatura	55
Figura 6.6	Ventana gráfica de nivel	56
Figura 6.7	Ventana alarmas	56
Figura 6.8	Ventana de variables de InControl	57
Figura 6.9	Aspecto del PIDX a) sin habilitar b) habilitado	57
Figura 6.10	Páginas de propiedades del PIDX	58
Figura 6.11	Gráficas de respuesta al escalón de las plantas de primer orden a, b, c y d de la tabla 6.1	59
Figura 6.12	Gráficas de respuesta al escalón de las plantas de segundo orden subamortiguadas a, b, c y d de la tabla 6.2	61
Figura 6.13	Gráficas de respuesta al escalón de las plantas de segundo orden sobreamortiguadas a, b, c y d de la tabla 6.3	63
Figura 6.14	Gráficas de respuesta al escalón de las plantas de segundo orden críticamente amortiguadas a, b, c y d de la tabla 6.4	64
Figura 6.15	Diagrama de flujo del algoritmo utilizado para el cálculo de los parámetros del modelo de plantas de primer y segundo orden	66
Figura 6.16	Algoritmo para la sintonización PID y PI para modelos de orden uno	69
Figura 6.17	Algoritmo para la sintonización PID y PI para modelos de orden dos	70
Figura 6.18	Señal de ruido generada en el programa Matlab	70
Figura 6.19	Curvas de respuesta de las plantas a, b y c de la tabla 6.5 controladas por los valores de sintonización PID de la tabla 6.6	71

ÍNDICE DE TABLAS

Tabla 3.1	Fórmulas de sintonización de Ziegler-Nichols	24
Tabla 3.2	Fórmulas de sintonización de Cohen – Coon	24
Tabla 6.1	Valores teóricos y valores calculados por el PIDX para cuatro plantas de primer orden	58
Tabla 6.2	Valores teóricos y valores calculados por el PIDX para cuatro plantas de segundo orden subamortiguadas	60
Tabla 6.3	Valores teóricos y valores calculados por el PIDX para cuatro plantas de segundo orden sobreamortiguadas	62
Tabla 6.4	Valores teóricos y valores calculados por el PIDX para cuatro plantas de segundo orden críticamente amortiguadas	64
Tabla 6.5	Datos teóricos contra datos calculados por el PIDX para el modelo de la planta	67
Tabla 6.6	Datos teóricos contra datos calculados por el PIDX para la sintonización PID y PI de las plantas de la tabla	68
Tabla 6.7	Valor teórico y calculados de una planta de orden uno en presencia de ruido	72
Tabla 6.8	Valores de los parámetros de la planta, obtenidos mediante diferentes métodos	81
Tabla 6.9	Valores de los parámetros de la planta, obtenidos mediante diferentes métodos	81
Tabla 6.10	Valores de los parámetros de la planta, obtenidos mediante diferentes métodos	82

Capítulo 1: Introducción

1.1 Problema existente e importancia de su solución

En las diferentes industrias y empresas en las cuales se efectúan procesos productivos, es común que se busque controlar estos procesos de forma automática. Con el objetivo de suplir esta necesidad, han surgido diferentes empresas que se dedican a prestar estos servicios, entre las cuales se encuentra GNE, en la que se realizará este proyecto.

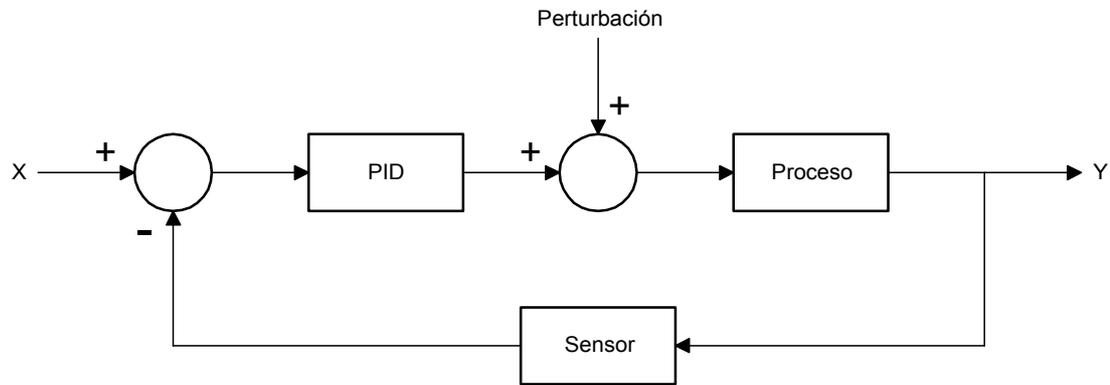
Uno de los algoritmos usados comúnmente para controlar procesos es el conocido como PID. Este controlador se encarga de disminuir o eliminar el error de estado estacionario y mejorar el tiempo de respuesta; logrando así un control efectivo de la variable deseada.

Para que el PID funcione correctamente se deben ajustar valores específicos de K, Td y Ti (constante proporcional, derivativa e integral respectivamente) para cada uno de los diferentes tipos de planta que se quieran controlar.

El cálculo de estos parámetros toma tiempo y requiere de medios para registrar la información. Además presenta el inconveniente de que una vez que se tiene una estimación, el ajuste final se hace por prueba y error, lo cual demanda más tiempo por parte del ingeniero y demora la finalización del trabajo.

Además de las horas profesionales invertidas, este procedimiento resulta en otros gastos innecesarios como lo es el transporte una y otra vez del equipo al sitio donde se encuentra el sistema. Asimismo necesita de un tiempo mayor para dar solución a cada proyecto y no se hace ningún estudio de la planta en caso de que en el futuro se requiera algún ajuste o mejora. En este caso se debe partir de cero, sin tener la posibilidad de aprovechar el trabajo desarrollado anteriormente.

En la figura 1.1 se muestra un diagrama de bloques del proceso de sintonización. Mediante una computadora se programa un PID en un PLC, el cual se conecta directamente al proceso por controlar.



VISIO32

Figura 1.1 Diagrama de bloques de un proceso industrial controlado con un PID

Se espera que al finalizar el proyecto los ingenieros de GNE reduzcan el tiempo de estudio y ejecución de la sintonización del PID. También, que los trabajos se dejen de realizar por prueba y error y que se apliquen los conocimientos de control automático. Como los trabajos desarrollados son para otras empresas, GNE se beneficiaría con una atención al cliente más eficiente, en un menor tiempo.

Algunos beneficios que se obtendrán al resolver el problema planteado anteriormente serían:

- a. Se contará con la función de transferencia de la planta estudiada.
- b. Se tendrá una simulación del comportamiento real de la planta en la computadora.
- c. Se tendrá una propuesta de los parámetros de sintonización PID adecuados de forma automática.
- d. Se tendrá la curva de respuesta del sistema ante una entrada de tipo escalón.
- e. El tiempo de duración de los proyectos se reducirá.

1.2 Solución seleccionada

El proyecto se plantea en la empresa GNE estipulando los siguientes requerimientos:

- a. El programa debe de encontrar el modelo de plantas de al menos segundo orden.
- b. El modelo debe hallarse a partir de la curva de respuesta de la planta a un estímulo tipo escalón.
- c. El programa debe estimar la sintonización PID para la planta que corresponde al modelo encontrado.
- d. El programa final debe de ser un objeto ActiveX.
- e. La programación del objeto debe de hacerse en Visual Basic o Visual C++.
- f. Las propiedades del objeto deben ser públicas, o sea, que se puedan acceder externamente.
- g. El programa debe trabajar en InTouch.

Como parte de la solución se realizó una aplicación en los programas InTouch e InControl, en la cual se simula el proceso de llenado de un tanque de agua y del control de la temperatura de éste.

El algoritmo utilizado para encontrar el modelo de primer orden lo expone Bollinger en [1] y para encontrar el modelo de segundo orden se utiliza el método propuesto por este mismo autor pero para sistemas de segundo orden, para este último caso el autor no plantea ninguna solución para encontrar el tiempo muerto de la planta por lo que se procede a usar el algoritmo planteado en [3].

Para estimar la sintonización PID se usa como base el método de Ziegler-Nichols [3] para sintonización a lazo abierto, a éste se le efectuaron dos modificaciones. La primera es que se escaló el valor de la constante derivativa resultante del método a 200 veces menor, esto para adaptarlo al

valor usado por el simulador. El segundo cambio es que el método no está hecho para plantas con respuesta rápida por lo que en estos casos la sintonización se halló empíricamente.

El programa también encuentra la sintonización para un controlador PI, se basa en el método antes mencionado y de igual forma hubo que adaptarlo para sistemas con respuesta rápida.

Cuando el objeto ActiveX está leyendo los datos de la prueba al escalón crea un archivo de texto y allí guarda los datos de la entrada, la salida y el tiempo.

En la figura 1.2 se pueden observar las entradas y salidas del programa, así como los diferentes cálculos que se realizan internamente.

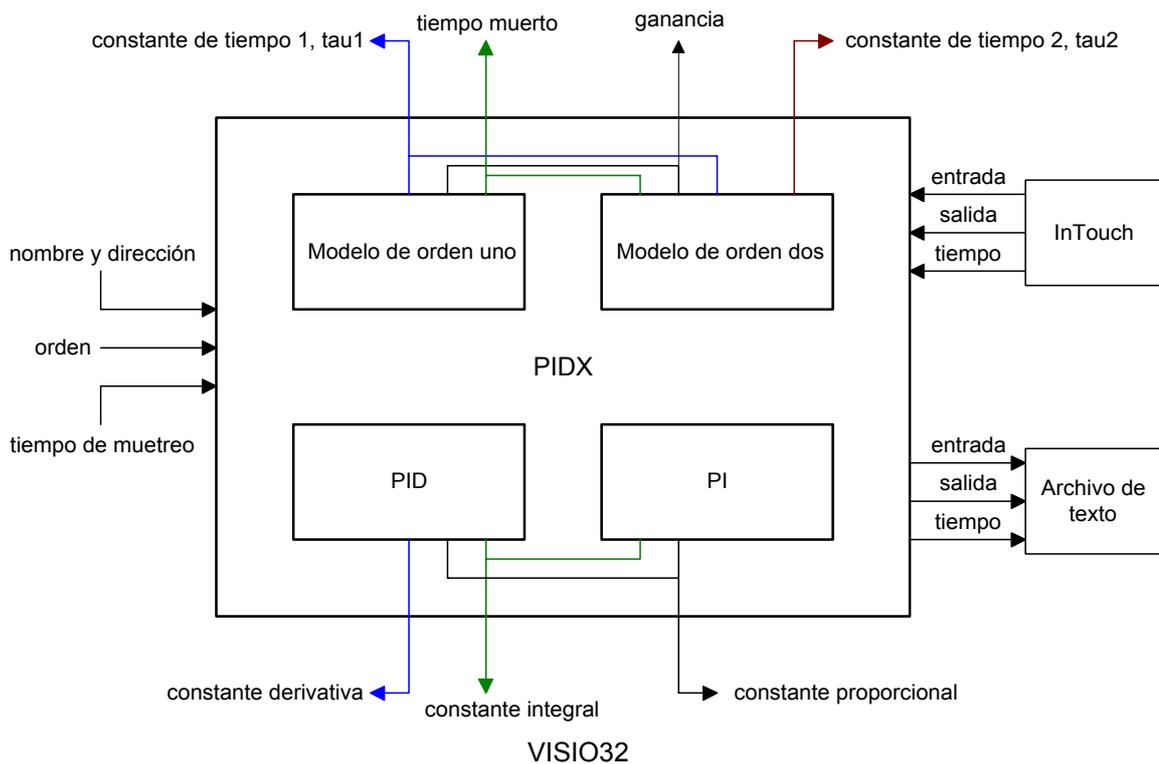
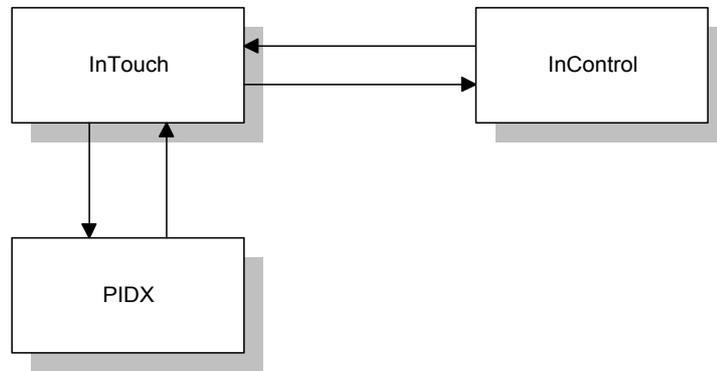


Figura 1.2 Entradas y salidas del PIDX

La forma de funcionamiento del programa es la introducción de este en simulaciones hechas en InTouch, como se muestra en la figura 1.3, y

con este mismo programa se puede tener acceso a las diferentes propiedades del PIDX como lo son los parámetros del modelo encontrado.



VISIO32

Figura 1.3 Flujo de información entre programas

Capítulo 2: Meta y objetivos

2.1 Meta

Disminuir el tiempo dedicado a determinar los parámetros de sintonización PID de los sistemas de control de las diferentes industrias a las que la empresa GNE brinda servicio.

2.2 Objetivo general

Crear un objeto ActiveX que determine la función de transferencia de una planta a partir de su respuesta al escalón y proponga los parámetros de sintonización PID para controlarla.

2.3 Objetivos específicos

2.3.1 Objetivos de software

1. Definir un algoritmo que genere la función de transferencia de una planta a partir de la respuesta al escalón de la misma.
2. Formular un algoritmo que proponga los parámetros de sintonización PID a partir de la función de transferencia de la planta.
3. Producir un software en Visual Basic capaz de determinar la función de transferencia de una planta a partir de su respuesta al escalón.
4. Elaborar un programa en Visual Basic que proponga una sintonización PID a partir de la función de transferencia de un sistema.

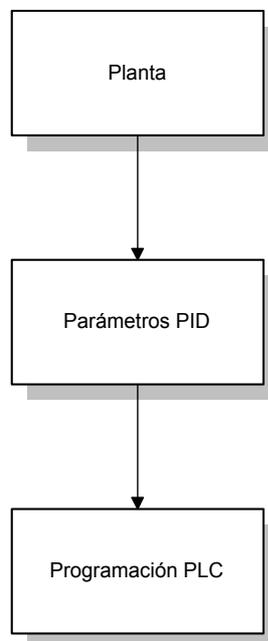
2.3.2 Objetivos de implementación

1. Realizar una aplicación demostrativa de un PLC utilizando el programa InControl.
2. Efectuar una simulación del comportamiento de una planta haciendo uso del programa InControl.
3. Diseñar una planta usando la aplicación InTouch para obtener la respuesta al escalón de ésta.
4. Crear un objeto tipo ActiveX a partir del programa desarrollado en Visual Basic.

Capítulo 3: Marco teórico

3.1 Descripción del sistema o proceso a mejorar

El proceso que se quiere mejorar consta de varias etapas: el reconocimiento de la planta, la obtención de los parámetros de sintonización PID, y por último, la programación del PID en un PLC. En la figura 3.1 se muestra el diagrama de flujo de dicho proceso.



VISIO32

Figura 3.1 Diagrama de flujo del proceso actual

Como primer paso se definen para la planta en estudio, cuáles son las variables que se quieren controlar; esto es específico de cada sistema.

Cuando esto se ha definido, se procede con la siguiente etapa, que es la obtención de los parámetros de sintonización PID de la planta. Esto se hace normalmente a prueba y error por parte del encargado, debido a que no se cuenta con una herramienta adecuada que facilite la obtención de los mismos de forma rápida y precisa. Para poder encontrar los valores

adecuados de sintonización, se debe de poner a trabajar la planta una y otra vez.

Una vez que se encuentran los valores de sintonización que controlan la planta de la forma que se requiere, se procede a hacer el programa final en un PLC, que luego se conectará a dicho sistema.

Este sistema se quiere mejorar excluyendo la parte de la programación final del PLC y la conexión de éste con el sistema; se limitará en esta parte a proponer los parámetros de sintonización adecuados, y se incorporaran otras etapas intermedias así como la mejora de las primeras dos etapas anteriormente explicadas.

3.2 Antecedentes bibliográficos

Se utilizarán libros de teoría de control automático, como lo es el libro Sistemas de Control Automático del autor Benjamín C. Kuo, también se usarán las notas del curso Análisis de Sistemas hechas por el Ing. José Alberto Díaz y las del curso Control Automático del profesor Ing. Eduardo Interiano.

En la empresa se cuenta con un libro en el que se explica la teoría de modelado de sistemas, así como diferentes métodos de sintonización PID. También se encontraron varios artículos en Internet relacionados con estos temas.

Se usarán los manuales de los programas computacionales InTouch e InControl. De igual forma se cuenta con notas técnicas del uso de Matlab con otros programas realizadas por el asesor de empresa Ing. Johnny Alvarado.

En Internet se encontraron algunas notas sobre el uso de Visual Basic y sobre la elaboración de objetos ActiveX, también se cuenta con un libro sobre la generación de este tipo de objetos.

3.3 Descripción de los principales principios físicos y/o electrónicos relacionados con la solución del problema

Sistemas físicos

Los sistemas de control se usan para controlar sistemas físicos de diferentes tipos como lo son los sistemas eléctricos, los sistemas mecánicos (traslacionales y rotacionales), los sistemas hidráulicos, los térmicos, entre otros; también se pueden encontrar combinaciones de éstos [2]. Se les puede clasificar de orden 1, orden 2, y estudiarlos según su comportamiento transitorio sin que sea relevante el tipo de sistema físico.

Respuesta al escalón

Se utiliza para obtener la respuesta transitoria de un sistema de control. De acuerdo a [8] y [6], de la respuesta al escalón de un sistema podemos identificar lo siguiente:

Sobreimpulso máximo (M_p):

Normalmente se da en valor porcentual y se define como:

$$M_p = y_{\text{máx}} - y_{\text{ss}} \quad (3.1)$$

éste se usa para medir la estabilidad relativa.

Tiempo de retardo (t_d):

Es el tiempo necesario para que se alcance el 50% del valor final de la respuesta al escalón.

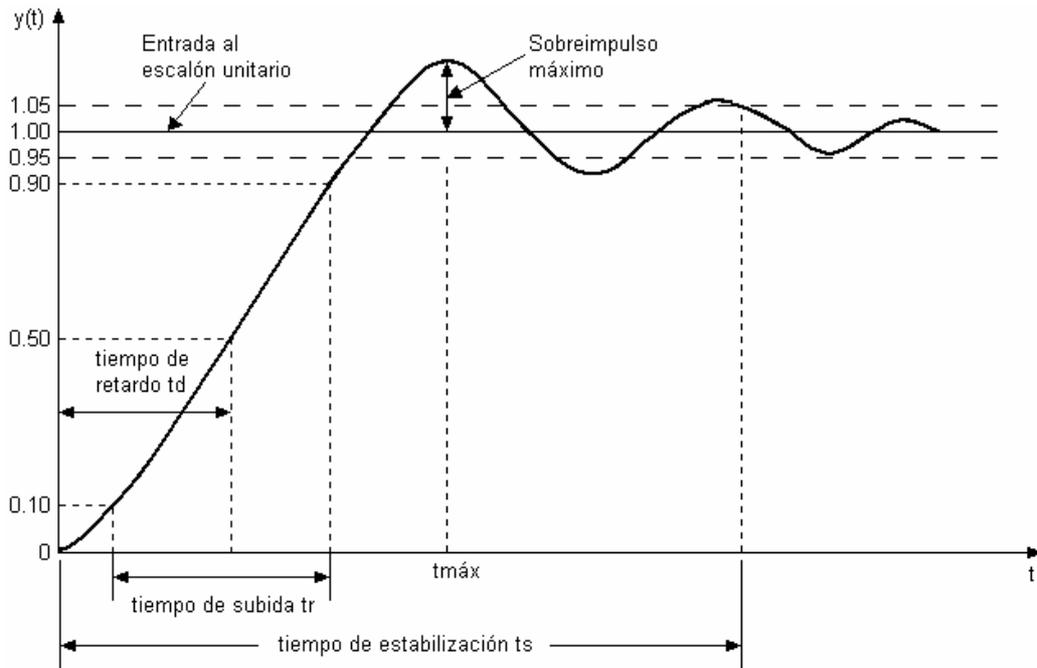
Tiempo de subida (t_r):

Tiempo que se tarda en pasar del 10 al 90 % del valor final.

Tiempo de estabilización (ts):

Es el tiempo que tarda el sistema en responder dentro de un porcentaje no mayor al 5 % de su valor final.

Se pueden observar estos parámetros en la figura 3.2.



PAINT

Figura 3.2 Respuesta al escalón unitario

Modelo de una planta de orden uno

Antes de entrar de lleno con los métodos de modelado es importante estudiar brevemente sobre el resultado que deseamos obtener con ellos, o sea, la función de transferencia de la planta en estudio.

La forma general de la función de transferencia de una planta de primer orden es la siguiente [13]:

$$G(s) = \frac{ke^{-t_d s}}{\tau s + 1} \quad (3.2)$$

En donde k es la ganancia de la planta, t_d es el tiempo que tarda la planta una vez que ha sido estimulada en reaccionar, llamado tiempo muerto de la planta y τ el cual es inversamente proporcional al valor del polo.

Es importante también saber que las pruebas hechas a la planta, en donde se obtiene la respuesta al escalón, deben realizarse a lazo abierto (13).

Existen diferentes métodos por los cuales se puede hallar el modelo de una planta; se hará referencia a cuatro de ellos que usan la respuesta al escalón de la planta para este propósito. La figura 3.3 muestra los valores que se deben encontrar para utilizar dichos métodos.

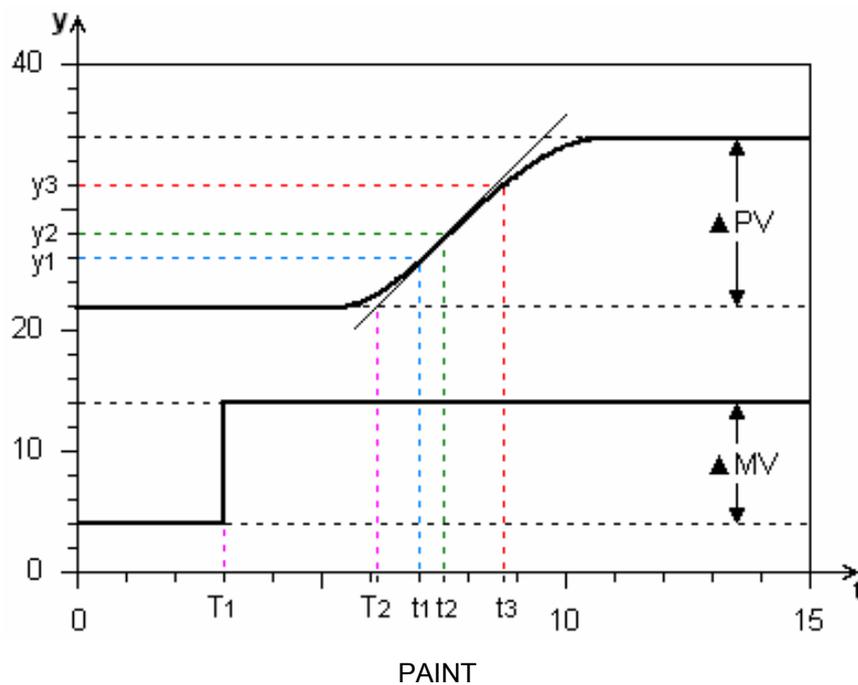


Figura 3.3 Valores usados para el cálculo del modelo de la planta

Primer método

Este método se basa en el método de Ziegler-Nichols [3]. La ganancia de la planta se calcula mediante la siguiente fórmula:

$$k = \frac{\Delta P}{\Delta M} = \frac{PV_2 - PV_1}{MV_2 - MV_1} \quad (3.3)$$

donde ΔP es la diferencia de magnitud, en la respuesta del sistema, entre el valor en estado estable (valor final) y el valor antes de presentarse el

estímulo (valor inicial); ΔM representa la misma variación anterior pero para la señal de entrada (escalón), como se muestra en la figura anterior.

El tiempo muerto del sistema, o sea, el tiempo que tarda el sistema en iniciar una respuesta al estímulo se calcula de la forma que sigue:

Primero se debe trazar una recta tangente al punto en que la curva de respuesta presenta mayor pendiente, y hacer que esta cruce por el eje correspondiente al valor inicial.

Obtenemos el tiempo correspondiente al lugar donde esta recta cruza dicho eje, el cual nombramos T_2 .

En [3], T_1 corresponde al momento en el cual se da el escalón; el tiempo muerto es la diferencia entre T_1 y T_2 .

$$t_d = T_2 - T_1 \quad (3.4)$$

Este retardo en el tiempo se puede observar en la figura 3.3 como la diferencia en el eje horizontal entre las líneas punteadas de color morado.

Por último, se debe calcular el valor de τ (tau), cuyo valor es inversamente proporcional al valor del polo que presenta la planta. Para esto hay que hacer los siguientes cálculos:

Primero se obtiene el tiempo en el cual la señal de salida alcanza el 63,2 % de ΔP y el 28,3 % de ΔP , t_3 y t_1 respectivamente. En la figura 3.3 se distinguen como las líneas de color rojo y azul.

Finalmente, el valor de τ se halla con la siguiente fórmula:

$$\tau = t_3 - t_1 \quad (3.5)$$

Segundo método

En este método la ganancia se calcula de la misma forma que en el método anterior, por lo que no se hará más referencia a ello.

Para calcular τ se utiliza la fórmula:

$$\tau = 1.5*(t_3 - t_2) \quad (3.6)$$

donde t_2 es el momento en que la salida alcanza un 39,3 % de ΔP .

Una vez que se tiene este valor, se puede proceder a calcular el tiempo muerto directamente de la siguiente fórmula [3]:

$$t_d = t_3 - \tau - T_1 \quad (3.7)$$

Los valores aquí utilizados se muestran en la figura 3.3.

Tercer método [1]

Este método se puede usar para encontrar la función de transferencia de la planta de la ecuación 3.2, así como la función de transferencia en tiempo discreto, la cual está definida por:

$$C_n = \delta C_{n-1} + k(1 - \delta)m_{n-1-d} \quad (3.8)$$

En este método se consigue el valor de la ganancia por medio de la siguiente división:

$$k = \frac{C_{ss}}{M} \quad (3.9)$$

donde C_{ss} corresponde al valor final de la salida y M corresponde al valor final de la entrada, que es el valor del escalón aplicado al sistema.

Para encontrar el valor de τ se debe hallar el tiempo en el cual la variable de proceso alcanza un 39,3 % de su valor final (y_2). Una vez hecho esto, tau se define como:

$$\tau = 2*(t_3 - t_2) \quad (3.10)$$

Cuando se tiene el valor de τ , se puede hallar el tiempo muerto restando a t_3 dicho valor, como sigue:

$$t_d = t_3 - \tau \quad (3.11)$$

Los valores de δ y d que aparecen en la ecuación (3.8) no se toman en cuenta ya que se quiere trabajar con procesos en tiempo continuo.

Cuarto método

En este último método, el valor de k se obtiene de la misma manera que en los métodos anteriores.

El tiempo muerto y tau [13] se adquieren al resolver el siguiente juego de ecuaciones:

$$t_1 = t_d + \tau/3 \quad (3.12)$$

$$t_3 = t_d + \tau \quad (3.13)$$

Los resultados obtenidos con este método son muy aproximados a los obtenidos mediante el segundo método.

En todos los métodos estudiados, es importante tomar en cuenta lo siguiente:

Si la muestra se inicia antes que la función escalón, este tiempo “sobrante” se debe restar al calcular los tiempos t_1 , t_2 y t_3 .

Si los valores iniciales de la variable de estado y del escalón son diferentes de cero, este valor debe sumarse a todos los cálculos porcentuales hechos sobre ΔP y ΔM .

Modelo de una planta de orden dos

Los sistemas de segundo orden se dividen en sistemas subamortiguados, críticamente amortiguados y sobreamortiguados. Los sistemas sobreamortiguados y críticamente amortiguados se pueden describir de forma general como sigue:

$$G(s) = \frac{ke^{-s\theta_D}}{(\tau_1s + 1)(\tau_2s + 1)} \quad (3.14)$$

y los sistemas subamortiguados con:

$$G(s) = \frac{ke^{-s\theta_D}}{(s/\omega_n)^2 + 2\xi(s/\omega_n) + 1} \quad (3.15)$$

Esta última es la forma general que describe a todas las plantas de segundo orden con tiempo muerto, el valor de ξ define de que tipo se trata. Si $\xi = 1$ entonces la planta esta críticamente amortiguada, si $\xi > 1$ el sistema esta subamortiguado y si $\xi < 1$ esta sobreamortiguado. En la figura 3.4 se muestra la curva de respuesta al escalón para diferentes valores de ζ .

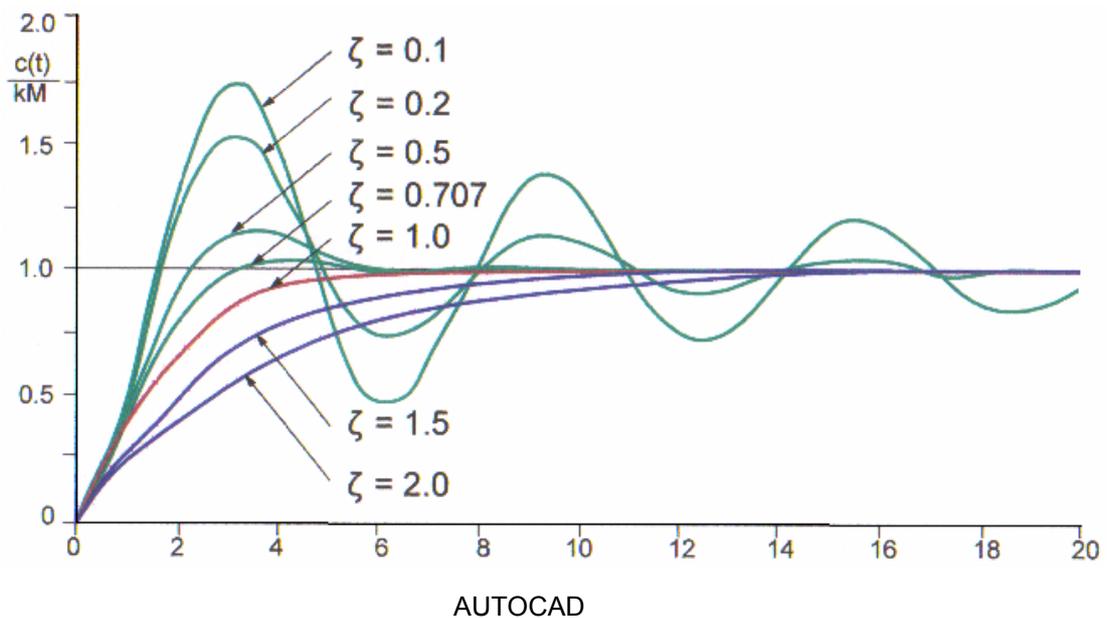


Figura 3.4 Respuesta al escalón para diferentes valores de ζ

Los sistemas subamortiguados presentan raíces complejas, mientras que los otros sistemas presentan raíces reales. Las constantes de tiempo de los sistemas críticamente amortiguados, sus raíces, son iguales, o sea; tienen el mismo valor.

Existen diferentes métodos para encontrar el modelo de una planta de segundo orden, aquí se van a explicar dos de ellos, ambos se basan en la curva de respuesta de la planta a un estímulo tipo escalón; son métodos gráficos.

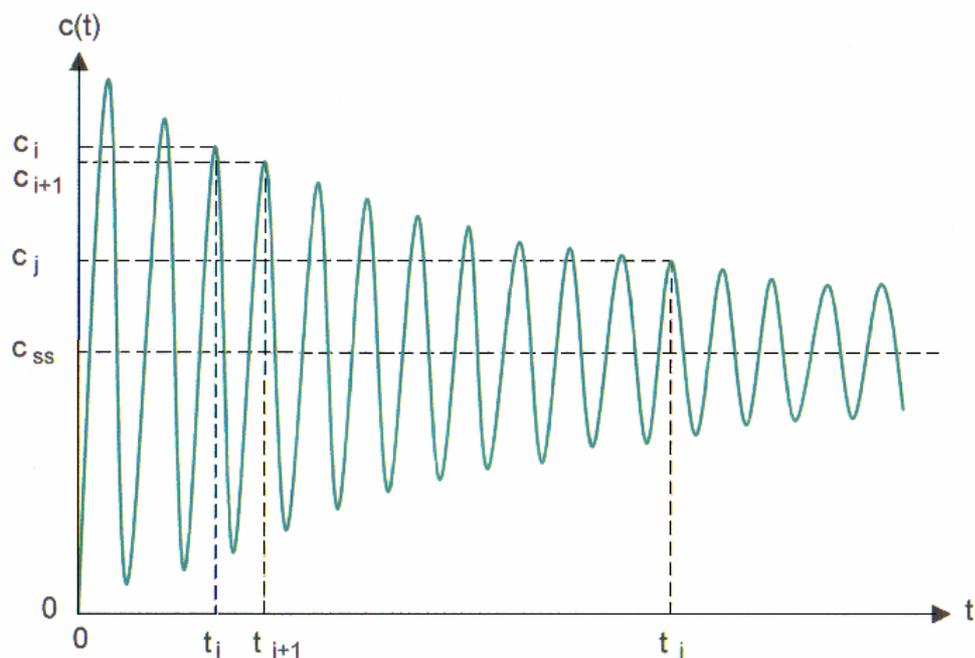
Primer método [1]

Se divide en tres algoritmos, uno para cada tipo de sistema, como se muestra a continuación.

Procesos subamortiguados:

El primer paso es encontrar el valor de ζ , para esto se deben trazar líneas tangentes a los picos de las oscilaciones que presenta la curva de respuesta, como se observa en la figura 3.5 y usando el valor de dos de estos picos, podemos usar la siguiente ecuación:

$$\zeta \approx \frac{1}{2\pi(j-i)} \ln \left(\frac{c_i - c_{ss}}{c_j - c_{ss}} \right) \quad (3.16)$$



AUTOCAD

Figura 3.5 Curva de respuesta al escalón de plantas altamente subamortiguadas

Se puede encontrar el valor de ζ a partir de picos adyacentes cuando las oscilaciones son más largas.

$$\zeta = \frac{\ln\left(\frac{c_i - c_{ss}}{c_{i+1} - c_{ss}}\right)}{\left[4\pi^2 + \ln\left(\frac{c_i - c_{ss}}{c_{i+1} - c_{ss}}\right)\right]^2} \quad (3.17)$$

Cuando ya se tiene el valor de ζ se puede encontrar el valor de la frecuencia natural ω_n con:

$$\omega_n = \frac{2\pi(j-i)}{(t_j - t_i) \sqrt{1 - \zeta^2}} \quad (3.18)$$

donde t_i e t_j son los tiempos en los que se dan los picos i e j .

Procesos sobreamortiguados:

Normalmente se puede aproximar a un proceso de primer orden con tiempo muerto. La constante de tiempo mayor domina la respuesta de la planta, que es cercana a la de primer orden y el efecto de la constante de tiempo menor se puede representar como un retraso.

$$d = \frac{\tau^2}{T} \quad (3.19)$$

Procesos críticamente amortiguados:

Se marca el punto donde el valor de la respuesta de la planta alcanza el 59,4 % del valor final o en estado estable, y se le llama al tiempo en que esto ocurre $t_{0.594}$, la constante de tiempo y la frecuencia natural se pueden encontrar con las siguientes ecuaciones:

$$\tau = \frac{1}{\omega_n} \quad (3.20) \quad \tau = \frac{1}{2} t_{0.594} \quad (3.21) \quad \omega_n = \frac{2}{t_{0.594}} \quad (3.22)$$

Segundo método [3]

Este método usa algunas fórmulas generales y otras específicas para cada tipo de sistema. Se explican las generales en el primer caso expuesto, para mostrar en los dos últimos casos sólo las ecuaciones específicas de estos.

Procesos sobreamortiguados:

El primer paso a seguir es encontrar la ganancia del sistema.

$$K = \frac{\Delta PV}{\Delta MV} \quad (3.23)$$

El segundo paso es determinar el valor de m_1

$$m_1 = \int_0^{\infty} [1 - c(t)] dt \quad (3.24)$$

el cual puede aproximarse en forma útil con la siguiente ecuación:

$$m_1 \approx \frac{T}{\Delta PV} \sum_{k=0}^{k=n} [PV_2 - PV(t_k)] \quad (3.25)$$

donde t_k es el tiempo en el que se da el escalón, T es el tiempo de muestreo o tiempo entre muestras, que debe ser igual en todos los casos, y PV_2 es el valor final de la planta. Un número adecuado de muestras puede estar entre 10 y 20.

Luego, se debe dibujar una recta tangente al punto en el cual la pendiente de la curva es máxima, como se muestra en la figura 3.7 para el caso subamortiguado. Se debe marcar el punto en el cual dicha recta alcanza el valor final de la curva de respuesta del sistema y el punto en el cual tiene el valor inicial de la variable de proceso, los tiempos en que esto ocurre se nombran como t_m y t_x respectivamente.

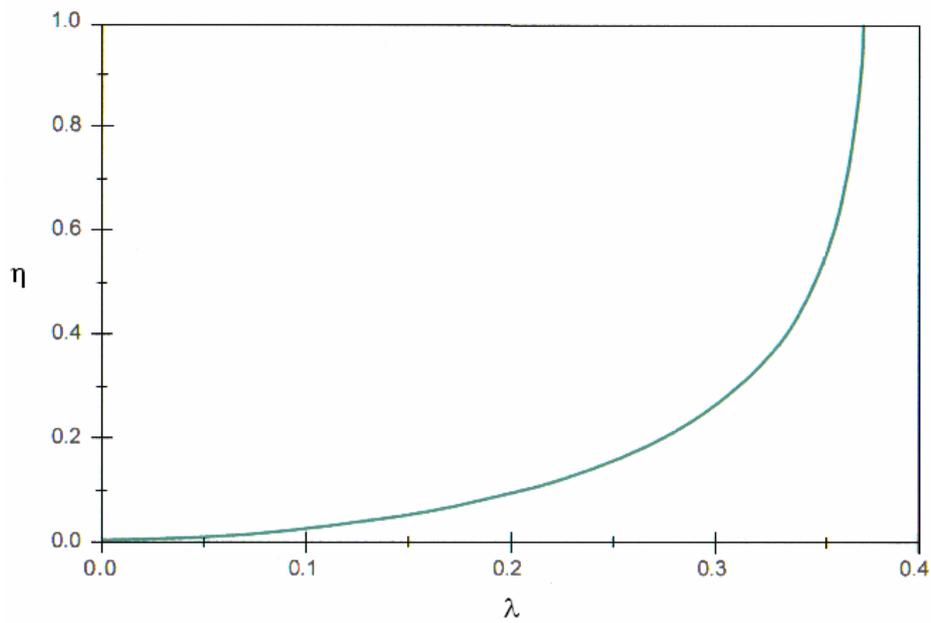
Calcular Mi de

$$Mi = \frac{1}{t_m - t_x} \quad (3.26)$$

Calcular λ de

$$\lambda = (t_m - T_1 - m_1) Mi \quad (3.27)$$

Ahora hay que determinar el valor de η de la figura 3.6.



AUTOCAD

Figura 3.6 Gráfica normalizada η vs λ

Por último, se puede hallar el valor de τ_1 , τ_2 y θ_D haciendo uso de las siguientes fórmulas:

$$\tau_1 = \frac{\eta^{1/(1-\eta)}}{Mi} \quad (3.28)$$

$$\tau_2 = \frac{\tau_1}{\eta} \quad (3.29)$$

$$\theta_D = m_1 - \tau_1 - \tau_2 \quad (3.30)$$

Procesos críticamente amortiguados:

Aquí se dá la particularidad de que τ_1 es igual a τ_2 y se calcula como sigue:

$$\tau_1 = \tau_2 = \frac{1}{eMi} \quad (3.31)$$

y la ecuación 3.30 se adapta para este caso de la siguiente manera:

$$\theta_D = m_1 - 2\tau_1 \quad (3.32)$$

Procesos subamortiguados:

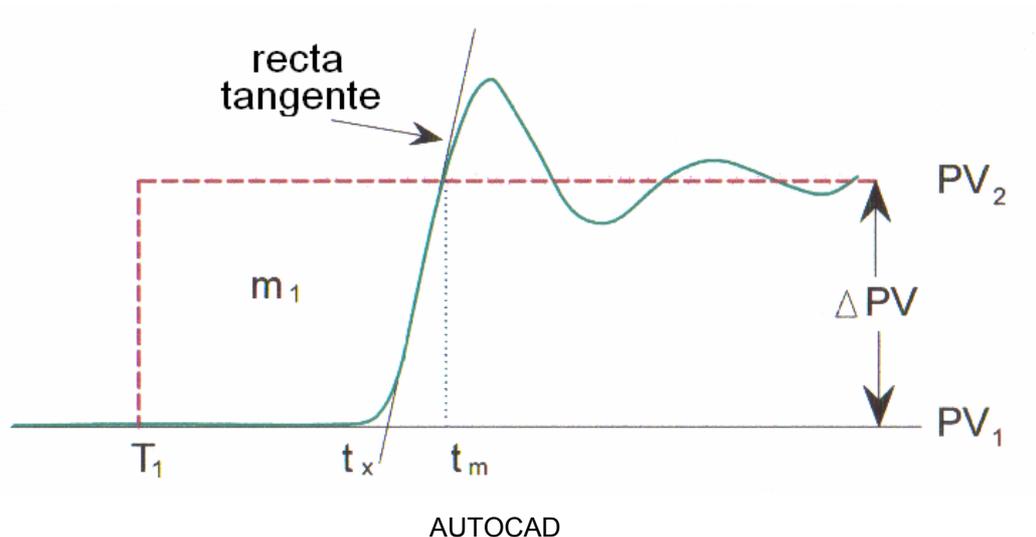
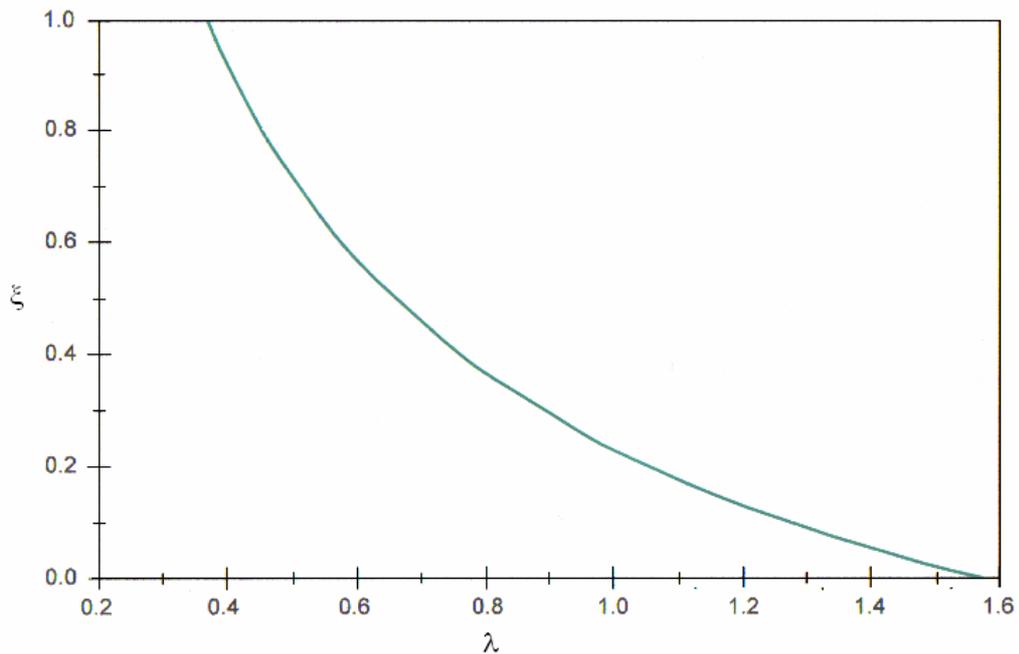


Figura 3.7 Respuesta al escalón de una planta de segundo orden subamortiguada

Este último caso es sólo una variación de los anteriores algoritmos. Cuando se tiene el valor de λ siguiendo el mismo procedimiento para sistemas sobreamortiguados, se obtiene el valor de ξ directamente de la figura 3.8.



AUTOCAD

Figura 3.8 Gráfica normalizada ξ vs λ

Finalmente, la frecuencia natural y el tiempo muerto se calculan con las ecuaciones 3.33 y 3.34 respectivamente:

$$\omega_n = \frac{\cos^{-1} \xi}{\sqrt{1 - \xi^2}} \frac{1}{t_m - T_1 - m_1} \quad (3.33)$$

$$\theta_D = m_1 - \frac{2\xi}{\omega_n} \quad (3.34)$$

¿Qué es un PID?

Según Ogata [11] el PID es un controlador formado por un PI y un PD en cascada.

El controlador PD consta de una parte proporcional y otra derivativa que resulta en sobreimpulsos menores tanto positivos como negativos, ya que su parte derivativa predice el sobreimpulso excesivo y corrige el error antes de que este ocurra.

El PD además, reduce el tiempo de levantamiento y el tiempo de asentamiento y su función de transferencia es:

$$G_c(s) = k_p + k_D s \quad (3.35)$$

El controlador PI tiene una parte proporcional y otra integral. La función de transferencia del controlador PI es:

$$G_c(s) = k_p + \frac{k_I}{s} \quad (3.36)$$

El PI mejora el error de estado estacionario en un orden, o sea, que si el error es constante, el PI lo reduce a cero.

El PID añade amortiguamiento a un sistema y mejora la estabilidad relativa y el error en estado estable, utilizando las mejores características del PI y del PD.

La función de transferencia del PID es

$$G_c(s) = k_p + k_D s + \frac{k_I}{s} \quad (3.37)$$

Métodos de sintonización PID [3]

Existen varios métodos para encontrar la sintonización PID adecuada para el control de sistemas, algunos gráficos y otros no. A continuación se detallan tres métodos que basan el cálculo de las tres constantes (proporcional, integral y derivativa) en el modelo empírico de la planta.

Estos métodos son llamados métodos de sintonización a lazo abierto ya que los sistemas no son controlados durante las pruebas. Todos están hechos para procesos autorregulados, y usan un modelo de primer orden con tiempo muerto del proceso.

Con estos métodos se puede también calcular las constantes para controladores P y PI; se incluirán también las ecuaciones para éste último.

Primer método

Fue propuesto por Zeigler y Nichols en 1942 [3], utiliza una constante llamada controlabilidad del proceso de la forma:

$$\alpha = \frac{\theta_D}{\tau_1} \quad (3.38)$$

La tabla 3.1 contiene las fórmulas necesarias para encontrar el valor de las diferentes constantes.

Tabla 3.1 Fórmulas de sintonización de Ziegler-Nichols

Tipo de controlador	K_P	T_I (min)	T_D (min)
PI	$\frac{0.9}{K\alpha}$	$3.333\theta_D$	0
PID	$\frac{1.2}{K\alpha}$	$2\theta_D$	$0.5\theta_D$

Segundo método [3]

Este método fue expuesto por Cohen y Coon en 1953 y presenta resultados similares al método anterior. Utiliza la misma constante de controlabilidad de la ecuación 3.38; las constantes de los controladores se calculan como se muestra en la tabla 3.2.

Tabla 3.2 Fórmulas de sintonización de Cohen – Coon

Tipo de controlador	K_P	T_I (min)	T_D (min)
PI	$\frac{1}{K} \left(\frac{0.9}{\alpha} + 0.082 \right)$	$\theta_D \left[\frac{3.333 + 0.333\alpha}{1 + 2.2\alpha} \right]$	0
PID	$\frac{1}{K} \left(\frac{1.35}{\alpha} + 0.27 \right)$	$\theta_D \left[\frac{2.5 + 0.5\alpha}{1 + 0.6\alpha} \right]$	$\theta_D \left[\frac{0.37}{1 + 0.2\alpha} \right]$

Tercer método

El último método es llamado el método Fertik [3], y a diferencia de los anteriores, éste produce una respuesta no oscilatoria. Se basa en unas pocas ecuaciones y una serie de curvas para encontrar las constantes de sintonización PID y PI.

La controlabilidad de Fertik se calcula de la siguiente manera:

$$\alpha_F = \frac{\theta_D}{\theta_D + \tau_1} = \frac{T_d}{T_{ps}} \quad (3.39)$$

T_d y T_{ps} se calculan de las ecuaciones 3.40 y 3.41 respectivamente.

$$T_d = \theta_D \quad (3.40)$$

$$T_{ps} = \theta_D + \tau_1 \quad (3.41)$$

Los parámetros linealizados se pueden obtener de las figuras de la 12 a la 16 y con esto las constantes de sintonización. El método Fertik no se recomienda para procesos en los que la controlabilidad α_F es mayor a 0.5.

Error estándar [10]

El error estándar determina el error entre el ajuste teórico y el ajuste experimental. Se obtiene a partir del cálculo de la desviación estándar entre la raíz del número de muestras. La desviación estándar depende del tamaño de la muestra y su fórmula es:

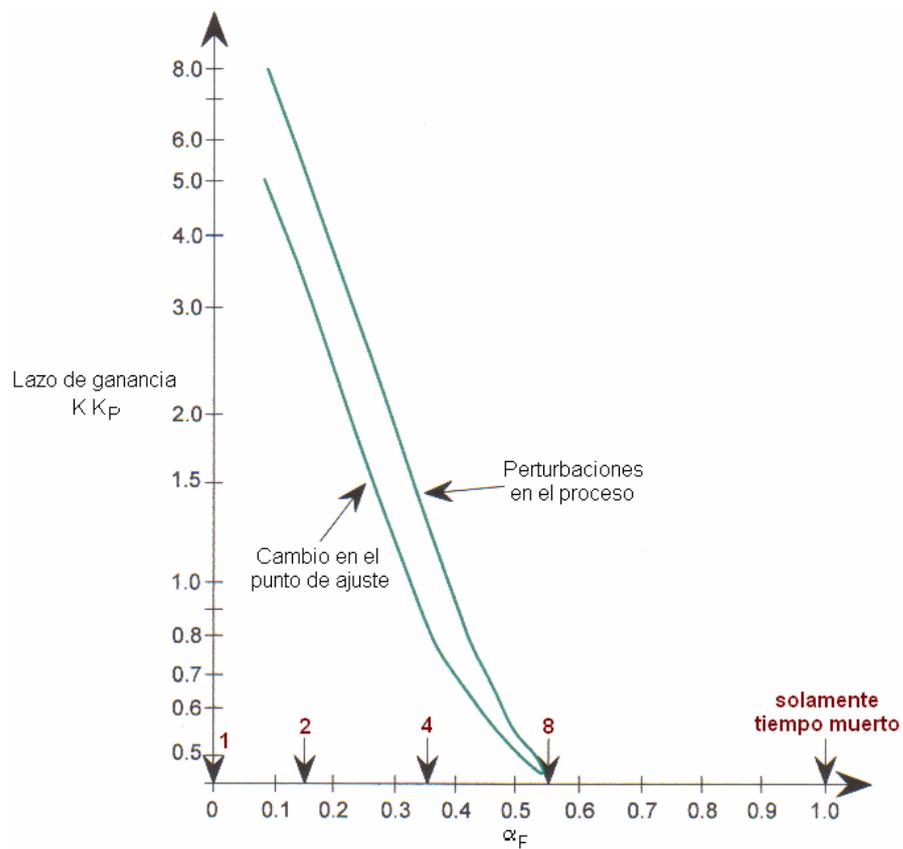
$$s = \sqrt{\frac{\sum_{n=0}^i (y_{1i} - y_{2i})^2}{n}} \quad (3.42)$$

por lo que la ecuación para el error estándar es:

$$s_{y_{ii}} = \frac{s}{\sqrt{n}} \quad (3.43)$$

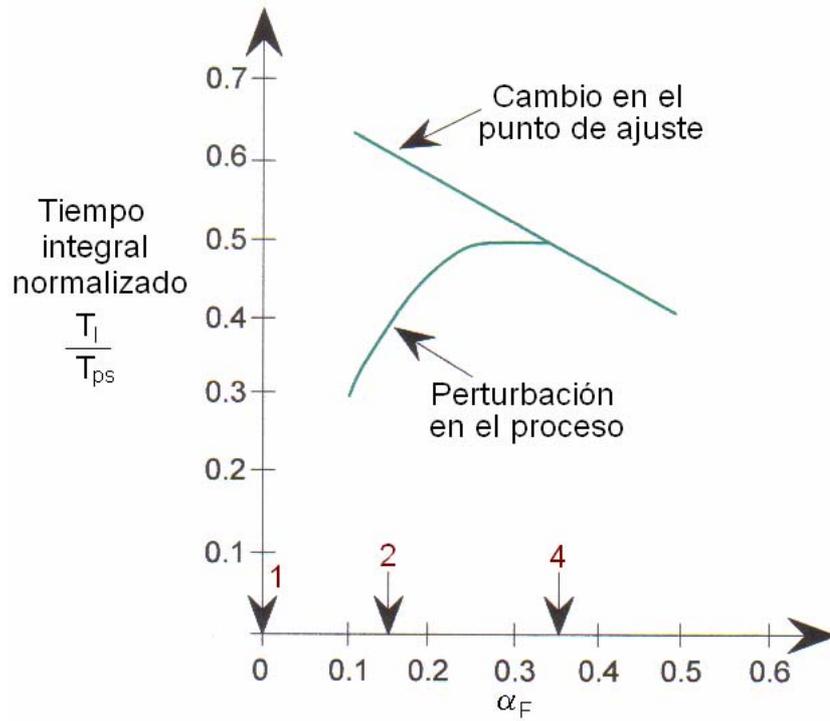
donde n es el número de muestras.

Una aplicación del error estándar es el de encontrar el ajuste entre dos curvas. Entre más bajo es el error, mayor es el ajuste entre las curvas.



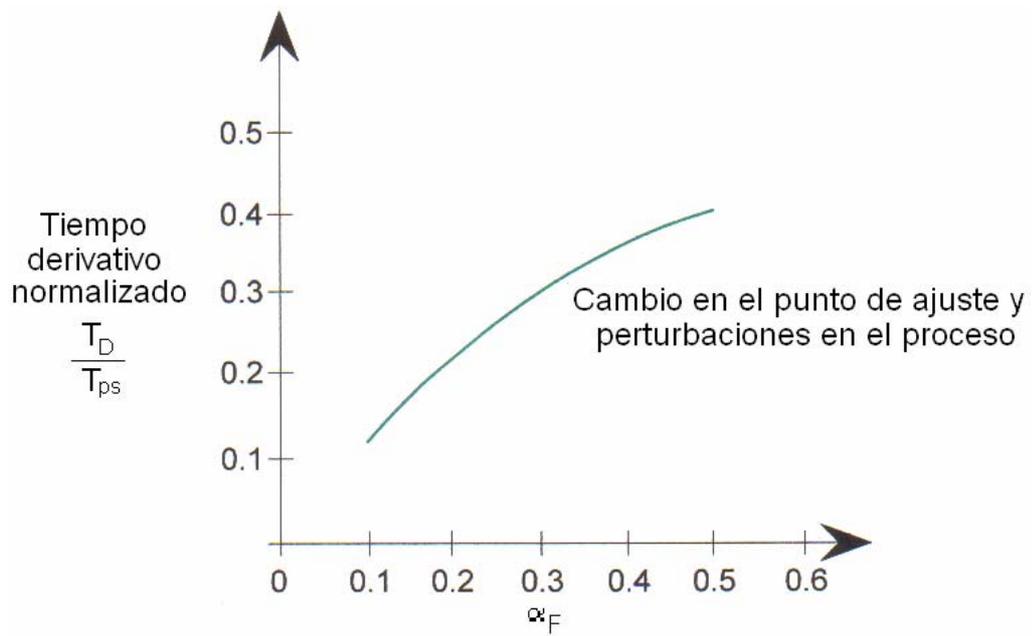
AUTOCAD

Figura 3.9 Ganancia de Fertik para un control PID



AUTOCAD

Figura 3.10 Tiempo integral de Fertik para un control PID



AUTOCAD

Figura 3.11 Tiempo derivativo de Fertik para un control PID

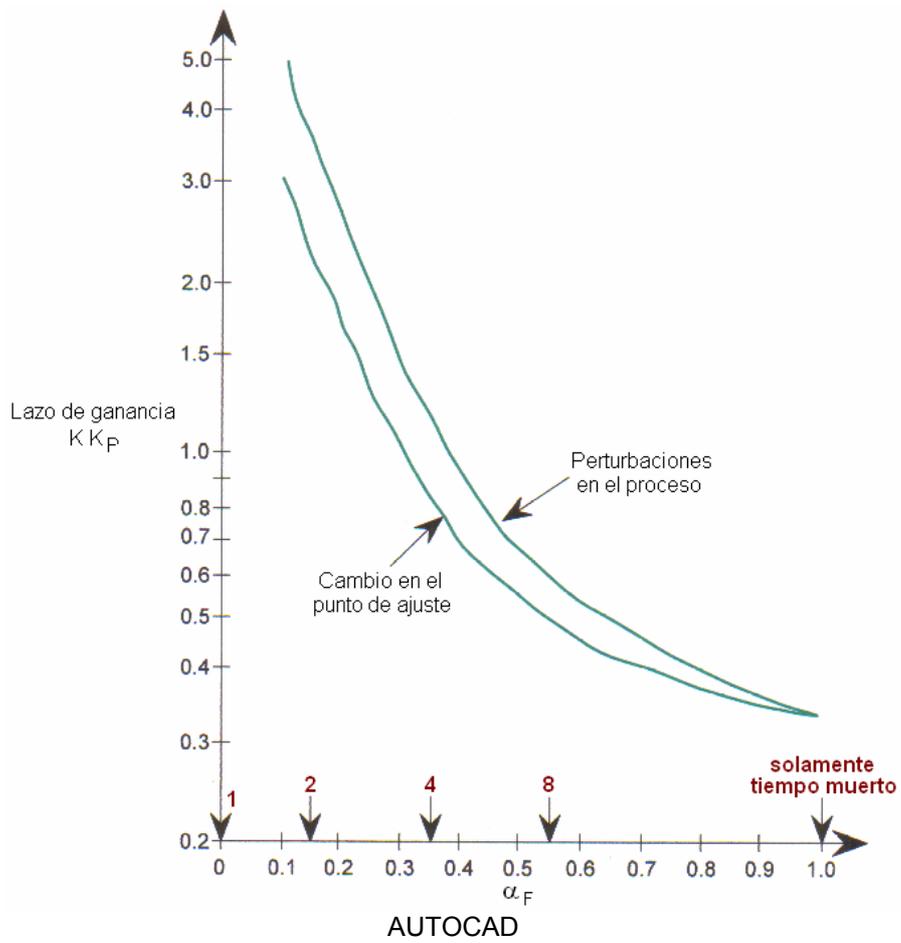


Figura 3.12 Ganancia de Fertik para un control PI

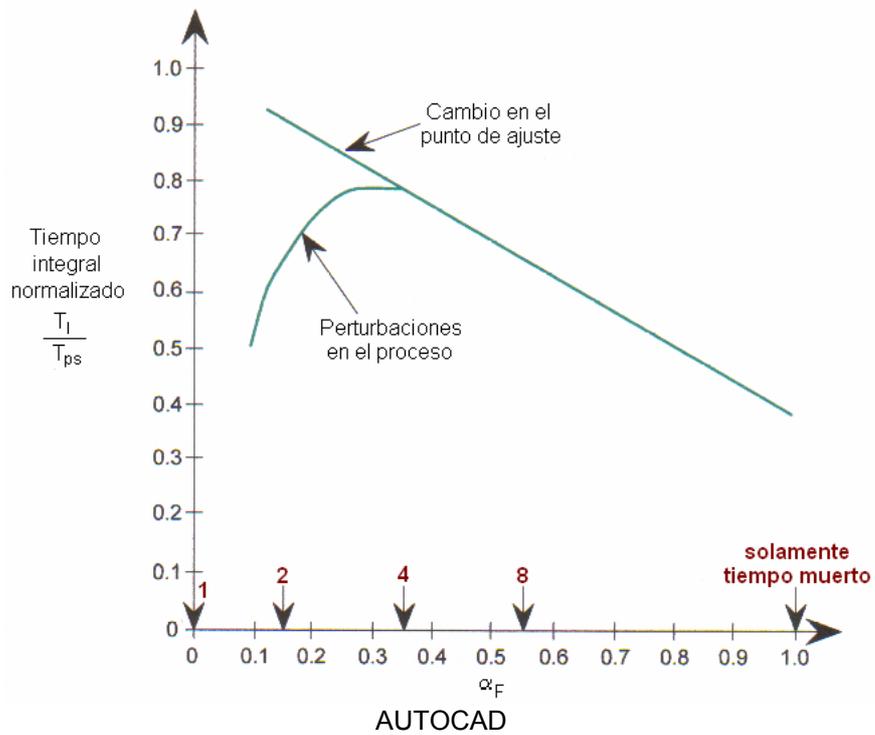


Figura 3.13 Tiempo integral de Fertik para un control

Capítulo 4: Procedimiento metodológico

4.1 Reconocimiento y definición del problema

Una de las formas utilizadas para identificar el problema fue por medio de entrevistas a los diferentes ingenieros que desarrollan proyectos dentro de la empresa y de las recomendaciones del ingeniero de proyectos. También se pudieron definir algunos elementos por medio de la observación del trabajo que realizan dichos funcionarios.

Se cuenta con software especializado como InTouch e InControl, con los que se hizo una serie de simulaciones del comportamiento de una planta como respuesta ante un escalón. Se hizo esta prueba con diferentes funciones de transferencia para la planta (con y sin tiempo muerto, de primer y segundo orden, etc).

Con el software Matlab se ha podido verificar algunas de las etapas del proceso y puntualizar cuáles son las principales complicaciones que éste presenta. Además con el uso del Ident, que es una herramienta contenida en este programa, se pudo encontrar la función de transferencia estimada por este programa para diferentes respuestas ante la función escalón.

Para este proyecto se quiere plantear tres posibles soluciones para la generación del modelado de la planta y tres posibles soluciones para la obtención de la sintonización PID.

Las propuestas planteadas para la creación de un modelado matemático de las plantas de primer orden (función de transferencia), se han probado exhaustivamente por medio del software Matlab, se han generado programas para cada uno de estos métodos y se ha calculado el error que cada uno de estos genera.

Se escogerá la opción que dé un resultado más exacto y que pueda ser programado en el lenguaje Visual Basic; además, debe ser el que tenga mayor alcance en cuanto a tipos de plantas se refiere.

Para proponer una buena sintonización PID, el algoritmo escogido debe de poderse programar en Visual Basic y además debe tener gran precisión en dicha propuesta; y ser realizable su programación en un PLC.

4.2 Obtención y análisis de información

Los temas relacionados con la respuesta al escalón de una planta y la obtención de la función de transferencia de la misma se estudian de libros de teoría de control. Estos libros se consiguen en la biblioteca del ITCR.

Para la sintonización PID se usarán libros específicos sobre los PIDs; además, se consultarán manuales de softPLC para aprender la programación de los PIDs en PLCs. En este caso la empresa proporciona estos manuales.

Se estudiarán los manuales de InTouch e InControl como sistema de aprendizaje de dichos software, así como los tutores correspondientes. Igual que en el caso anterior la empresa tiene esta información.

Se usará el manual o curso de Visual Basic y una guía sobre objetos ActiveX que se obtienen en Internet; esto se consigue a partir de una búsqueda por parte de la estudiante.

Se hará uso de las notas técnicas de Matlab, realizadas por el asesor del proyecto en la empresa, en donde se hace referencia a la interacción de éste con otros programas.

En cuanto a InTouch e InControl son software que la empresa usa y que además vende a sus clientes, por lo que su uso no sólo es accesible sino más bien necesario.

El lenguaje Visual Basic se escoge porque con éste se pueden crear objetos ActiveX y ya que es un lenguaje bastante utilizado, la información a la que se tiene acceso sobre éste es amplia.

4.3 Evaluación de las alternativas y síntesis de una solución

Se cuenta con cuatro métodos diferentes para hallar la función de transferencia de una planta de primer orden, estos se plantearon después de una búsqueda bibliográfica tomando en cuenta las sugerencias planteadas por los asesores del proyecto.

Para este punto se propusieron tres funciones de transferencia para estimarlas haciendo uso de cada una de las soluciones. Los métodos se verificaron primero usando solamente una calculadora.

Cuando se obtuvieron respuestas exitosas con cada método, se procedió a hacer uso del programa Matlab para desarrollar algoritmos que sirvieran para su posterior programación. Se hicieron dos programas en Matlab, para dos de estos algoritmos.

Luego, se programaron estos métodos en Visual Basic generando dos programas que estiman los parámetros de la planta.

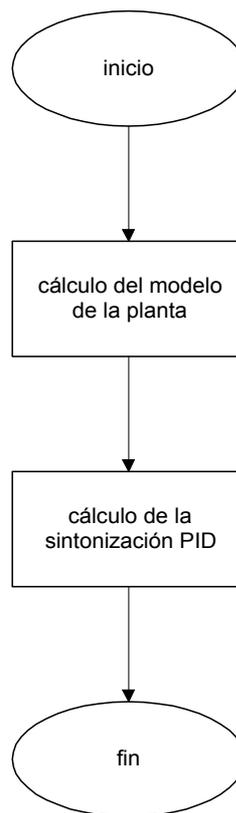
Se calculó el error cuadrático medio para cada uno de los diferentes métodos estudiados, comparando la curva teórica con la curva obtenida a partir de estos.

Los datos que se usaron para hacer las estimaciones antes mencionadas, se generaron con una simulación en los programas InTouch e InControl, en la cual se podía determinar o forzar la función de transferencia de la planta para obtener la respuesta deseada al aplicar un estímulo tipo escalón. Con InTouch se generaba un archivo que guardaba los datos de las diferentes variables de la planta.

4.4 Implementación de la solución

En el presente proyecto se trabaja enteramente con *software* y el único *hardware* que se usa es una computadora. Es por esto que la implementación de la solución es en este caso la programación de los diferentes algoritmos.

La solución esta dividida en dos grandes partes: los procedimientos necesarios para averiguar el modelo de la planta y el método de estimación de la sintonización PID. La primera parte no necesita de la segunda para ser implementada, mientras que la segunda si requiere de la primera parte, esto se puede observar en el diagrama de la figura 4.1.



VISIO32

Figura 4.1 Secuencia de pasos para la implementación de la solución

Debido a lo anterior se implementa la solución para encontrar el modelo de la planta sin que se tenga todavía una solución para la sintonización PID. Lo que se hace es un objeto ActiveX llamado PIDX, que encuentra el modelo de sistemas de primer orden y se comprueba que funciona correctamente, luego se añade la solución encontrada para el modelado de procesos de segundo orden programándola en el objeto ActiveX.

La solución que se encuentra para la sintonización del controlador necesita del cálculo previo del modelo de la planta, es por esto que esta es la última parte de la solución que se realiza.

Como parte del proyecto se encuentra una aplicación que sea capaz de simular el comportamiento de distintos sistemas (distintos modelos) y en la cual se pudiera sintonizar un PID. Esta simulación se hizo en los programas InTouch e InControl en diferentes partes, cada vez que se implementa una nueva parte de la solución se debe de alterar la programación de la simulación para que esta trabaje según lo que se necesita en ese momento. Esta simulación sufrió una serie de cambios durante el proceso de implementación antes de tener una versión final de la misma.

Otra parte que también se modificó es el algoritmo empleado para hallar el modelo de plantas sobreamortiguadas. Se implementa una solución con un algoritmo que se había probado previamente, sin embargo, debido a las múltiples aproximaciones que se deben hacer cuando se programa el algoritmo resultó en datos errados.

Como se mencionó anteriormente, se cuenta con una simulación en la cual se puede evaluar el desempeño de la solución implementada para las diferentes partes que conforman el PIDX. Se prueba de esta forma ya que en la empresa el objeto va a utilizarse para hacer pruebas a sistemas reales simulados en estos programas.

Para comprobar que los parámetros calculados por el programa son verdaderos, o válidos, se emplearon varias comparaciones. La primera es

directamente en la parte de la simulación que corresponde a InControl. En ésta se inserta un objeto en el que se puede escribir el modelo que determina el comportamiento del proceso. Así, se pueden comparar los datos encontrados por el objeto ActiveX contra los datos teóricos introducidos en InControl.

Otra forma para comprobar la validez del programa es graficar la curva de respuesta teórica junto con la curva de respuesta del modelo encontrado. Esto se hizo en el programa Matlab en donde se puede simular la respuesta al escalón de cualquier modelo.

Por último se incorporó como parte de la solución el cálculo del error estándar como parte de las respuestas del PIDX, así el usuario puede saber la validez de la estimación realizada.

4.5 Reevaluación y rediseño

Este proyecto puede servir de base para desarrollar *software* que estimen la sintonización PID para sistemas con algún comportamiento en específico.

El método utilizado para esto es una modificación del método de Ziegler y Nichols para lazo abierto. El valor de la constante de tiempo derivativa se disminuye para adaptarla a la fórmula empleada por el PID de InControl.

Si se quisiera ampliar el alcance de este proyecto para estimar la sintonización de controladores PID que utilicen otras ecuaciones (por ejemplo PLC de distintas marcas), deberá ajustarse cada una de las constantes a la nueva fórmula, ya sea por medio de equivalencia matemáticas o haciendo tantas pruebas como sea posible para diferentes tipos de plantas hasta encontrar un patrón de comportamiento de la planta.

Para cualquier modificación que se quiera hacer a la sintonización es importante saber el valor de la constante de controlabilidad del sistema, ya

que esta determina en gran medida el comportamiento que presenta la planta ante un controlador.

Por otro lado si se quiere tener el modelo exacto de los sistemas de segundo orden sobreamortiguados y críticamente amortiguados para alguna otra aplicación esto debe incorporarse al sistema, no sin antes evaluar el análisis de los distintos métodos a los que se hace referencia en este documento, ya que existen métodos más adecuados para cierto tipo de aplicaciones mientras que otros no es posible adaptarlos para el propósito buscado.

Capítulo 5: Descripción detallada de la solución

5.1 Análisis de soluciones y solución final

5.1.1 Simulación

Se plantea desde el principio que la simulación debe hacerse con los programas InTouch e InControl. La simulación consiste en un tanque de agua, hay que controlar la temperatura del agua contenida en este tanque, el punto de ajuste de esta temperatura se puede variar.

La primera solución que se usa es la de un control On – Off el cual abre y cierra un interruptor que corta el flujo de corriente de un calentador resistivo. Esta solución no es muy eficiente ya que la temperatura del agua no permanece estable. En esta versión de la simulación el comportamiento de la planta no está definido con ningún modelo, el agua se calienta en escalones de 5 °C hasta que el interruptor se desconecta. Se usan como límites superior e inferior de la temperatura ± 5 °C de la temperatura deseada.

El siguiente paso es insertar un modelo de comportamiento para el sistema. Se usa un objeto de InControl en el que se establecen los diferentes parámetros del modelo, puede ser de primer o segundo orden con tiempo muerto, y se le asigna cuál es la variable que tendrá este comportamiento. De esta forma se logra simular diferentes comportamientos para los cambios de temperatura del agua.

Con la simulación antes descrita se puede probar el primer programa que calcula el modelo de primer orden de una planta. Para esto se inserta un objeto en InTouch el cual lee las diferentes variables del sistema simulado y las guarda en un archivo tipo .csv de donde el otro programa lee los datos y con estos calcula el modelo.

El siguiente y último cambio significativo de la simulación es el reemplazo del controlador On – Off por un controlador PID desde InControl. Esto se hace para probar la versión final del objeto ActiveX que se crea en Visual Basic, que además de calcular el modelo, calcula también las constantes de sintonización PID. Los datos ya no se leen desde un archivo sino que se leen directamente del simulador, los resultados del programa se pueden insertar en el PID del simulador y así comprobar su funcionamiento.

Para hacer la prueba escalón se debe desconectar el PID, por lo que se hace en InTouch una función que realice esto para realizar ambas pruebas juntas. También es necesario que cuando en InTouch se realiza la prueba al escalón en InControl el PID esté inactivo.

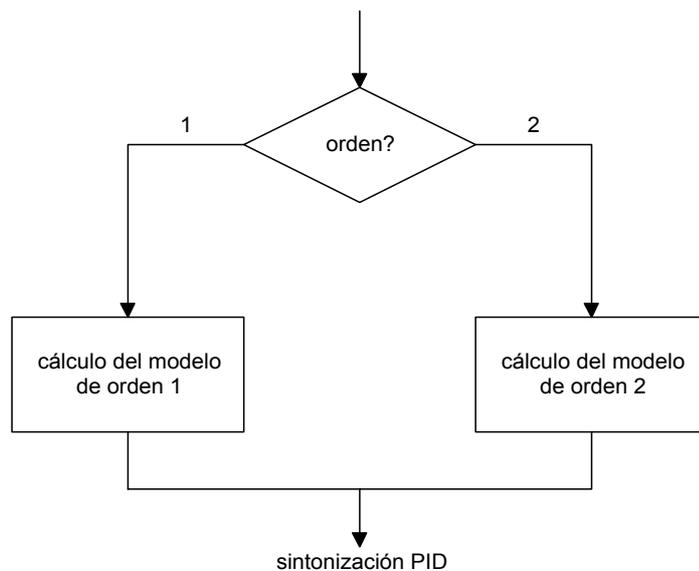
5.1.2 Modelo de la planta

Se estudiaron diferentes métodos para la obtención del modelo de la planta; estos son:

- a. Modelo de primer orden más tiempo muerto, 2 métodos [3]
- b. Modelo de segundo orden más tiempo muerto (sobreamortiguado, subamortiguado y críticamente amortiguado) ambos expuestos en [3].
- c. Método de las áreas características.
- d. Método de identificación directa de segundo orden con tiempo muerto.
- e. Sistemas de primer orden con tiempo muerto y sistemas de segundo orden.
- f. Identificación de sistemas de primer y segundo orden.
- g. Estimación de mínimos cuadrados (ARMA y MA) de [13].

De los anteriores métodos se decide que los más adecuados para encontrar el modelo son los métodos gráficos, ya que varios de ellos utilizan los datos obtenidos a partir de la respuesta al escalón de un sistema para hallar el modelo de éste, lo cual es uno de los objetivos de este proyecto. Usando este criterio se escogieron seis de los métodos antes mencionados.

Los métodos escogidos se dividen en dos categorías, modelos de primer orden con tiempo muerto y modelos de segundo orden con tiempo muerto, se cuenta con cuatro para primer orden y dos para segundo orden. En la figura 5.1 se muestra como el uso de uno u otro algoritmo depende directamente del orden de la planta en estudio.



VISIO32

Figura 5.1 División del algoritmo en primer y segundo orden para calcular el modelo de la planta

Modelo de primer orden con tiempo muerto

Los cuatro métodos para primer orden son:

- a. Método de Ziegler – Nichols [3].
- b. Método de Marlin [3].

- c. Método expuesto en [13].
- d. Método discreto de primer orden con tiempo muerto [1].

Para encontrar el modelo de una planta de primer orden con tiempo muerto se busca saber los tres parámetros que la caracterizan, a saber: la ganancia del sistema, la constante de tiempo característica del sistema y el tiempo muerto; esto es común para todos los métodos.

Se probó la funcionalidad de los cuatro métodos, en primera instancia, mediante la creación de una aplicación en InTouch e InControl, la cual puede simular la respuesta escalón de sistemas con diferentes valores característicos.

Esta curva de respuesta es guardada en forma automática en un archivo tipo .csv el cual puede ser leído desde el programa Excel. Guarda los valores de la entrada y la salida de la planta, así como el tiempo en que se tomó cada dato. El intervalo de tiempo entre muestras puede cambiarse a conveniencia.

Una vez que los datos están en Excel se puede entonces crear la gráfica característica de la planta a un estímulo tipo escalón.

Con dichas curvas se obtienen los parámetros de diferentes plantas con los cuatro métodos mencionados, los cuales son comparables a los parámetros teóricos. Se observa que los métodos citados por Bollinger y el expuesto por Taylor & Francis group, pese a tener cada uno su forma particular de calcular los parámetros, dan resultados iguales hasta el tercer decimal, por lo que se decide trabajar sólo con el segundo.

El siguiente paso fue probar los algoritmos en Matlab. Para esto se desarrollan los diferentes métodos, haciendo un programa por cada uno de ellos. A los programas se agrega el cálculo del error de la estimación, para lo cual se comparan los puntos que conforman la curva de respuesta al escalón teórica con los puntos de la curva obtenida con el método, para calcular finalmente el error estándar. Después de realizadas las pruebas

se encuentra que el error estándar no supera el 1.6 % en ninguno de los casos.

Con estos resultados se pasa a la etapa de programación en Visual Basic. Primero se crea un programa que estima los tres parámetros característicos de los sistemas de primer orden usando el método de Merlin. Para esto el programa lee los valores del archivo tipo .csv generado en InTouch, los guarda en una matriz temporal y aplica las diferentes ecuaciones del método.

Cuando el programa necesita algún valor específico de la curva, toma el valor del punto más cercano al resultado obtenido, y continúa con las ecuaciones. La diferencia entre ambos puntos introduce una desviación de milésimas de unidad, por lo que la estimación es bastante precisa. Ver apéndice B.1.

Luego, se programó, utilizando la misma metodología, el método discreto, con el que se obtuvieron resultados semejantes al programa anteriormente explicado. Se decide usar este último porque como se explica más adelante, es el mismo método que se utiliza para la estimación de los modelos de segundo orden.

Los resultados de ambos programas se usan para obtener la curva de respuesta al escalón el sistema en Matlab y compararla con la curva de respuesta teórica, para confirmar el ajuste entre curvas que predice el error calculado por el programa.

Modelo de segundo orden con tiempo muerto

En el caso de estimar el modelo para una planta de orden 2, se necesita encontrar un algoritmo que pueda calcular la función de transferencia de sistemas subamortiguados, sobreamortiguados y críticamente amortiguados. Por las mismas razones que se dieron para el modelo de primer orden, se escogieron los métodos gráficos.

Los métodos que buscan describir un modelo de segundo orden a partir de la respuesta al escalón de la planta son los siguientes:

- a. Modelo de segundo orden más tiempo muerto [3].
- b. Sistemas de segundo orden [1].

Se encuentra que la mayoría de estos métodos plantean un algoritmo diferente para las plantas que presentan un comportamiento subamortiguado (existen oscilaciones) y otro para las que comprenden los sistemas sobreamortiguados y críticamente amortiguados. Debido a lo anterior el estudio del algoritmo se divide en dos, dando origen al estudio de dos algoritmos diferentes para plantas de orden 2.

Para los sistemas subamortiguados se encuentran dos métodos. El primer método consiste en realizar una serie de cálculos matemáticos en los que se incluye una sumatoria de diferentes puntos de la curva de respuesta, y la búsqueda de valores en una curva normalizada para encontrar finalmente la función de transferencia de la planta.

El otro método se basa en ubicar los sobreimpulsos que presenta la curva de respuesta al escalón y con el uso de una serie de ecuaciones se llega a la ecuación característica del sistema. Este segundo método consiste en el uso de algunas pocas ecuaciones matemáticas, por lo que se procede a desarrollar un programa en Matlab para así evaluar su eficiencia.

El primer algoritmo para plantas sobreamortiguadas plantea que en el caso de esos sistemas, se puede aproximar el modelo de la planta a una de orden uno, tomando en consideración que la constante de tiempo de mayor valor será la que domine su comportamiento y se compensa el efecto de la otra constante de tiempo aumentando el tiempo muerto de la planta. Para el caso críticamente amortiguado se utilizan dos ecuaciones, una para encontrar la constante de tiempo y otra para encontrar ω_n , cuyos valores son inversamente proporcionales.

El otro método resuelve ambos casos en un solo algoritmo. Este se ocupa de resolver una serie de ecuaciones con el fin de encontrar un valor λ para luego utilizar una curva normalizada y hallar el valor correspondiente de η . Una vez obtenido se pueden encontrar los valores de las dos constantes de tiempo que caracterizan al sistema.

En el caso de estimar el modelo a un modelo de primer orden, se encuentra que la constante de tiempo del sistema define correctamente al sistema, sin embargo, el tiempo muerto agregado es muy grande y aumenta el error considerablemente. Considerando lo anterior y en vista de que se querían obtener las dos constantes de tiempo, se sigue trabajando con el segundo método.

Uno de los principales problemas de este segundo algoritmo es el uso de una gráfica para hallar uno de los valores necesarios para encontrar las constantes de tiempo del sistema. Se utilizó este método para encontrar dichas constantes para varias plantas. La diferencia entre el valor de las constantes de tiempo encontradas y el valor teórico es notorio, sin embargo, al graficar y comparar la curva de respuesta al escalón obtenida con la curva de respuesta teórica se observa una similitud notable dadas las mínimas diferencias entre una y otra.

Se usa el algoritmo estudiado de [1] para desarrollar un programa en Matlab, el cual es capaz de encontrar la función de transferencia de una planta con respuesta subamortiguada a partir de su curva de repuesta al escalón. Los resultados obtenidos describen el comportamiento subamortiguado esperado, esto se comprueba aplicando un estímulo tipo escalón a la planta calculada y comparando su curva de respuesta con la curva de respuesta teórica.

Dado lo anterior, se eligió este algoritmo para programarlo en Visual Basic. Debido a que el método utilizado no calcula el tiempo muerto del sistema, se incorpora en esta parte un cálculo sencillo planteado en [3], que está hecho para sistemas de orden uno, pero que funciona muy bien para los sistemas de orden dos.

Para hacer el programa que calcula la función de transferencia de plantas sobreamortiguadas y críticamente amortiguadas se utiliza un único algoritmo. El primer problema que se presentó a la hora de usar este algoritmo en Visual Basic es la sumatoria de entre 10 y 20 de los puntos que conforman la curva de respuesta de la planta ya que se tenía que encontrar una cantidad fija de muestras de datos que funcionara para todos los casos.

Se elige una muestra de 10 puntos contenidos entre el 5 % y el 95 % del valor final de la planta, o sea, durante la respuesta transitoria. Lo anterior se hace ya que a la hora de hacer la prueba al escalón a la planta, puede haber una cantidad significativa de muestras de valores en estado estable, en relación con el total de las muestras y esto podría alterar el resultado. Se prueba y se ajusta exhaustivamente hasta alcanzar resultados acordes a lo deseado.

Seguidamente se procede a discretizar la curva normalizada que se usa para encontrar el valor de η y así el valor de las constantes de tiempo. Aquí se presenta un problema mayor. La escala de valores de η se encuentra entre 0 y 1, siendo 1 el caso críticamente amortiguado. Al ser una escala tan pequeña y la diferencia entre uno y otro valor todavía más pequeña, se provoca que las estimaciones arrastradas desde la sumatoria y otras ecuaciones que forman parte de este método, tengan como consecuencia que el valor de η encontrado no sea satisfactorio y los valores finales de las constantes de tiempo sean muy diferentes a las esperadas.

Dado este resultado se retorna al concepto de aproximar el modelo a un modelo de primer orden con tiempo muerto. Como ya se cuenta con un programa que hace este cálculo para procesos de primer orden, se utiliza este mismo programa pero ahora con sistemas de segundo orden.

Se observa que la constante de tiempo calculada se aproxima a la constante de mayor valor del sistema, y que el tiempo muerto aumenta, reafirmando la teoría expuesta en [1]. Sin embargo no se usa este último

método sino el que ya está programado, primeramente para analizar los resultados.

Se grafica la curva de respuesta al escalón teórica así como la obtenida mediante la aproximación de primer orden, encontrando que la aproximación del modelo describe el comportamiento del proceso de segundo orden.

Se genera un objeto ActiveX que usa los algoritmos seleccionados para encontrar los modelos de primer y segundo orden. Primero se elige el algoritmo para calcular el modelo de primer orden y se crea un objeto ActiveX con este algoritmo, el cual se utiliza anteriormente en un programa en Visual Basic.

Cuando se tienen los algoritmos para modelos de segundo orden estos se anexan al objeto ActiveX (PIDX) existente y este se incorpora a la simulación hecha. En la simulación el PIDX interactúa con diferentes variables de dicha simulación, los cuales utiliza para realizar los diferentes procedimientos necesarios para hallar el modelo de la planta y los resultados los escribe en variables asignadas de la simulación.

5.1.3 Sintonización PID

Los métodos de sintonización PID estudiados se dividen en dos grandes grupos: Métodos de sintonización a lazo cerrado y métodos de sintonización a lazo abierto.

Métodos de sintonización a lazo cerrado: [3]

- a. Sintonización basada en la respuesta del proceso.
- b. Sintonización de Ziegler-Nichols.
- c. Sintonización de oscilación.

Métodos de sintonización a lazo abierto: [3]

- a. Sintonización de Ziegler-Nichols.
- b. Sintonización de Cohen-Coon.
- c. Sintonización de Fertik.

En este punto se quiere aprovechar el trabajo hecho hasta ese momento para encontrar el modelo de la planta, decidiendo trabajar con los métodos de sintonización a lazo abierto.

Sintonización PID para sistemas de primer orden con tiempo muerto

Como se menciona anteriormente, para la sintonización PID se cuenta con tres métodos, para los cuales se hicieron diferentes pruebas. Estas pruebas consisten en utilizar los distintos métodos para hallar la sintonización PID de varios modelos de plantas de orden uno.

El método de Fertik parecía ser el más adecuado ya que está hecho para que la respuesta de la planta sea no oscilatoria, lo cual se desea en la mayoría de los casos con el fin de evitar posibles daños en los equipos que se utilizan en los distintos procesos.

Se utiliza entonces el método Fertik para estimar la sintonización PID de tres plantas diferentes, y se encuentra con que está hecho para modelos en los que la constante de controlabilidad de la planta αF es menor o igual a 0.5. Aunque esto se sabía de antemano, lo que se encuentra a la hora de utilizar el método es que muchos de los procesos están fuera del rango de funcionalidad de Fertik lo que lo convierte en un método poco funcional para los propósitos del proyecto.

Al realizar las prueba para los otros dos métodos se observa que con el método de Cohen - Coon los sistemas oscilan más que con la sintonización Ziegler - Nichols, por lo que se decide trabajar finalmente con este último método.

Es importante mencionar que para realizar estas pruebas y comprobar los resultados obtenidos, se recurrió a la simulación hecha con InTouch e InControl. Los parámetros de la planta, los que conforman el modelo, se introducen para así simular el comportamiento real de la planta, y los resultados que se obtienen de los diferentes métodos de sintonización se colocan en un PID que controla el comportamiento de la planta.

Al contar con un objeto ActiveX que encuentra el modelo de la planta, se incorpora a éste el algoritmo Ziegler - Nichols de sintonización PID.

Luego de añadir la estimación de las constantes de sintonización PID se encuentra que la constante derivativa del objeto PID del simulador, no coincide con la constante que se obtiene como resultado del PIDX. Lo anterior lleva a hacer más pruebas y a encontrar, basándose en el método, cuál es la forma correcta de calcular dicha constante para el objeto PID de InControl.

Se llega a la conclusión de que es necesario escalar el resultado obtenido con el método, antes de programarlo en el objeto PID de la simulación. Lo que se hace es dividir el valor obtenido mediante el método y dividirlo entre 200. Esto último se programa en el PIDX, además se incorpora el cálculo de la sintonización para un controlador PI.

En algunos casos no es necesario que la constante de tiempo derivativo se reduzca en un 200 %, sin embargo éste es el resultado obtenido en la mayoría de los casos, y lo que se busca es usar un método general.

El siguiente paso fue encontrar la forma de sintonización PID que diera resultado para planta con respuesta rápida, esto quiere decir, con tiempo muerto cercano a cero o cuya constante de tiempo fuera pequeña, de un valor de uno o menos. Lo anterior dado que ninguno de los métodos sirve para este fin, ya que están hechos para sistemas con respuesta lenta.

Se hicieron numerosas pruebas de sintonización para diferentes plantas con el fin de establecer relaciones que facilitaran una precisa

predicción de las constantes de sintonización; la base de estas pruebas sigue siendo el método de sintonización a lazo abierto de Ziegler-Nichols.

Se dedujo finalmente que si se utilizan directamente los valores que resultan del método de Ziegler – Nichols para las constantes de sintonización PID los sistemas presentan sobreimpulsos mayores al 25% y que antes de estabilizarse oscilan por un largo periodo, en algunos de los casos el sistema no se estabiliza y en otros casos el control sobre el sistema no existe.

Por esto se procede a disminuir la ganancia y aumentar el tiempo integral cuando se quiere controlar estos sistemas. El programa reconoce que se trata de esta clase de procesos por medio del valor de controlabilidad que presenta la planta.

Otro cambio que se hace al algoritmo original es que cuando las plantas presentan valores de α cercanos a uno el sistema tampoco puede ser controlado por lo que se utiliza la misma conjetura del caso anterior sólo que esta vez la ganancia disminuye en menor grado, un 50%, y la constante integral aumenta al doble de su valor.

Al PIDX se incorporan todos estos cambios y además se incluye la sintonización PI cuyo análisis es similar al del controlador PID. Esto se hace como una alternativa de solución si se diera el caso de que el sistema no se estabilice con el PID, o se quiera reducir el sobreimpulso inicial generado por la constante de tiempo derivativa.

Sintonización PID para sistemas de segundo orden con tiempo muerto

Debido a que todos los métodos estudiados para encontrar la sintonización PID a partir del modelo de la planta utilizan aproximaciones a modelos de primer orden, éste es el procedimiento empleado en éstos casos.

Para estimar la sintonización de plantas de segundo orden, el programa aproxima el modelo de la planta a un modelo de primer orden simultáneamente con el cálculo de los valores del modelo de segundo orden. Anteriormente se analizó que las aproximaciones de primer orden predecían el comportamiento real de plantas de segundo orden, por lo que no se explicará de nuevo.

La diferencia que hay con el cálculo de la sintonización para sistemas de primer orden es que los cambios que se hacen al método, o sea, el de disminuir la ganancia y aumentar la constante de tiempo integral, se hacen para valores distinto de la constante de controlabilidad, ya que los sistemas de segundo orden se comportan diferente ante el control PID.

En la figura 5.2 se muestra a grandes rasgos la lógica que se sigue en la programación del algoritmo descrito para la sintonización PID y PI para procesos de primer y segundo orden.

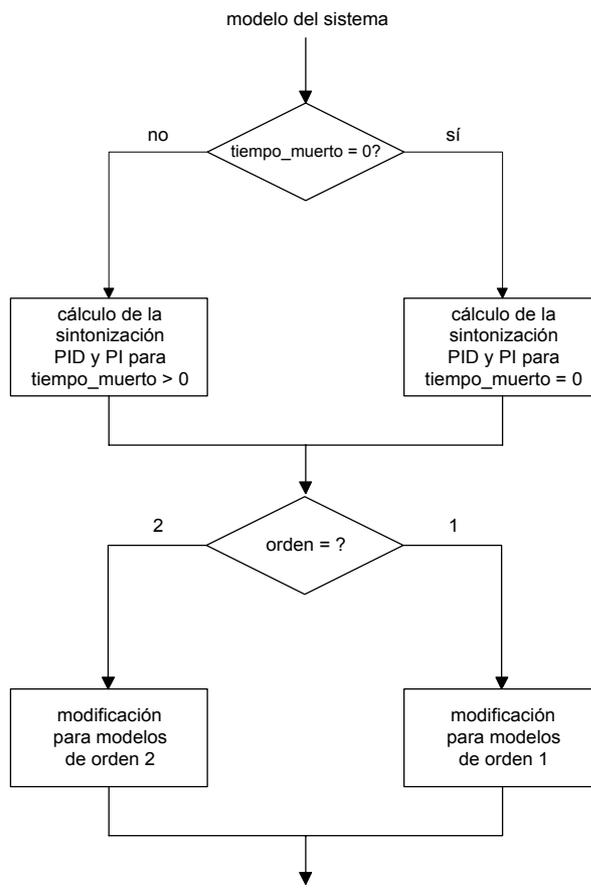
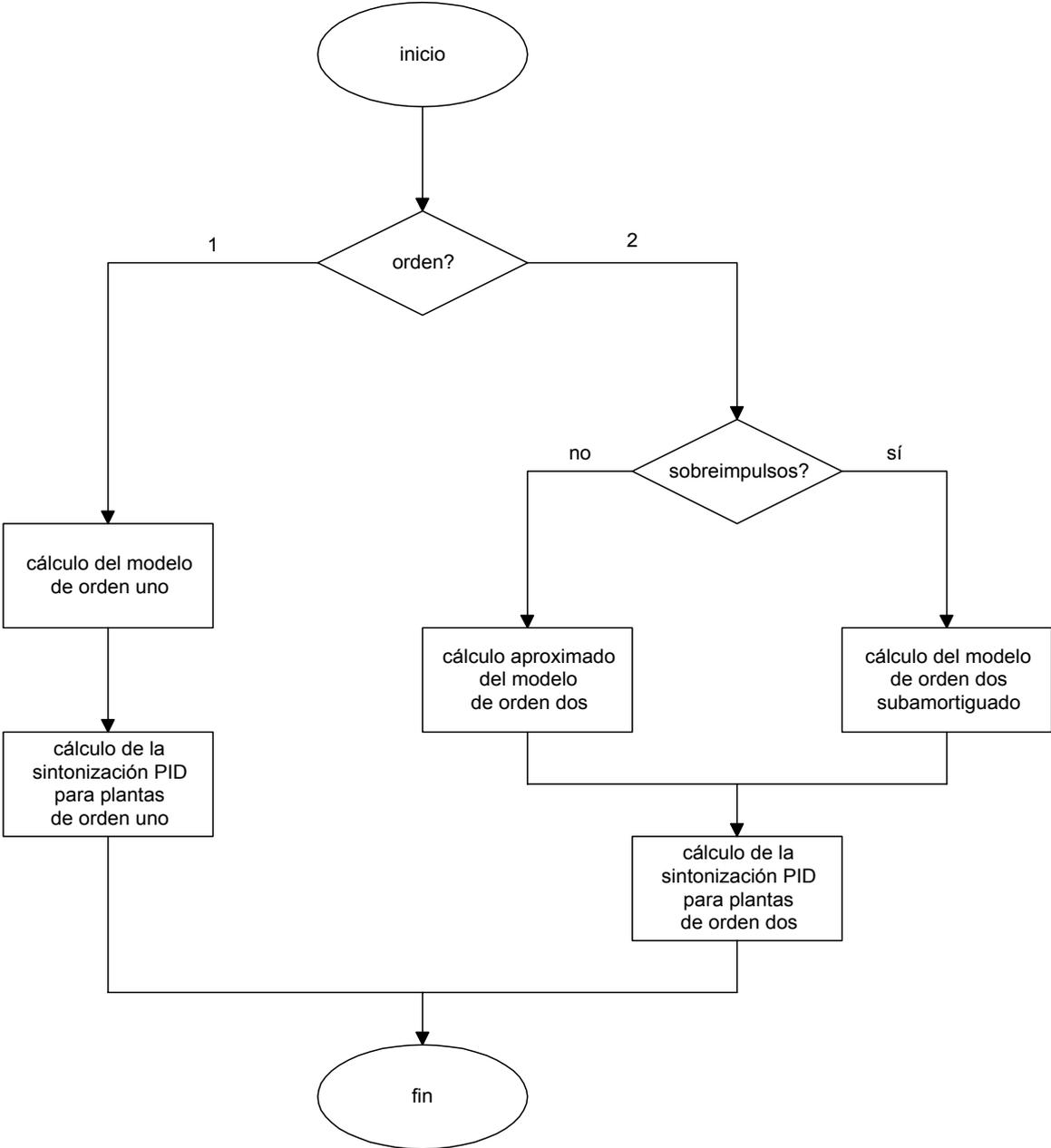


Figura 5.2 Diagrama de flujo de la solución para la sintonización PID

Por último, la figura 5.3 muestra la solución final seleccionada.



VISIO32

Figura 5.3 Diagrama de flujo de la solución final

5.2 Descripción del software

El software desarrollado en este proyecto es un objeto ActiveX que puede ser insertado en simulaciones creadas con el paquete computacional InTouch.

Las funciones que realiza el programa son encontrar los parámetros del modelo de sistemas de primer y segundo orden y estimar la sintonización PID para controlar estos procesos. Además, como elemento adicional se incorpora el cálculo de la sintonización de controladores PI.

Se crean diferentes propiedades públicas para los diferentes valores calculados por el programa, se mencionan a continuación para luego explicar cada un de ellas, estas son: la ganancia, la o las constantes de tiempo, el retardo del sistema, y las constantes proporcionales e integrales del PI y del PID, así como la constante derivativa del PID. También se establece una propiedad para el tiempo de muestreo de datos, una para la dirección y el nombre del archivo de texto, y otra para establecer el orden del modelo.

Las primeras cuatro propiedades corresponden a los parámetros del modelo, ya sea de primer orden o de segundo orden, y las otras tres consisten en los valores de las constantes de sintonización. El tiempo de muestreo establece cada cuánto se van a leer los valores de la entrada y la salida del sistema.

La propiedad que tiene como nombre `archivo_y_dirección` es utilizada para crear un archivo de texto en el cual se almacenan los datos leídos de la entrada y la salida, así como el momento en que se tomó cada dato. Por último existe una propiedad que le indica al programa si se quiere hacer una estimación de primer orden con los datos leídos o si se desea obtener un modelo de segundo orden.

Al establecer estas variables como propiedades del objeto ActiveX permite al usuario o programador acceder los valores por medio de la

asignación de éstos a variables de la simulación en la que se este usando, o para usar estos datos en otros programas.

Uno de los usos que se le puede dar a esto es asignar las constantes encontradas por el objeto a los valores de las constantes del PID de InControl para que este cambie automáticamente el valor de la sintonización.

El archivo de texto que crea el PIDX a partir de los datos leídos puede servir para un análisis posterior del comportamiento del proceso en estudio, así como para llevar un registro de las pruebas realizadas.

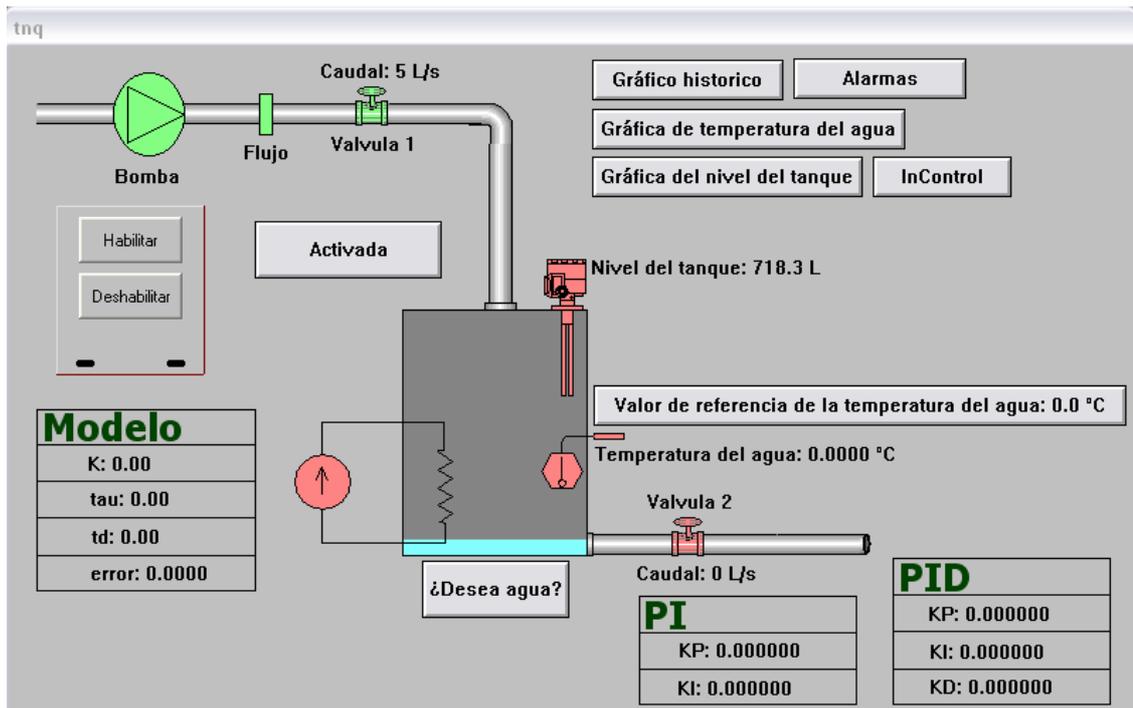
Capítulo 6: Análisis de Resultados

En primer lugar se explica la plataforma o entorno en el cual se va a utilizar la solución implementada. Esto va a ser mediante simulaciones desarrolladas en los programas InTouch e InControl, en los cuales se programan en un ambiente gráfico, las diferentes variables que determinan el comportamiento de diversos sistemas para los que se necesita un control PID.

Con el fin de probar el sistema durante las diferentes etapas del proceso de creación del objeto ActiveX, se hizo una simulación con un tanque de agua en donde el proceso a controlar era la temperatura del agua contenida en el tanque. La simulación sufre modificaciones conforme las pruebas a realizar lo van exigiendo, y va evolucionando hasta llegar a una versión final, cuyo funcionamiento se explica a continuación.

El tanque tiene capacidad para almacenar 10000 L de agua, más tiene un control de cierre de válvulas cuando llega a los 8000 L. La temperatura deseada para el agua es introducida por el operador y puede cambiarse cada vez que sea necesario. El agua se calienta por medio del flujo de corriente a través de una resistencia, esta corriente es la que debe ser controlada según la temperatura establecida por el operador. La ventana principal de la simulación se muestra en la siguiente figura.

Uno de los objetivos de esta simulación es que el comportamiento del proceso a controlar pueda variar, o sea, que el modelo que describe dicho proceso cambie. Lo anterior se logra a través de un objeto insertado en InControl en el cual se pueden introducir modelos de primer y segundo orden y asignarlos a la variable que va a presentar este comportamiento. De esta forma podemos simular diferentes sistemas haciendo uso de un único programa. La página de propiedades de este objeto se muestra en la figura 6.2.



INTOUCH

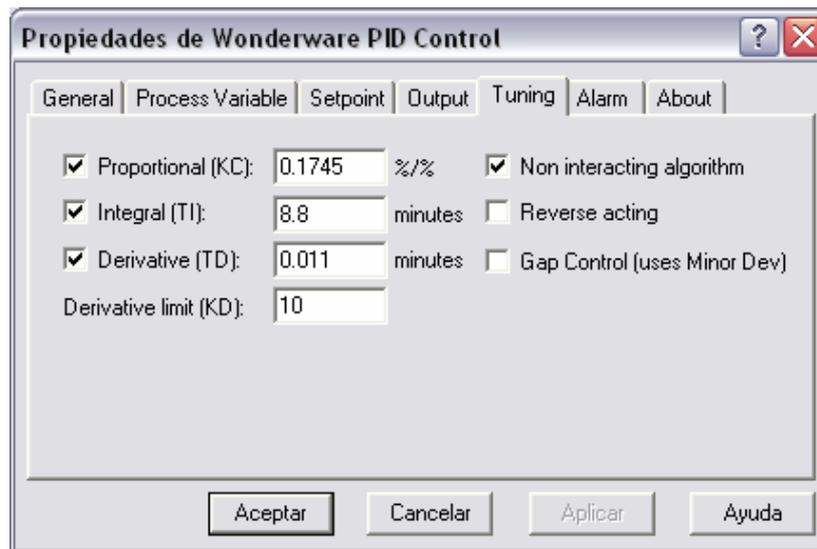
Figura 6.1 Ventana principal de la simulación en InTocuh



INCONTROL

Figura 6.2 Hoja de propiedades del objeto Lead Lag

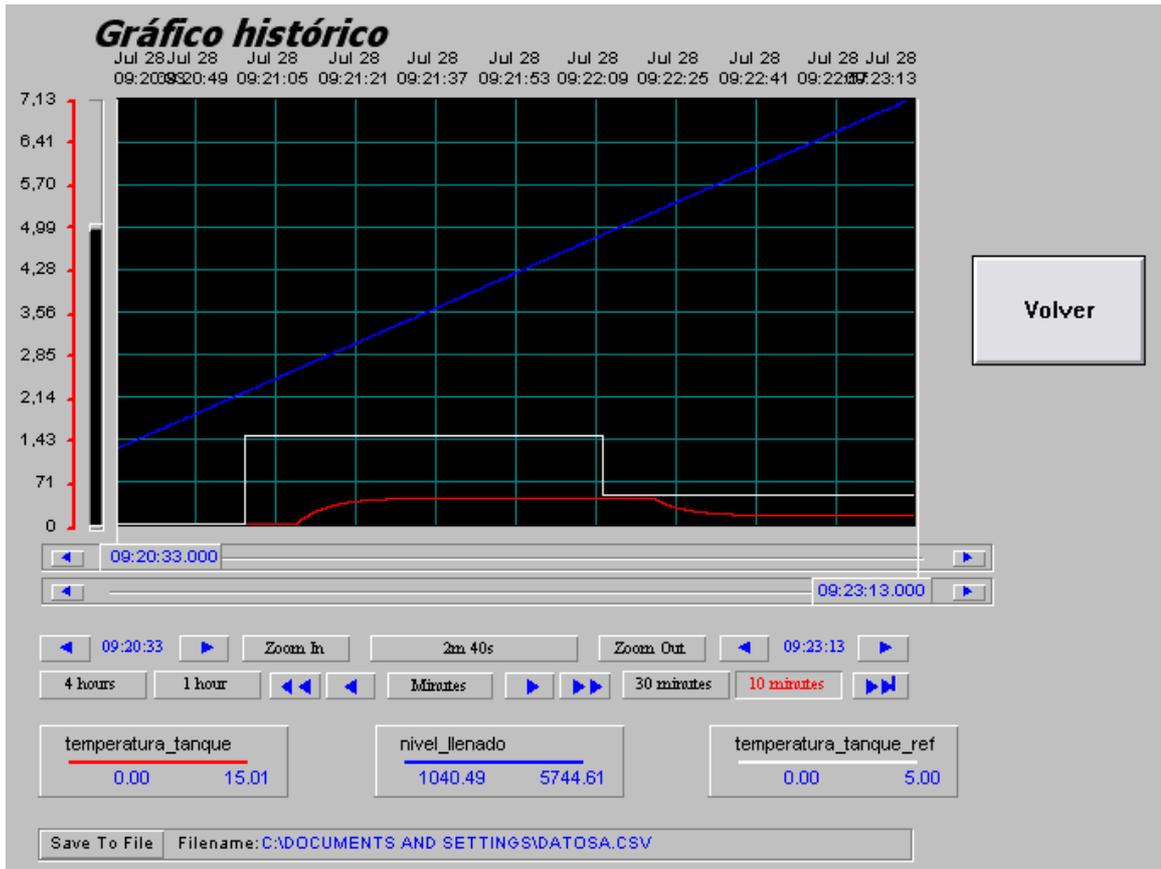
El otro objeto que se usa en InControl es un PID programable. Entre las variables más importantes están las constantes de sintonización proporcional, integral y derivativa, el tiempo de muestreo del PID, entre otras. Este objeto nos permite cambiar la sintonización de forma que sea efectiva según el modelo de planta que se haya establecido en el otro objeto, esto se hace en la página de propiedades de la figura 6.3.



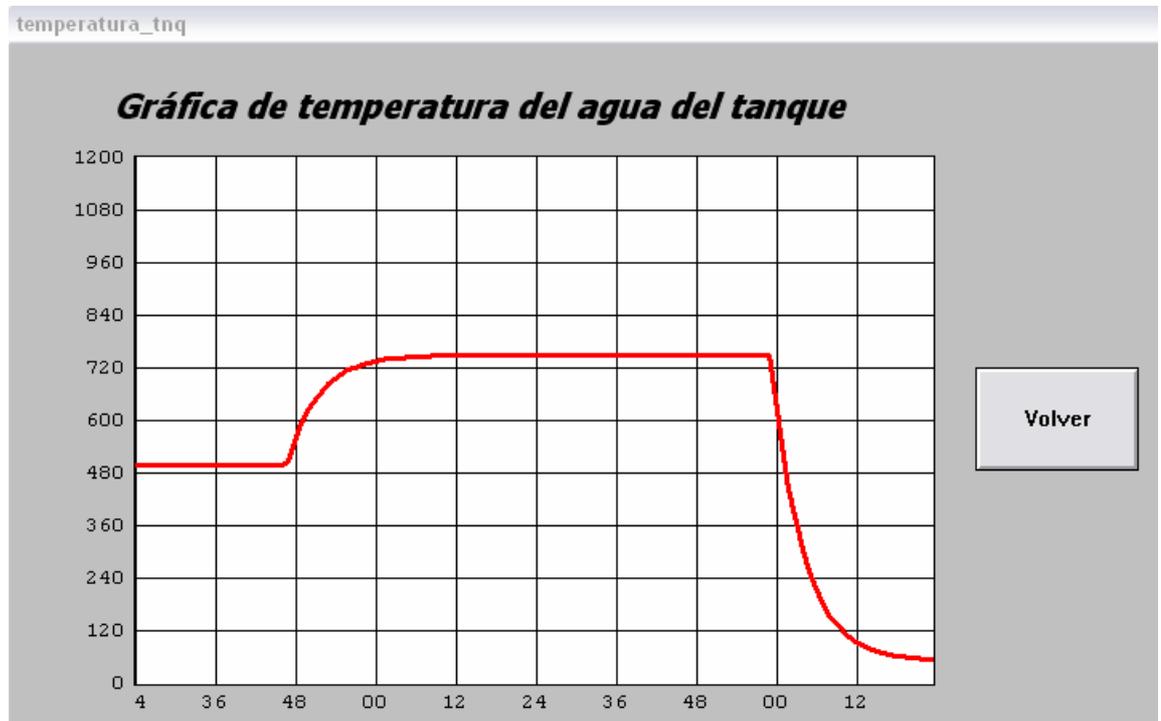
INCONTROL

Figura 6.3 Hoja de propiedades del objeto PID Control

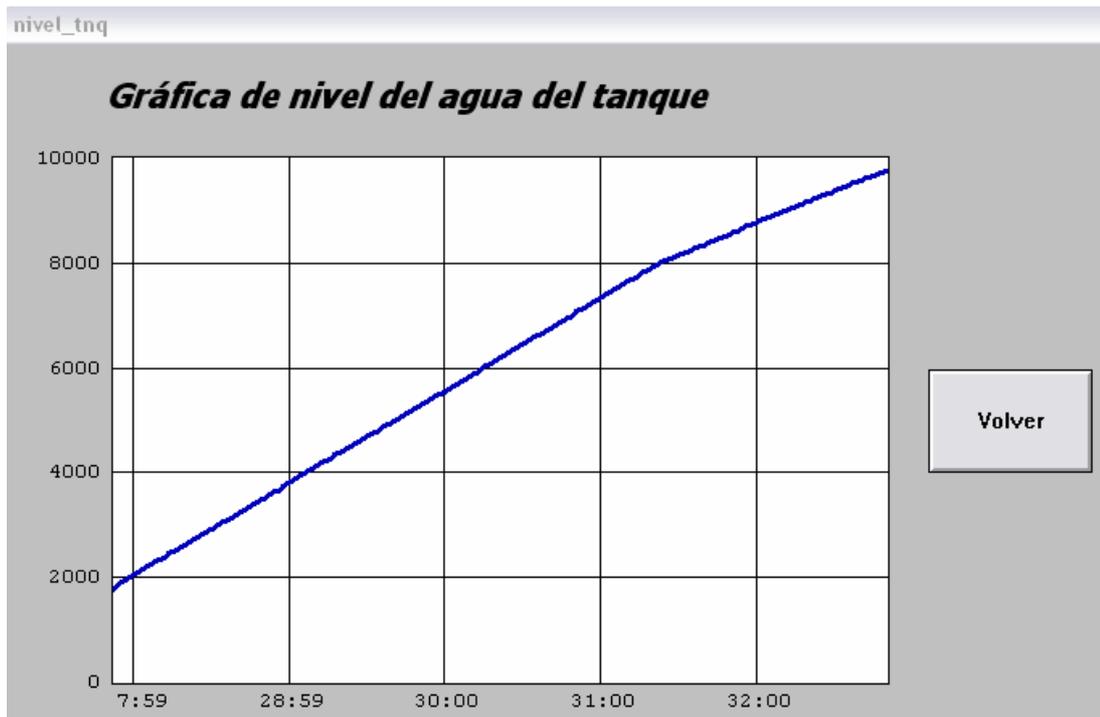
Adicionalmente se agregaron otras ventanas en InTouch además de la principal, las cuales son útiles para el análisis de resultados. Una de las ventanas muestra una gráfica de la temperatura del agua (figura 6.5), otra muestra una gráfica del nivel de agua contenido en el tanque (figura 6.6), una tercera ventana muestra si se activa alguna de las alarmas programadas y si ésta fue atendida (figura 6.7), otra ventana contiene un gráfico histórico de tres de las variables del sistema (figura 6.4), a saber la temperatura del agua, el nivel de agua en el tanque y la temperatura de referencia. Por último se cuenta con una ventana desde la cual se pueden modificar las diferentes variables de InControl, del modelo y del PID (figura 6.8).



INTOUCH
 Figura 6.4 Ventana gráfico histórico



INTOUCH
 Figura 6.5 Ventana gráfica de temperatura



INTOUCH
Figura 6.6 Ventana gráfica de nivel

alarmas

Grupo de alarmas

Time	State	Class	Type	Priority
07/12/2005 02:16...	UNACK	VALUE	LO	1
07/12/2005 02:16...	UNACK	VALUE	LO	1

Volver

Displaying 1 to 2 of 2 alarms. Default Query 100 % Complete

INTOUCH
Figura 6.7 Ventana alarmas

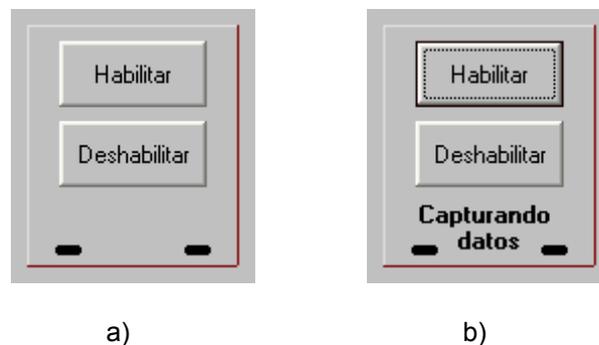


INTOUCH

Figura 6.8 Ventana de variables de InControl

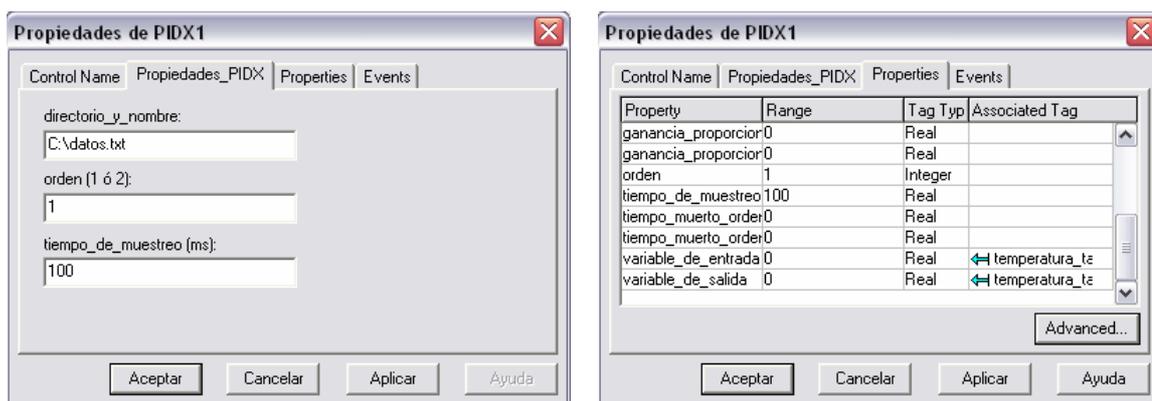
Un problema que se encontró con la forma en que se hizo la simulación es que el objeto en el cual se programa el modelo o comportamiento de la planta no maneja números complejos, esto tiene como consecuencia que no se simulen sistemas de segundo orden subamortiguados. Para simular procesos de este tipo es necesario programar un sistema cuya respuesta real presente oscilaciones.

Pasando a lo que es la aplicación desarrollada en Visual Basic, lo que se tiene es un objeto ActiveX capaz de estimar el modelo de plantas de primer y segundo orden, así como de proponer la sintonización PID para el control de dicho modelo. La figura 6.9 muestra el aspecto del PIDX, mientras que la figura 6.10 corresponde a las páginas de propiedades del PIDX.



VISUAL BASIC

Figura 6.9 Aspecto del PIDX a) sin habilitar b) habilitado



VISUAL BASIC

Figura 6.10 Páginas de propiedades del PIDX

El algoritmo utilizado para proponer el modelo de plantas de primer orden con tiempo muerto es el expuesto en [1], y como se puede apreciar en la tabla 6.1 el error máximo encontrado es de un 13,29 % para la planta d). Se hicieron numerosas pruebas para comprobar el correcto funcionamiento del programa, así como del algoritmo, algunas de ellas se muestran en esta tabla.

Tabla 6.1 Valores teóricos y valores calculados por el PIDX para cuatro plantas de primer orden

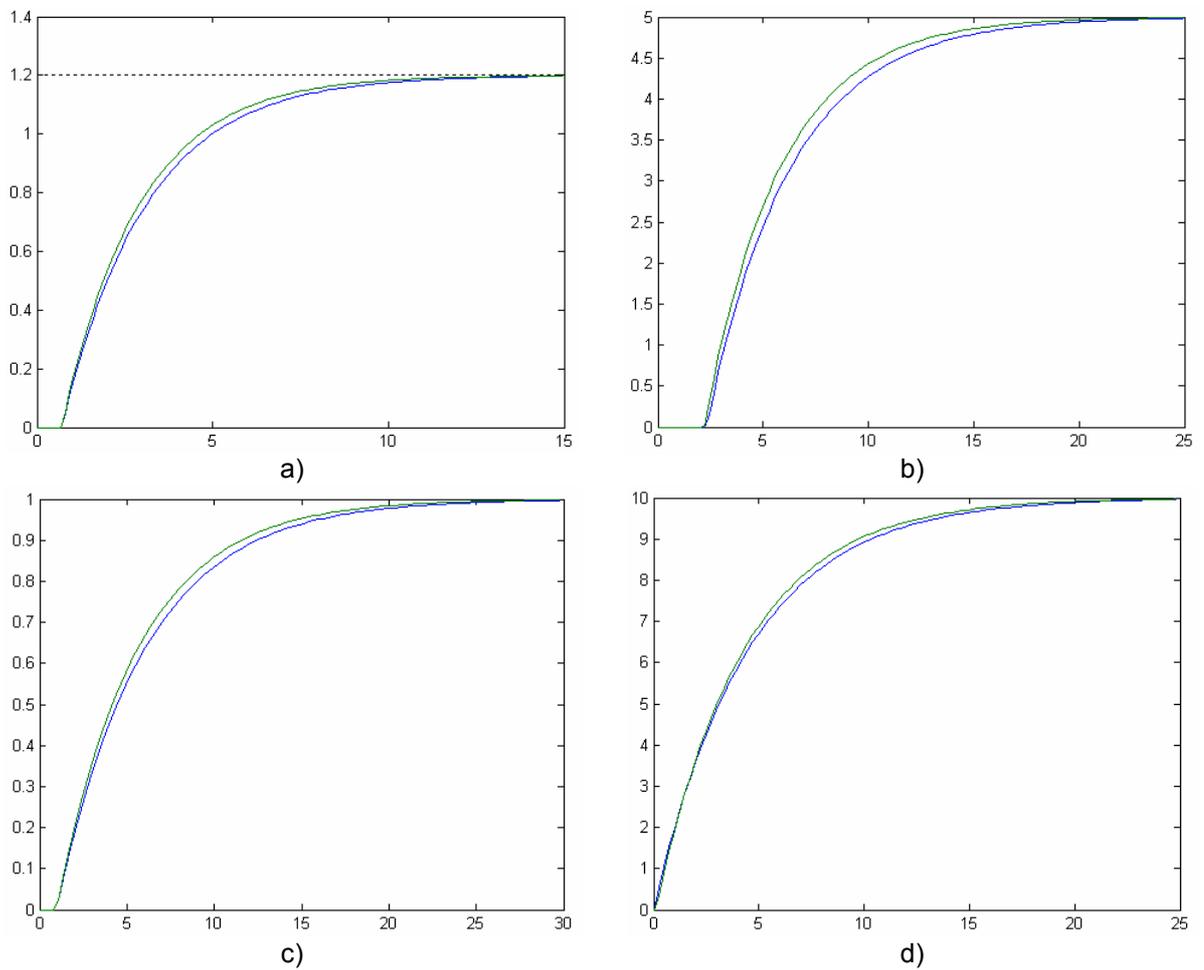
planta / parámetro	a		b		c		d	
	teórico	PIDX	teórico	PIDX	teórico	PIDX	teórico	PIDX
ganancia	1.2	1.2	5	5	1	1	10	10
constante de tiempo	2.4	2.2	4	3.6	5	4.6	4.5	4.2
tiempo muerto	0.7	0.7	2.3	2.2	1	1	0	0.1
error	0	0.0447	0	0.153	0	0.0242	0	0.1329

Se puede observar que no existe error en el cálculo de la ganancia del sistema, y que con los otros dos parámetros, la constante de tiempo y el retardo, el error es mínimo. El error que se calcula aquí es el error que presenta la curva de respuesta escalón del modelo encontrado respecto a la curva de respuesta del modelo teórico, para esto se utiliza el error estándar.

El error estándar se calcula a partir de la desviación estándar la cual es una sumatoria de las diferencias entre las muestras teóricas y las muestras

experimentales. Se determina que si el error estándar resulta en un valor pequeño, significa que la diferencia entre los puntos encontrados de los puntos teóricos es pequeña. Si el error es mínimo significa que el ajuste entre curvas es máximo y que el modelo estimado describe el comportamiento real de la planta.

En las curvas de la a) a la d) de la figura 6.11 se muestra la curva de respuesta al escalón teórica en azul, junto con la curva de repuesta del modelo encontrado por el programa en verde. Es notoria la similitud entre una y otra, verificando así que la estimación hecha por el programa es bastante aceptable. Las curvas mostradas en esta figura corresponden a los modelos de la tabla 6.1.



MATLAB

Figura 6.11 Gráficas de respuesta al escalón de las plantas de primer orden a, b, c y d de la tabla 6.1

La estimación del modelo de primer orden es uno de los resultados más importantes de este proyecto, ya que la mayoría de los procesos industriales se pueden describir y analizar mediante un modelo de este tipo.

Para encontrar el modelo de plantas de segundo orden, se divide la solución en dos grandes partes, la primera es la parte que contempla a todos los sistemas que presentan sobreimpulsos en su curva de respuesta al escalón, o sea, los procesos subamortiguados. La segunda gran parte contempla a los sistemas críticamente amortiguados y sobreamortiguados; se discutirá cada una por separado.

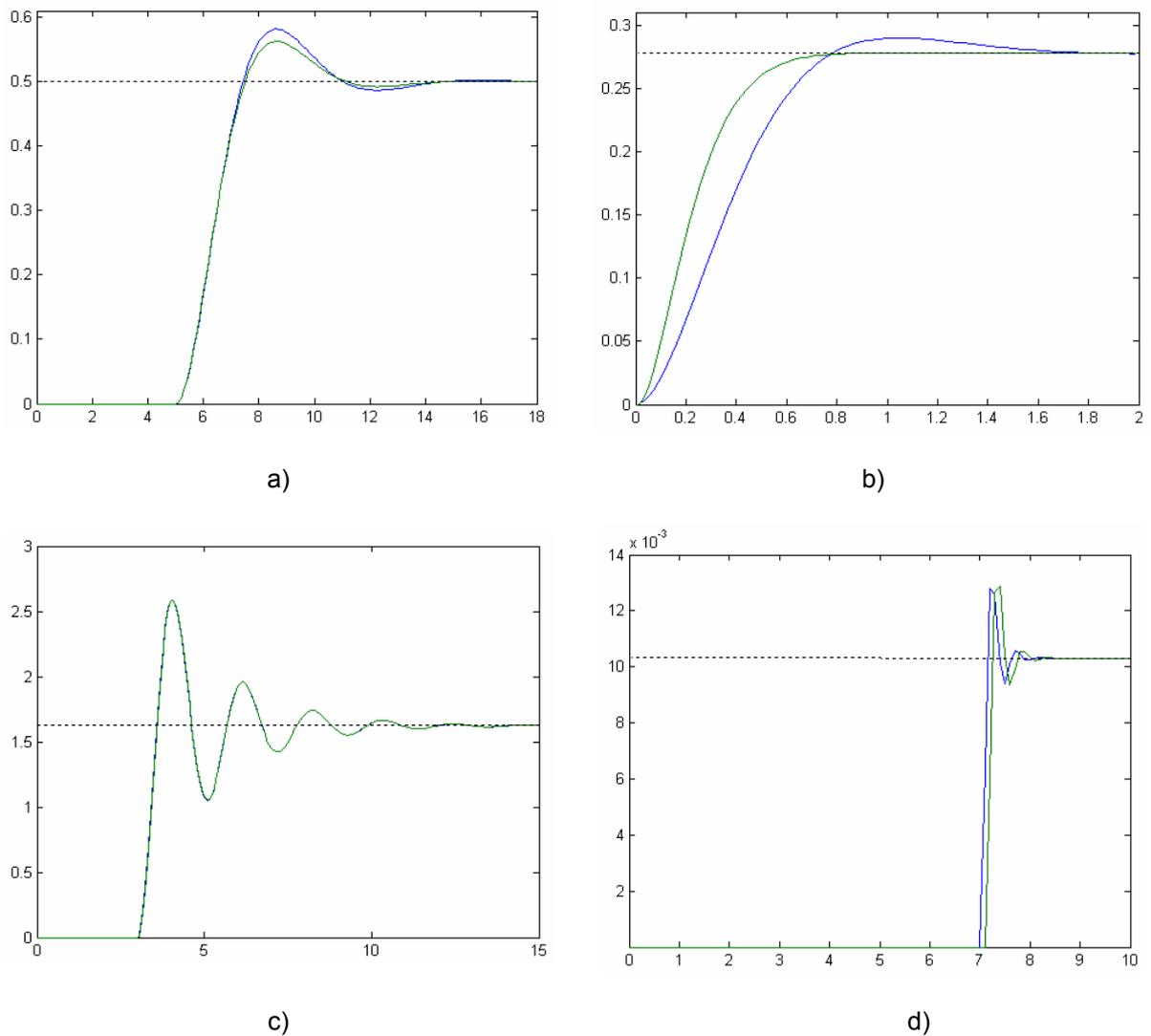
Los procesos que presentan un comportamiento subamortiguado son fáciles de reconocer gracias a que en la parte transitoria de su curva de respuesta al escalón sobrepasan al menos una vez su valor final o el valor que alcanzan en estado estacionario. De esta forma se pudo separar este tipo de sistemas en un grupo aparte, y emplear una solución específica para hallar el modelo.

El algoritmo que fue programado en el PIDX, es el propuesto por Bollinger [1]. La razón por la cual estos procesos presentan estas oscilaciones en su curva de respuesta es porque tienen polos complejos, también llamados raíces complejas, a diferencia de los otros tipos de plantas cuyos polos tienen solamente valores reales.

En la tabla 6.2 se presentan cuatro resultados de las muchas pruebas que se hicieron en esta parte; en la figura 6.12 se muestran tanto la curva teórica (azul) como la curva obtenida con el programa (verde). También se calculó el error para cada uno de los modelos de la tabla, ninguno sobrepasa el 4 %, siendo éste más exacto que el cálculo del modelo de primer orden.

Tabla 6.2 Valores teóricos y valores calculados por el PIDX para cuatro plantas de segundo orden subamortiguadas

planta / parámetro	a		b		c		d	
	teórico	PIDX	teórico	PIDX	teórico	PIDX	teórico	PIDX
ganancia	1	1	5	5	15	15	2	2
a	2	1.85	1	0.2734	1	0.996	1	1.076
b	2	2.126	6	4.122	1	0.9978	10	10.17
c	2	2	18	18	9.25	9.248	194	194
tiempo muerto	5	5.01	0	0.01	3	3.01	7	7.1
error	0	0.0348	0	0.0111	0	0.0352	0	9.78E-05



MATLAB

Figura 6.12 Gráficas de respuesta al escalón de las plantas de segundo orden subamortiguadas a, b, c y d de la tabla 6.2

Este método no propone ninguna solución para el cálculo del retardo del sistema, por lo que se empleó el descrito en [3]. Este último es sencillo y muy exacto, y utiliza estrictamente la definición de tiempo muerto, o sea, la diferencia de tiempo entre el momento en que se produce un cambio en la entrada y el momento en que se produce un cambio a la salida.

Respecto de los sistemas sobreamortiguados y críticamente amortiguados lo que se hace es una estimación de un modelo de primer orden. El programa no

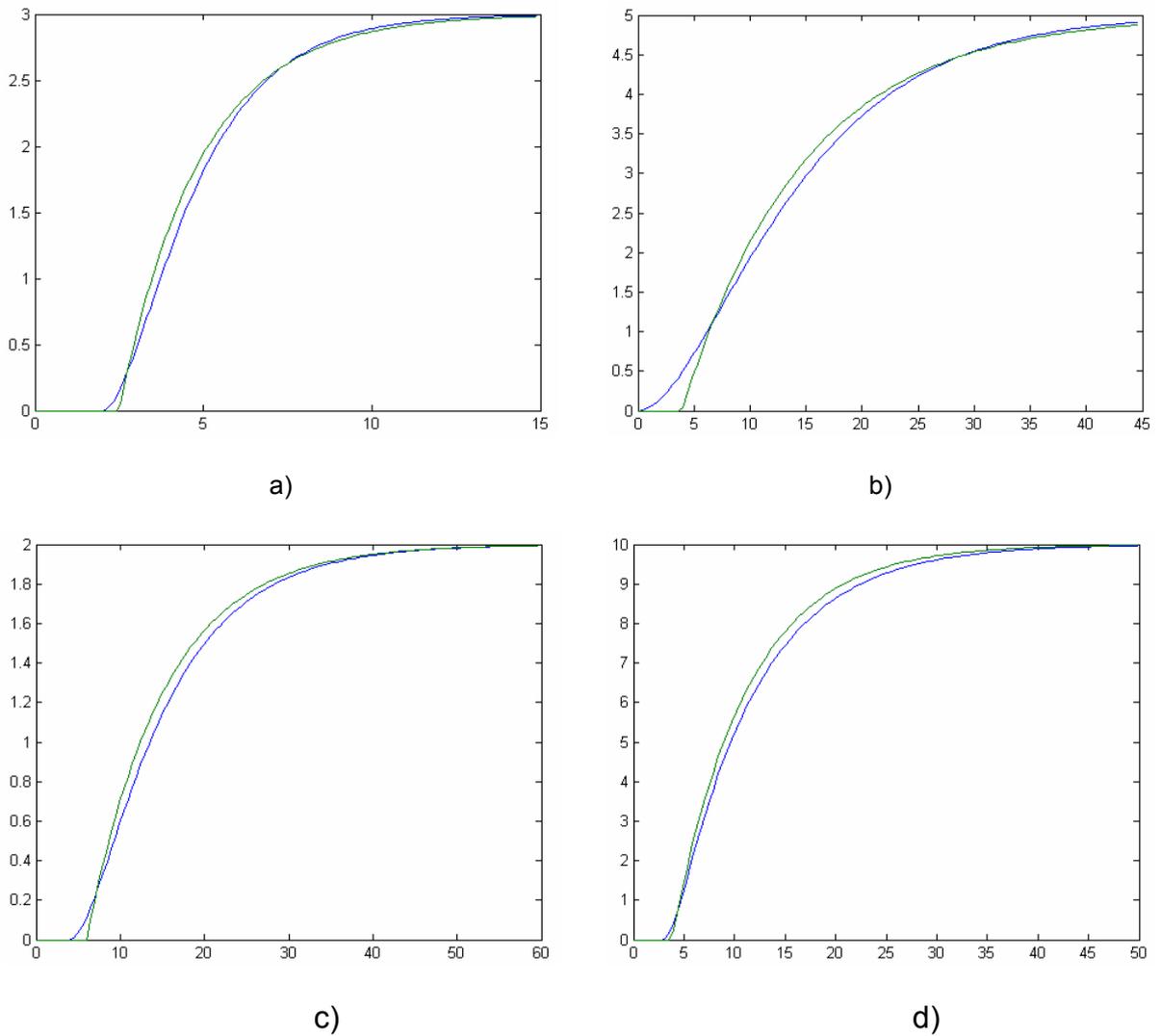
discrimina entre sistemas sobreamortiguados y críticamente amortiguados, más los resultados obtenidos si pueden ser analizados en forma separada.

Para los sistemas sobreamortiguados se conoce que la constante de tiempo de mayor valor es la que domina el comportamiento o la “forma” de la curva en el lapso transitorio de ésta, por lo que al estimar el modelo con una sola constante de tiempo, el resultado se aproxima a la constante de mayor valor.

El efecto de la otra constante de tiempo se compensa aumentando el tiempo muerto de la planta. Se puede observar en la tabla 6.3 que todos los retardos calculados por el programa son mayores a los del modelo teórico y en las curvas de respuesta correspondientes de la figura 6.13, se ve como al aumentar el tiempo muerto del sistema la curva encontrada se acerca aún más a la curva de respuesta teórica, y el error punto a punto disminuye.

Tabla 6.3 Valores teóricos y valores calculados por el PIDX para cuatro plantas de segundo orden sobreamortiguadas

planta / parámetro	a		b		c		d	
	teórico	PIDX	teórico	PIDX	teórico	PIDX	teórico	PIDX
ganancia	3	3	5	5	2	2	10	10
constante de tiempo1	1	2.4	8	11	9	9.2	8	7.4
constante de tiempo2	2	---	7	---	3	---	1	---
tiempo muerto	2	2.5	0	3.9	4	6	3	3.8
error	0	0.1254	0	0.0226	0	0.0376	0	0.0798



MATLAB

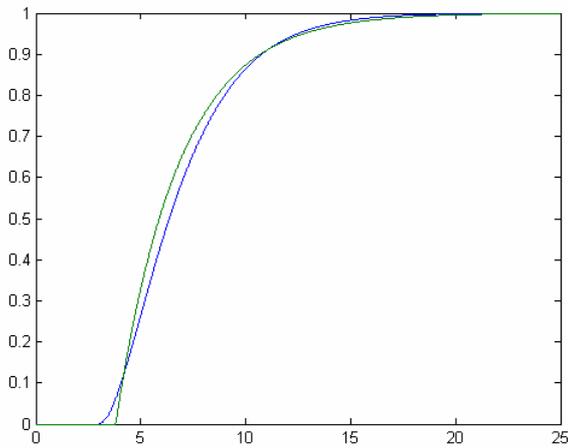
Figura 6.13 Gráficas de respuesta al escalón de las plantas de segundo orden sobreamortiguadas a, b, c y d de la tabla 6.3

Se sabe que en los sistemas críticamente amortiguados las dos constantes de tiempo son iguales, o sea, tienen el mismo valor. Los cálculos que se hacen a partir de los datos obtenidos de la curva de respuesta al escalón son los mismos que en el caso anterior.

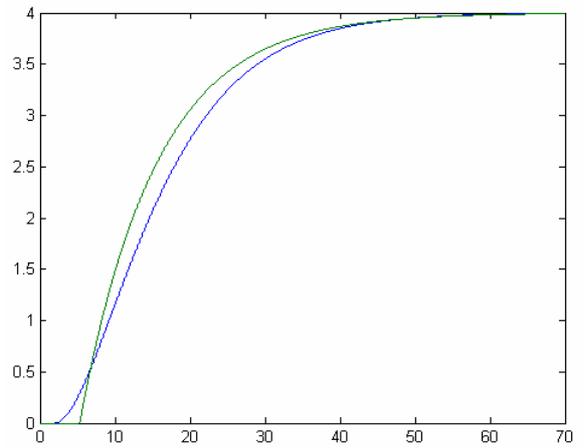
En la tabla 6.4 se puede ver que el valor de la constante de tiempo calculada por el programa es en todos los casos un poco mayor que el valor de las constantes teóricas, de esta forma se aproxima más al comportamiento real de la planta; el retardo calculado es mayor al teórico para reducir el error y obtener una mejor aproximación como se hizo con los procesos sobreamortiguados.

Tabla 6.4 Valores teóricos y valores calculados por el PIDX para cuatro plantas de segundo orden críticamente amortiguadas

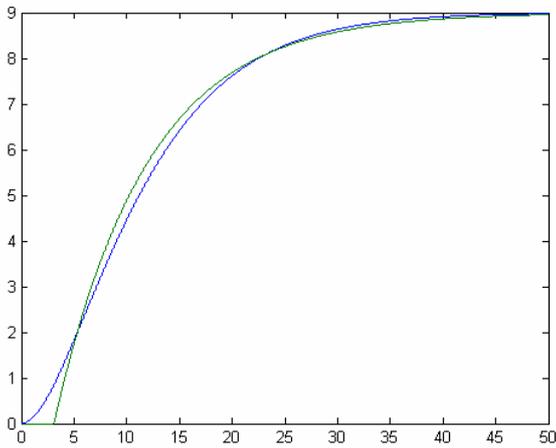
planta / parámetro	a		b		c		d	
	teórico	PIDX	teórico	PIDX	teórico	PIDX	teórico	PIDX
ganancia	1	1	4	4	9	9	3	3
constante de tiempo1	2	3	7.5	10.2	6	8.8	4.2	6
constante de tiempo2	2	---	7.5	---	6	---	4.2	---
tiempo muerto	3	3.8	2	5.3	0	3.1	8	9.7
error	0	0.0978	0	0.0291	0	0.0551	0	0.4675



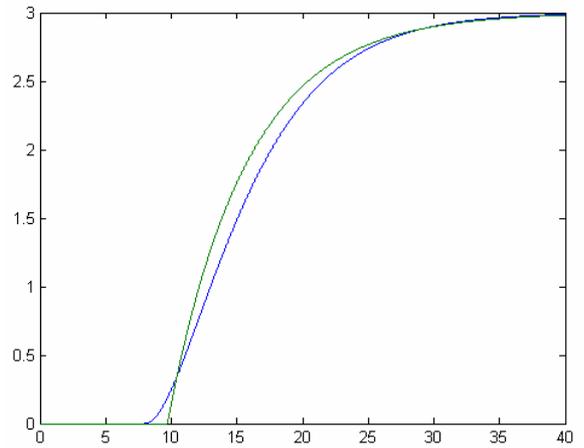
a)



b)



c)



d)

MATLAB

Figura 6.14 Gráficas de respuesta al escalón de las plantas de segundo orden críticamente amortiguadas a, b, c y d de la tabla 6.4

El PIDX además, calcula el tiempo muerto teórico de la planta, esto lo hace usando el mismo algoritmo para todos los sistemas de segundo orden, ya que este se adaptaba para todos los casos y fue con el que se obtuvo mejores resultados pese a que está planteado en un método para la estimación de modelos de primer orden.

La parte del programa que se encarga de la estimación del modelo del sistema utiliza en su gran mayoría la teoría y los algoritmos de [1]. Se pudo observar que pese a que en algunos casos se utilizan aproximaciones, al obtener la respuesta al escalón de estas estimaciones, el ajuste entre curvas es el esperado, lo que indica que el modelo encontrado tiene el mismo comportamiento que el modelo teórico. La figura 6.15 muestra el algoritmo para el cálculo del modelo de la planta.

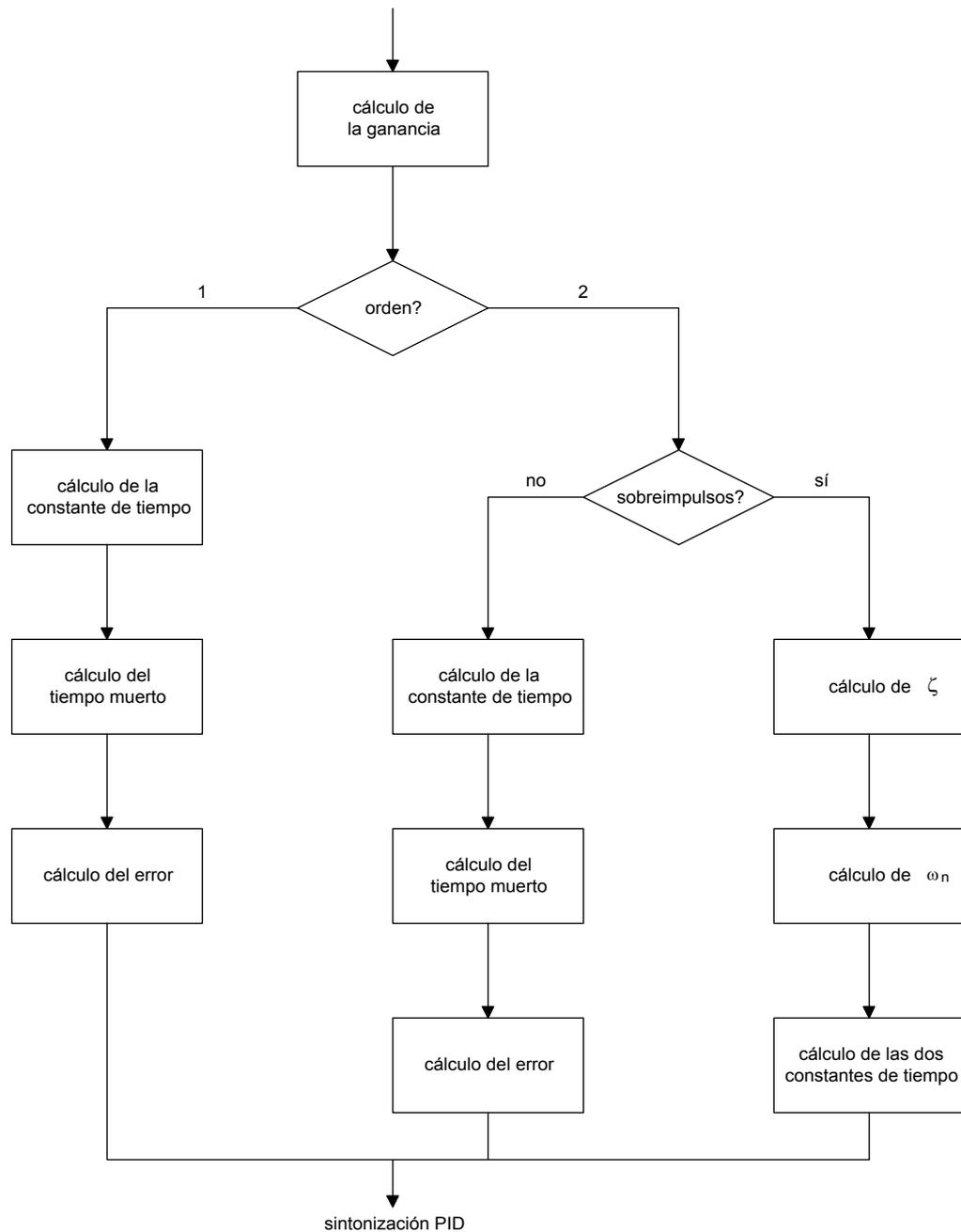
El error con el que se ha venido trabajando, es determinado por el programa. La justificación de que el PIDX calcule por si mismo este valor es porque cuando éste se use en simulaciones de sistemas reales no será posible conocer el modelo teórico de la planta con anterioridad, por lo que es útil saber que tan buena es una estimación para así ser considerada.

El error calculado es el error estándar entre las dos curvas y con éste podemos determinar el grado de ajuste entre la curva teórica y la curva experimental, que en este caso serían la curva medida y la curva calculada por el programa; entre más bajo sea el error se tendrá un mayor ajuste ente curvas.

El cálculo de la sintonización PID se basa en el método de Ziegler – Nichols a lazo abierto, el cual se modifica con el fin de adaptarlo a las necesidades del proyecto. Se quiere que la solución final sea capaz de sintonizar sistemas de respuesta rápida y además se quiere que la sintonización corresponda a la sintonización del PID que se usa en InControl.

El primer problema se da gracias a que el método de Ziegler – Nichols se creó con el fin de controlar plantas cuya respuesta al escalón sea lenta. Para resolver esto se calculó el valor de la constante de controlabilidad de las diferentes plantas de respuesta rápida, como lo son las todas aquellas que

presentan constantes de tiempo pequeñas, normalmente menores a uno, y todas aquellas cuyo tiempo muerto es igual a cero o cercano a este valor.



VISIO32

Figura 6.15 Diagrama de flujo del algoritmo utilizado para el cálculo de los parámetros del modelo de plantas de primer y segundo orden

Se encontró que el valor de α en estos casos es sumamente pequeño, de centésimas de unidad. Para estos casos un valor alto de la constante derivativa del PID es contraproducente debido a que provoca un sobreimpulso inicial de gran valor, junto a un comportamiento oscilatorio. Para disminuir estos efectos se disminuye la ganancia del controlador y se aumenta la constante integral del mismo.

Esto último es lo que se hace para controlar el mismo sistema con un PI, sin embargo, la constante derivativa aporta una respuesta más rápida del sistema. A partir de este punto se procede a calcular tanto la sintonización para PID como para PI, así el interesado tiene la opción de usar ambas respuestas y escoger la de mayor conveniencia en cada caso particular.

Para adaptar el método de Ziegler –Nichols a las ecuaciones utilizadas por el objeto PID de InControl se procede a introducir las constantes determinadas por dicho método en el objeto, esto se realiza para diferentes modelos de plantas. Luego se procede a cambiar los valores, a prueba y error, hasta alcanzar la sintonización adecuada. Se determina que en InControl el efecto de la constante derivativa se ve aumentado en un 200% en la mayoría de los casos.

Una vez resueltos estos problemas se procede a incorporar las soluciones encontradas al objeto ActiveX. En la tabla 6.6 se muestran los resultados de tres pruebas hechas a tres modelos de plantas, dos de primer orden y una de segundo orden; en la tabla 6.5 se tienen los valores teóricos y calculados de las plantas a, b y c de la tabla 6.6. Se observa que el cambio entre el método de Ziegler –Nichols y el método adaptado del PIDX se encuentra en el valor de la constante de tiempo derivativa.

Tabla 6.5 Datos teóricos contra datos calculados por el PIDX para el modelo de la planta

planta / parámetro		a		b		c	
		teórico	PIDX	teórico	PIDX	teórico	PIDX
modelo	ganancia	1.2	1.2	5	5	5	5
	constante de tiempo 1	2.4	2.2	4	3.8	8	10.8
	constante de tiempo 2					7	
	tiempo muerto	0.7	0.7	2.3	2	0	3.4
	error	0	0.0447	0	0.1054	0	0.0447

Tabla 6.6 Datos teóricos contra datos calculados por el PIDX para la sintonización PID y PI de las plantas de la tabla

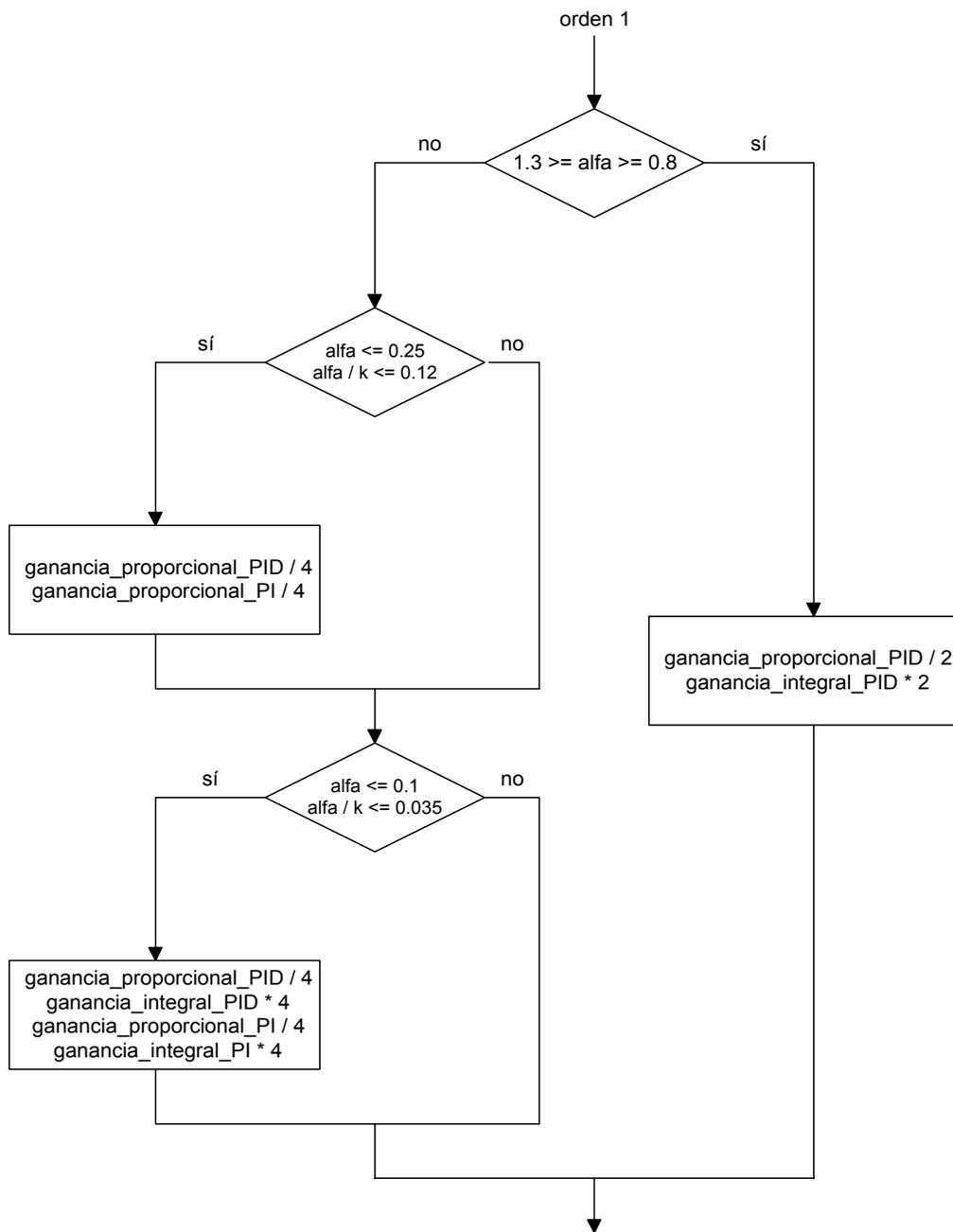
planta / parámetro		a		b		c	
		teórico*	PIDX	teórico*	PIDX	teórico*	PIDX
	α		0.318181		0.526315		0.324561
	α / k		0.265151		0.105263		0.064912
PID	ganancia proporcional	3.142857	3.142864	0.456	0.456001	0.762352	0.762354
	ganancia integral	1.4	1.4	4.6	4	0	6.8
	ganancia derivativa	0.35	0.00175	1.15	0.005	0	0.0085
PI	ganancia proporcional	2.357142	2.357148	0.342	0.342001	0.571764	0.571766
	ganancia integral	2.331	2.331	7.659	6.66	0	11.322

* Los valores teóricos de PID y del PI corresponden a los valores del método de Ziegler- Nichols.

Los valores de la constante de controlabilidad α son pequeños, sin embargo no lo suficiente como para que sea necesario aplicar ningún otro cambio significativo en los valores de las constantes de PID o del PI. El valor de alfa dividido entre la ganancia se incluye en la tabla debido a que el valor de la ganancia puede ser un factor determinante en el comportamiento de un sistema ante el controlador.

El diagrama de flujo de los algoritmos creados para la sintonización PID y PI para primer y segundo orden se muestran en las figuras 6.16 y 6.17 respectivamente. En estas figuras se observa la importancia del valor de alfa y de la ganancia de la planta en el cálculo de las constantes de sintonización.

En la figura 6.19 se muestran las curvas correspondientes a las repuestas de los sistemas de la tabla 6.5 controlados por un PID cuya sintonización es estimada por el PIDX (tabla 6.6). Es importante recalcar que en la simulación tanto las curvas de respuesta como la salida del PID tienen un comportamiento análogo y cuando se implementan en dispositivos como lo son los PLCs, las salidas de éstos son discretas, por lo que los valores deseados se alcanzan más rápidamente ya que el incremento o decremento en los valores es de tipo escalonado, en otras palabras, el tiempo de asentamiento disminuye.



VISIO32

Figura 6.16 Algoritmo para la sintonización PID y PI para modelos de orden uno

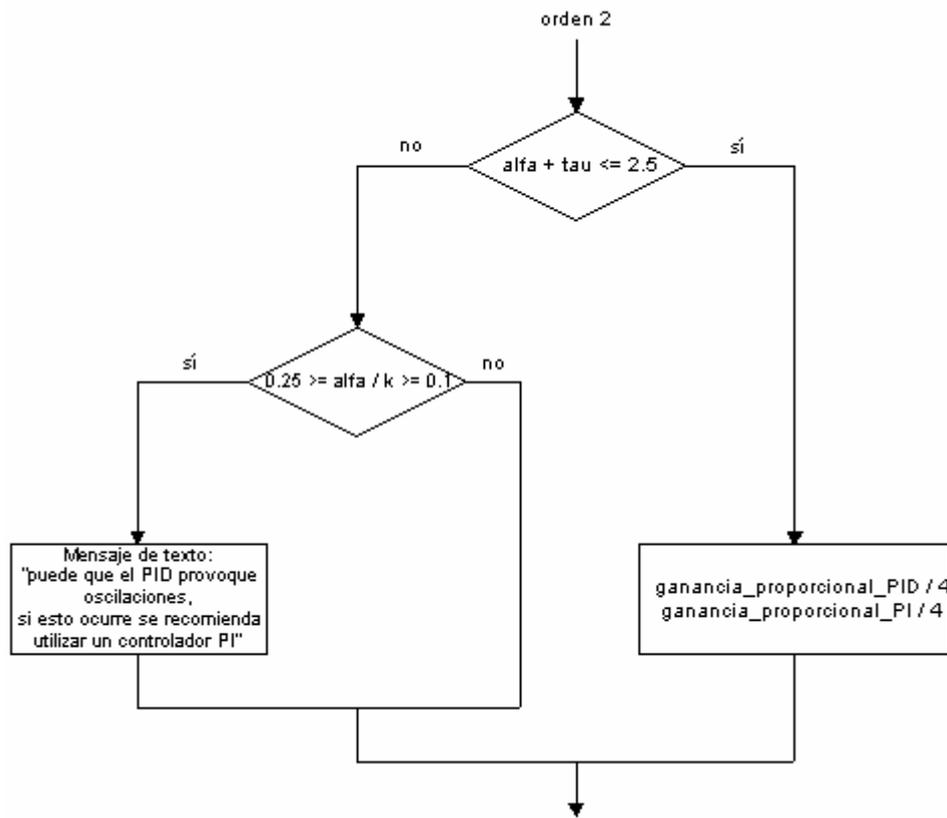


Figura 6.17 Algoritmo para la sintonización PID y PI para modelos de orden dos

Como parte del análisis se obtiene la respuesta del PIDX cuando se toman los datos de la planta en presencia de ruido. Para esto se uso el programa Matlab y a la señal de respuesta al escalón se sumo la señal de ruido que se muestra en la figura 6.18.

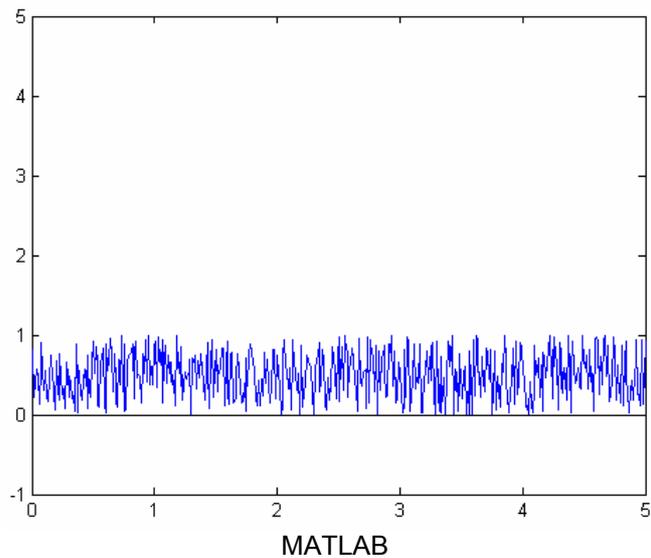
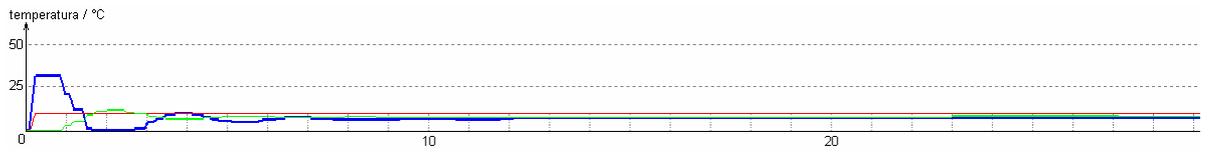
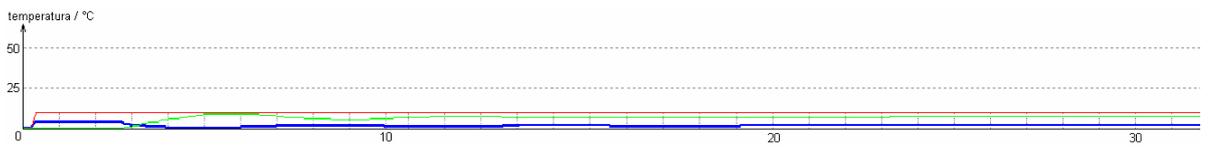


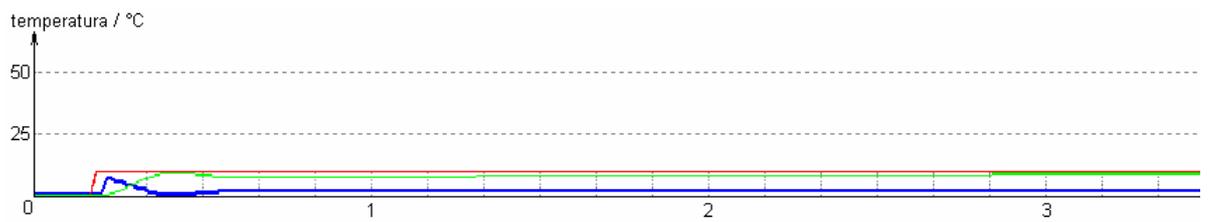
Figura 6.18 Señal de ruido generada en el programa Matlab



a)



b)



c)

INCONTROL

Figura 6.19 Curvas de respuesta de las plantas a, b y c de la tabla 6.5 controladas por los valores de sintonización PID de la tabla 6.6

Se calculan los parámetros de los modelos de las plantas de primer orden con tiempo muerto de la tabla 6.7 con ruido, los resultados se encuentran en la misma tabla. Lo que se hizo es utilizar los mismos valores para tiempo muerto y para la constante de tiempo y variar el valor de la ganancia de la planta, para encontrar la relación del error en los datos obtenidos según el porcentaje de la magnitud del ruido vs la magnitud o ganancia de la planta.

Tabla 6.7 Valor teórico y calculados de una planta de orden uno en presencia de ruido

parámetros	teórico	PIDX			
ganancia	10	9.993	9.926	9.258	2.579
constante de tiempo	2.4	2.4	2.36	2.32	0.22
tiempo muerto	0.7	0.69	0.71	0.93	0
porcentaje de ruido	0	0.1	1	10	100

Se encuentra que si la magnitud máxima del ruido es cercana o igual a la ganancia de la planta, como en el caso a de la tabla 6.7, el error es mayor al 100 %, y el modelo encontrado no coincide con el modelo teórico de la planta. En los otros casos b, c y d se observa que el error disminuye y que el modelo encontrado define correctamente el comportamiento de la planta en estudio. En estos últimos tres ejemplos, la amplitud máxima del ruido no sobrepasa el 110 % de la ganancia de la planta, y en la planta d se tiene el menor error aunque la diferencia entre este y los otros dos es mínima.

Capítulo 7: Conclusiones y recomendaciones

7.1 Conclusiones

1. El tiempo de solución de los proyectos de control de procesos se reduce de días a horas, al contar con un programa que determina la sintonización PID para dicho proceso.
2. El comportamiento de plantas de segundo orden sobreamortiguadas y críticamente amortiguados se puede describir usando un modelo de primer orden con tiempo muerto.
3. El método de Ziegler – Nichols a lazo abierto para sintonización PID se puede usar para controlar procesos con tiempo muerto cero disminuyendo el valor de la constante proporcional y aumentando el valor de la constante integral del PID.
4. La constante derivativa del PID provoca oscilaciones continuas para plantas con respuesta rápida, esto es, con tiempo muerto cero o cercano a cero y con constantes de tiempo menores a 1.
5. El error introducido en la aproximación del modelo de la planta no perjudica la estimación de una correcta sintonización PID.
6. Las plantas de orden uno cuya constante de controlabilidad es menor a 0.5 no pueden ser controladas mediante la sintonización de Ziegler-Nichols.
7. Los procesos de segundo orden que presentan constantes de controlabilidad menores a 2.5 no pueden controlarse con el método de Ziegler – Nichols de lazo abierto.
8. Al disponer de la sintonización de dos controladores diferentes, PI y PID, se puede escoger la mejor solución para los diferentes procesos.

9. El registro de datos proporcionados por el PIDX crea la posibilidad de aprovechar el trabajo realizado en caso de ajustes o mejoras futuras al sistema.
10. El conocimiento del modelo del proceso a controlar permite analizarlo aplicando los conocimientos de la teoría de control.
11. El PIDX se limita a encontrar el modelo de la planta y la sintonización PID de procesos de primer y segundo orden.
12. El uso del PIDX en simulaciones de sistemas en InTouch e InControl crea la sintonización automática del PID.
13. Las propiedades públicas del PIDX permiten la invención de múltiples aplicaciones y usos de este programa.
14. La sintonización PI y PID de plantas de primer y segundo orden son diferentes, pese a que estas últimas se puedan aproximar a modelos de primer orden con tiempo muerto.
15. El PIDX encuentra los parámetros de plantas de orden uno con tiempo muerto en presencia de ruido, si este no sobrepasa el 10% del valor de la ganancia de la planta.

7.2 Recomendaciones

1. Para encontrar el modelo de plantas sobreamortiguadas se hace una aproximación al modelo de plantas de orden uno, sin embargo, no debe de intentarse usar un modelo de primer orden para encontrar la sintonización PID, ya que esta es diferente para las plantas de orden dos.
2. Cuando el modelo de la planta coincida con un modelo de respuesta rápida es importante que se prueben la respuesta del sistema con el controlador PI para compara si es mejor que la que proporciona el PID.

3. Se recomienda utilizar un tiempo de muestreo de 100 ms, porque al ser pequeño disminuye el error de los cálculos del PIDX y segundo porque no es tan pequeño como para necesitar de una computadora con un procesador muy rápido.

Bibliografía

1. Bollinger, J. y Duffie, Neil. (1988). Computer Control of Machines and Processes. U.S.A.: Addison – Wesley Publishing Company.
2. Díaz, J. (2002). Antología del curso Análisis de sistemas lineales. Cartago.
3. Ericsson, K. y Hedrick, J. (1999). Plant Wide Process Control. U.S.A.: John Wiley & Sons, Inc.
4. GNE Automatización Industrial de Centroamérica S.a. Recuperado el 30 de noviembre de 2004, de <http://www.gnecorp.com>
5. Greenberg, S. ActiveX Control. Recuperado el 19 de enero de 2005, de <http://cuinl.tripod.com/ActiveX.htm>
6. Interiano, E. (2004). Antología del curso Control automático. Cartago.
7. Invensys Systems, Inc. (2003). InTouch 8.0 Basic Course. Miami.
8. Kuo, B. (1996). Sistemas de control automático. México: Prentice Hall.
9. Montenegro, A. (2004, Octubre). Entrevista con Johnny Alvarado, Gerente de Ingeniería, GNE Automatización Industrial de Centroamérica S.a., San José.
10. Montenegro, A. (2005, Julio). Entrevista con Orlando Muñoz, Estadístico, INEC , San José.
11. Ogata, K. (1993). Ingeniería de Control Moderna. México: Prentice – Hall Mexico Hispanoamérica, S.A.

12. Programación con Visual Basic. Recuperado el 19 de enero de 2005, de <http://www.lawebdelprogramador.com/cursos/mostrar.php?id=93&texto=Visual+Basic>
13. Spon Press Taylor & Francis group. (1999). HVAC Control Systems: Modelling, Analysis and Design. New York.
14. Wonderware Corporation. (2001). InControl 7.11 Basic. California.

Apéndices

A.1 Glosario, abreviaturas y simbología

Abreviaturas

PID: Controlador proporcional, derivativo e integral.

PLC: Controlador lógico programable.

K: Constante de proporcionalidad del PID.

Td: Constante derivativa del PID.

Ti: Constante integral del PID.

A.2 Información sobre la empresa

Descripción de la empresa

El proyecto se desarrollará para la empresa GNE Automatización Industrial de Centroamérica S. A., la cual tiene 7 años de brindar sus servicios en Costa Rica. El gerente de la empresa es el Ingeniero Johnny Alvarado. En GNE se cuenta con 15 empleados.

La Empresa se especializa en la implementación de sistemas de control para la industria en las áreas de mantenimiento, producción y calidad. También se hacen trabajos para empresas distribuidoras de energía y agua, en los centros de control, subestaciones y plantas de generación.

Los proyectos que se desarrollan cumplen con la función de controlar el funcionamiento de las máquinas y los procesos productivos.

Descripción del departamento se realizó el proyecto

El proyecto se realizará en el Departamento de Ingeniería de la empresa, en el cual todos sus funcionarios son ingenieros. En total son 7 ingenieros, 5 en electrónica y 2 eléctricos.

En este departamento es donde se encargan del diseño electrónico de los diferentes sistemas de control, y la mayor parte del tiempo se emplea en programación, tanto de PLCs como en las computadoras en los diferentes programas utilizados por estos.

El jefe de departamento es el ingeniero Johnny Alvarado. En este departamento se cuentan con las herramientas necesarias para poder desarrollar sus proyectos, cada uno de los ingenieros cuenta con su propia computadora, y se dispone de diferentes PLCs, así como los diferentes

dispositivos que complementan el funcionamiento de éstos como lo son los sensores, los cables, etc.

B.1 Resultados parciales

Estimaciones de la función de transferencia de primer orden con tiempo muerto

En las siguientes tres tablas se muestran los resultados obtenidos de aplicar los cuatro métodos gráficos a tres modelos de primer orden con tiempo muerto.

Función de transferencia teórica:

$$G(s) = \frac{2e^{-3s}}{s+1} \quad (\text{B.1})$$

Tabla 6.8 Valores de los parámetros de la planta, obtenidos mediante diferentes métodos

	Teoría	Ziegler-Nichols [3]	II [3]	Gp [13]	Discreto [1]
ganancia	2	2	2	2	2
tiempo muerto	3	3,2	3,255	3,255	3,31
constante de tiempo	1	1,21	1,155	1,155	1,1
error	---	0.0135	0.0142	---	0.0152

Función de transferencia teórica:

$$G(s) = \frac{e^{-s}}{s+1} \quad (\text{B.2})$$

Tabla 6.9 Valores de los parámetros de la planta, obtenidos mediante diferentes métodos

	Teoría	Ziegler-Nichols	II	Gp	Discreto
ganancia	1	1	1	1	1
tiempo muerto	1	1,118	0,7025	0,7025	1,089
constante de tiempo	5	5,217	5,6325	5,6325	5,246
error	---	0.000749	0.000921	---	0.000731

Función de transferencia teórica:

$$G(s) = \frac{10}{4,5s + 1} \quad (\text{B.3})$$

Tabla 6.10 Valores de los parámetros de la planta, obtenidos mediante diferentes métodos

	Teoría	Ziegler-Nichols	II	Gp	Discreto
ganancia	10	10	10	10	10
tiempo muerto	0	0	0,197	0,1965	0,221
constante de tiempo	4,5	4,767	4,57	4,5705	4,546
error	---	0.0078	0.0098	---	0.0102