

Planta Piloto para Control de Procesos v2

Informe Final

Federico Ferreira – Sebastián Rodríguez – Pablo Rodríguez

Facultad de Ingeniería
Universidad de la República

Agradecimientos

- A la empresa ABB por facilitarnos el DCS necesario para el Proyecto. En particular a Javier Román
- A Nelson Ventura, Roberto Rodríguez y Sergio Beheregaray del taller del IIE por su disposición para la fabricación de algunas piezas de la planta.
- A Ángel Coimbra del taller de IQ por su trabajo de torneado
- A nuestro tutor Rafael Canetti por sus consejos.
- Y a todos los que colaboraron con el proyecto.

Índice

1 INTRODUCCIÓN

1	Introducción a la Planta.....	8
1.1	Descripción del la planta piloto inicial	8
1.2	Comunicación de las partes	9
1.3	Modelado de la planta	10
1.3.1	Nivel en el tanque.....	11
1.3.2	Temperatura a la salida	11
1.4	Objetivos del Proyecto	12
1.5	Resumen	12
1.5.1	Estudio del proyecto original.....	12
1.5.2	Planta y Sistema	13
1.5.3	Calibración de sensores y actuadores	13
1.5.4	Control PID	14
1.5.5	Control Adaptivo	14
1.5.6	Prácticas de Laboratorio.....	14

2 MEJORAS REALIZADAS a la PLANTA

2	Mejoras de sensores y actuadores	15
2.1	Introducción	15
2.2	Caudalímetros	15
2.3	Sensor de nivel	16
2.4	Válvulas de control.....	17
2.5	Válvula ON-OFF	23
2.6	Nuevo Tanque de agua	24
2.7	Pruebas y reparaciones realizadas a la electrónica.....	26
2.7.1	Sensores de temperatura (RTD)	26
2.7.2	Servos	26
2.7.3	Placa Driver de Servos	26
2.7.4	Fuente para la placa driver	28
2.8	Mejoras en la Estructura de la Planta	30
2.9	Nuevo Tanque de agua	25
2.9	Problemas encontrados en el rearmado final.....	31

3 INTERFACES

3	Interfaces.....	32
3.1	Introducción	32
3.2	Definición de las distintas interfaces y funcionalidades	35
3.3	PLC.....	36
3.3.1	Resumen.....	36
3.3.2	Programación	36
3.3.2.1	Configuración de la comunicación PLC-DCS	36
3.3.2.2	Nivel del Tanque	38
3.3.2.3	Caudalímetros.....	39
3.3.2.4	Válvulas.....	40
3.3.2.5	Alarmas Hardware	42
3.3.2.6	Resistencia del Tanque	43
3.4	Comunicación DCS - PLC.....	44
3.4.1	Modo de direccionamiento	44

3.4.2 Configuración de la comunicación	46
3.5 Matlab-DCS	47

4 CALIBRACIÓN de INSTRUMENTOS

4 Calibración de Instrumentos	51
4.1 Auto Calibración	51
4.1.1 Introducción	51
4.1.2 Procedimiento de Calibración	51
4.1.3 Programación de la Autocalibración	53
4.1.3.1 Introducción	53
4.1.3.2 Implementación	53
4.1.3.3 Evaluación Autocalibración	55
4.2 Calibración Manual	56
4.2.1 Caudalímetros	57
4.2.2 Sensor de nivel	58
4.2.3 Acción de la resistencia	58
4.2.4 Calibración de las válvulas	58

5 CONTROL PID

5 Controladores PID	60
5.1 Introducción	60
5.2 Control PID	60
5.3 Representaciones Alternativas.....	62
5.3.1 Controlador PID Paralelo.....	62
5.3.2 Controlador PID Serie.....	62
5.3.3 PID con Ponderación de la Referencia	63
5.4 Windup.....	65
5.4.1 Introducción	65
5.4.2 Técnicas Anti-Windup	67
5.4.2.1 Limitación de la referencia	67
5.4.2.2 Integración condicional	67
5.4.2.3 Limitación estática del término Integral	68
5.4.2.4 Seguimiento Integral	68
5.4.2.5 Back-Calculation	69
5.5 Autosintonía del PID por el método del relé.....	71
5.5.1 Objetivos	71
5.5.2 Introducción	71
5.5.3 Métodos para la obtención de los parámetros primarios.....	72
5.5.3.1 Métodos basados en la respuesta escalón	72
5.5.3.2 Métodos de oscilación mantenida con un controlador proporcional	72
5.5.3.3 Métodos de realimentación con relé	72
5.5.4 Estudio teórico del método.....	73
5.5.4.1 Ventajas de la autosintonía por el método del relé	73
5.5.4.2 Comportamiento del relé	74
5.5.4.3 Obtención de los parámetros primarios, Tc y Kc.....	75
5.5.5 Consideraciones prácticas para la aplicación del método	76
5.5.5.1 Relé con histéresis	76
5.5.5.2 Offset	77
5.5.5.3 Ajuste de la amplitud de la oscilación de salida	78
5.5.6 Proceso de Autosintonía	79
5.5.6.1 Experimento con el relé para la obtención de Kc y Tc.....	79
5.6 Implementación del PID.....	81
5.6.1 Introducción	81

5.6.2 Implementación en MATLAB.....	81
5.7 Implementación del Autotuning	85
5.7.1 Introducción	85
5.7.2 Explicación del programa	86
5.7.3 Pruebas.....	89
5.7.4 Control Paralelo de Temperatura y Nivel.....	90
5.8 Conclusiones PID.....	93

6 CONTROL AVANZADO

6 Aplicación de control Avanzado	94
6.1 Introducción	94
6.2 Resumen de los distintos objetivos.....	94
6.2.1 Autocalibración de los instrumentos.....	94
6.2.2 Adquisición de señales	94
6.2.3 Estudio de las distintas técnicas de control avanzado	95
6.2.4 Diseño y simulación del control avanzado	95
6.2.5 Implementación	95
6.3 Control Adaptivo	96
6.3.1 Introducción	96
6.3.2 Control Adaptivo	96
6.3.3 Definición	96
6.4 Objetivo.....	97
6.5 Introducción al Control adaptivo	97
6.6 Control Adaptivo por Modelo de Referencia	99
6.6.1 Suposiciones.....	99
6.6.2 Estructura del Controlador	100
6.6.3 Representación en el espacio de estados	101
6.7 Modelo de Referencia (MR) con error a la salida	103
6.7.1 MR con error a la salida y grado relativo 1.....	103
6.7.2 Ecuaciones del Controlador.....	105
6.7.3 Implementación Matlab.....	107
6.7.4 Simulación	110
6.7.5 Prueba Física.....	111
6.7.6 Interpretación de los parámetros del control MR.....	113
6.7.7 Conclusiones.....	119
6.8 Modelo de Referencia con error a la Entrada.....	120
6.8.1 Implementación del algoritmo	120
6.8.2 Implementación Matlab	122
6.8.3 Simulaciones.....	127
6.8.4 Prueba Física.....	127
6.8.5 Conclusiones.....	131

7 IDENTIFICACIÓN

7 Identificación	133
7.1 Introducción	133
7.2 Estructura del Identificador	133
7.3 Representación en espacio de estados.....	134
7.4 Estructura del Identificador	135
7.5 Ecuación de error lineal y algoritmo de Identificación.....	137
7.6 Identificación de planta de 2º orden.....	138
7.7 Identificación de planta de 1º orden.....	139
7.8 Implementación de la Identificación en Matlab	140
7.9 Pruebas y resultados	143

8 MANUAL de USUARIO

8 Manual de Usuario	144
8.1 Planta Actual	144
8.1.1 Descripción	144
8.1.2 Pasos a seguir para iniciar la planta	148
8.2 Comunicación de datos Planta-Matlab	149
8.2.1 Comunicación Planta → Matlab	149
8.2.2 Comunicación Matlab → Planta	152
8.3 DCS	153
8.3.1 Como ingresar al sistema de control	153
8.3.2 Uso de los Controladores	156
8.3.2.1 Objeto Controlador (LIC o TIC).....	156
8.3.2.2 Faceplates de los Controladores.....	156
8.3.2.3 Modos de funcionamiento del Controlador	157
8.3.2.3.1 Modo Manual.....	157
8.3.2.3.2 Modo PID Automático.....	158
8.3.2.3.3 Modo Control por MR.....	159
8.3.2.4 Autotuning PID.....	161
8.3.2.5 Identificación	162
8.3.2.6 Botones Auxiliares	163
8.3.3 Otros objetos implementados.....	163
8.3.3.1 Bomba.....	163
8.3.3.2 Resistencia	163
8.3.3.3 Válvula de realimentación	164
8.3.3.4 Indicaciones	164
8.4 Implementación de un controlador en Simulink.....	165
8.4.1 Ambiente de trabajo	165

9 CONCLUSIONES FINALES

9 Conclusiones Finales.....	171
9.1 Introducción	171
9.2 Planificación	171
9.3 Técnicas de Control.....	171
9.4 Experiencia.....	172
9.5 Prácticas de Laboratorio	172
9.6 Proyectos a Futuro.....	172

A MODBUS

A ANEXO: 'Comunicación MODBUS'.....	173
A.1 Introducción.....	173
A.2 Estructura de la red.....	173
A.3 Protocolo	174
A.4 Las funciones más importantes del protocolo.....	176
A.5 Nivel de aplicación	179

B OPC

B ANEXO: 'Comunicación OPC'	180
B.1 Introducción.....	180
B.2 Motivos para el desarrollo de la especificación OPC	180
B.3 Ventajas de OPC.....	181

B.4 Integración de sistemas heterogéneos con OPC	182
B.5 Componentes OPC	183
B.6 Hardware	183
B.6.1 Tecnología Cliente-Servidor	184
B.6.2 Modo De Operación	185
B.7 Servidores OPC.....	187

C DCS Vs SCADA

C ANEXO: 'DCS Vs SCADA'	189
-------------------------------	-----

D MATLAB - SIMULINK

D ANEXO: 'MatLab-Simulink'	190
----------------------------------	-----

E PROCEDIMIENTO de INSTALACIÓN del DCS

E ANEXO: 'Procedimiento de instalación del sistema 800xA en una PC'	191
E.1 Usuarios del Sistema	191
E.2 Instalación	191
E.2.1 Sistema Operativo	192
E.2.2 Instalación del software de terceros	194
E.2.3 Grupos y usuarios necesarios para correr el sistema	195
E.2.4 Instalación del software del sistema	195
E.3 Post-instalación	197
E.3.1 Licenciamiento del sistema	197
E.3.2 Creación del sistema.....	197
E.3.3 Carga de extensiones.....	198
E.4 Configuración y puesta a punto del sistema	198
E.4.1 Registro de usuarios del sistema	198
E.4.2 Configuración de servicios.....	199

Bibliografía.....	205
-------------------	-----

1 Introducción a la Planta

Este proyecto surge de la necesidad planteada por el profesor Rafael Canetti de tener una planta piloto a escala. Se pretende que con la misma se tenga la opción de realizar demostraciones y prácticas de control en las asignaturas del departamento de Sistemas y Control.

El punto de partida fue el proyecto realizado por ¹Berenfus-Casas-Rezk en el año 2006. El mismo consistió en una primera construcción de una planta piloto controlada por un PLC. A continuación se presenta la planta en cuestión y se realiza una muy resumida explicación de cómo se realizaba la comunicación de datos y se exponen los resultados (realizados en el proyecto previo) del modelado de las funciones de transferencia para el nivel dentro del tanque y la temperatura a su salida.

1.1 Descripción de la planta piloto inicial

La planta (ver figura a continuación) consta de un tanque, un intercambiador de calor, una bomba, tres válvulas de control, dos caudalímetros, cuatro sensores de temperatura y un sensor de nivel. Dichos componentes están interconectados por medio de cañerías y accesorios.

El tanque es un recipiente abierto, con una resistencia interna de 1200W. En dicho tanque ingresan agua fría, agua caliente y agua tibia (proveniente de la recirculación).

El agua fría proviene directamente de OSE, por lo tanto la presión a la que es suministrada está determinada por dicha institución en (aproximadamente) 3bar. Por otro lado el agua caliente es agua, también proveniente de OSE, calentada en un termotanque a aproximadamente 70°C. El agua tibia es agua que sale del tanque y es enfriada mediante el intercambiador de calor por el cual circula agua fría con este propósito, también proveniente de la red de OSE. Dicha recirculación se realiza mediante la bomba.

Los tres caudales que ingresan al tanque son controlados por las válvulas de control correspondientes a cada uno de ellos. El caudal de agua caliente se controla con la servoválvula Vc1, el frío con la Vc2 y la recirculación con la Vc3.

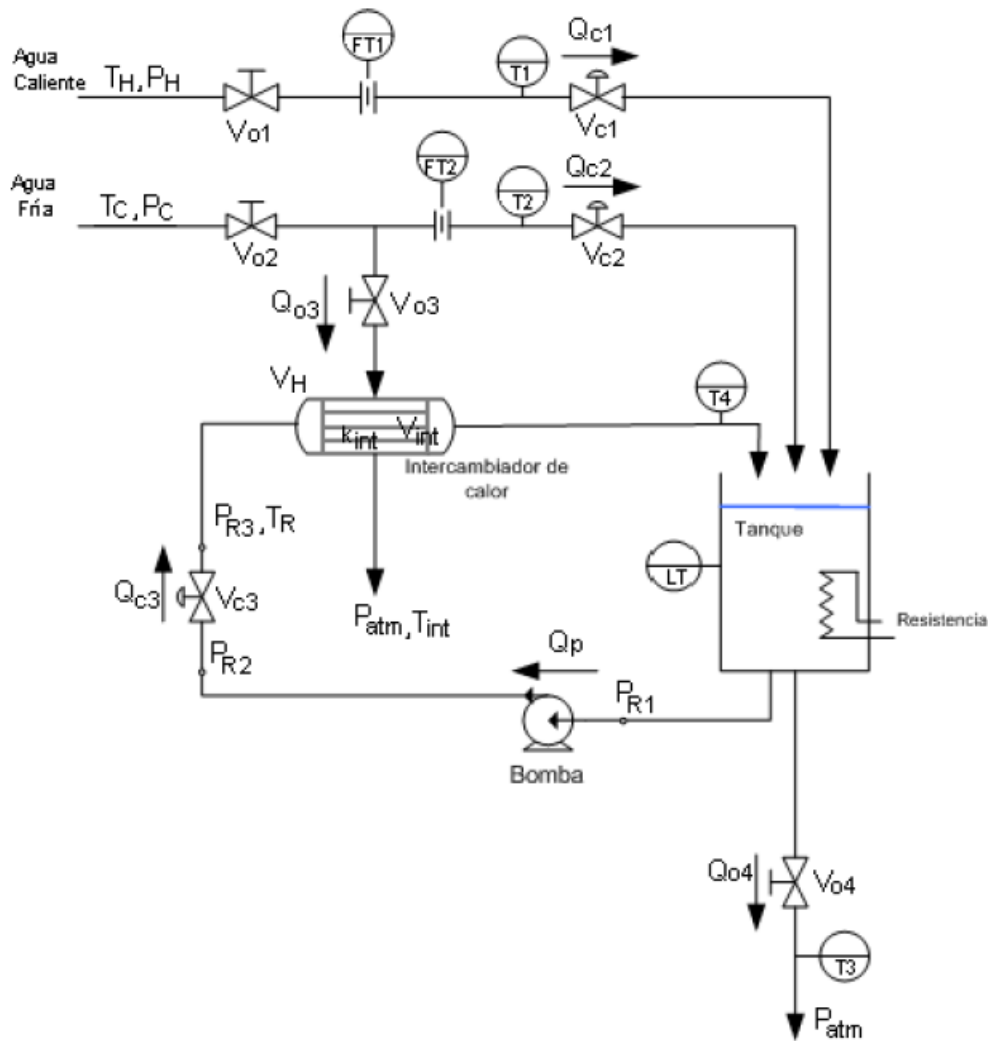
Las válvulas manuales Vo1 y Vo2 se utilizan para limitar los caudales máximos de agua caliente y agua fría respectivamente (éstas no son válvulas de control). El pasaje de caudal a través de la carcasa del intercambiador de calor se habilita mediante la apertura de la válvula Vo3 y finalmente la salida del tanque es manejada por una válvula (Vo4). Existe también una válvula manual general de entrada a la planta, la cual sirve para limitar el caudal máximo de entrada.

En lo que se refiere a los sensores dispuestos a lo largo del circuito de circulación de agua existen cuatro sensores de temperatura (RTD) para medir las temperaturas de agua

¹ (Berenfus, Casas, & Rezk, 2006)

caliente ($T1$), agua fría ($T2$), agua enfriada por el intercambiador de calor ($T4$) y salida del tanque ($T3$). Además dos caudalímetros miden el caudal de agua caliente ($FT1$) y fría ($FT2$) que ingresan al tanque. Para medir el nivel de agua dentro del tanque hay un sensor de nivel (LT).

Los actuadores (servos) así como los sensores (RTD, caudalímetros, sensor de nivel) están conectados a un PLC, el cual se encarga de leer las señales, procesarlas y escribir en los actuadores el resultado del procesamiento.

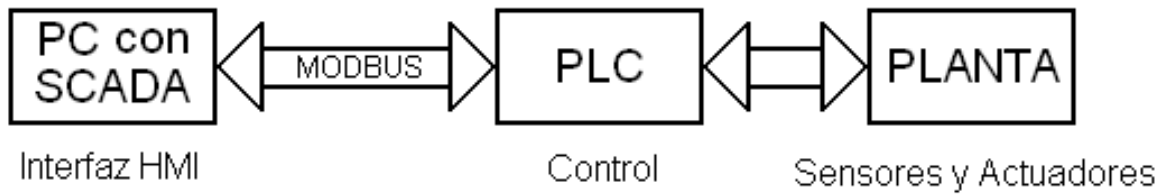


² Esquema de la planta original

1.2 Comunicación de las partes

En el proyecto original el PLC era el encargado de realizar todo el control, gracias a esto el PLC podía funcionar en solitario ('Stand Alone'), sin la necesidad de una conexión permanente con la PC. La comunicación se realizaba como se muestra en la siguiente figura:

² (Berenfus, Casas, & Rezk, 2006)



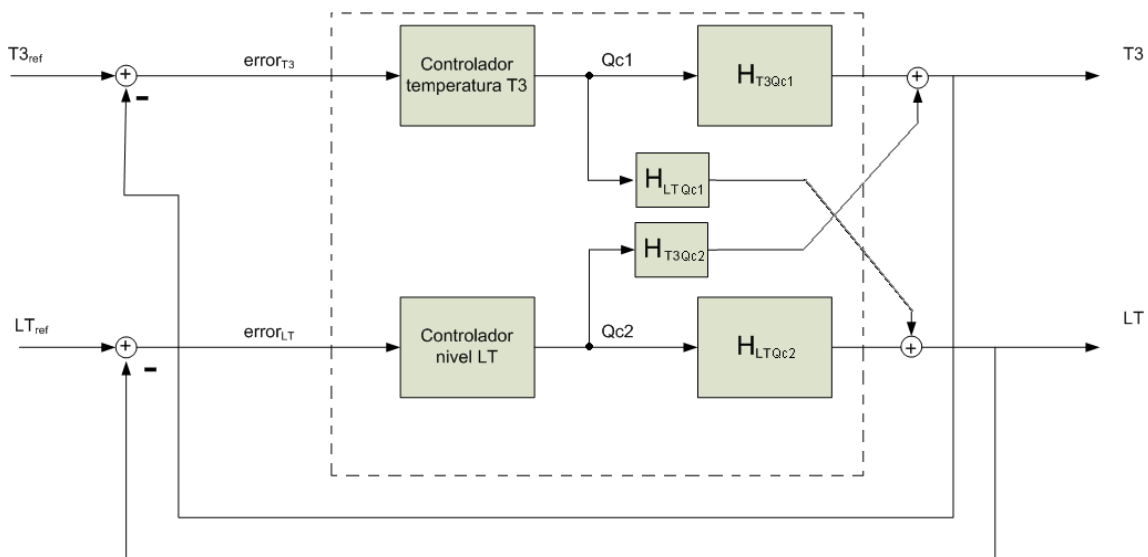
En la PC se ejecutaba el **SCADA Portal de ABB**, en el que se creó una HMI (Human-Machine-Interface) “amigable” con el usuario. Desde ésta interfaz se podían leer y escribir las variables del PLC así como:

- relevar la evolución de una temperatura, del nivel
- configurar un lazo de control para que funcionara en modo manual
- manejar de forma manual una válvula de control
- configurar los parámetros de los PID

Las variables en el SCADA se conectaban mediante el protocolo MODBUS con sus pares en el PLC, y algunas de las mismas se conectaban (de forma cableada) con los sensores y actuadores disponibles en la planta.

1.3 Modelado de la planta

Originalmente para la realización del control se partió del siguiente esquema:



³ Diseño del sistema de control

Donde:

- H_{T3Qc1} : Función de transferencia entre $T3$ y $Qc1$.
- H_{LTQc2} : Función de transferencia entre LT y $Qc2$.
- H_{T3Qc2} : Función de transferencia entre $T3$ y $Qc2$.
- H_{LTQc1} : Función de transferencia entre LT y $Qc1$.

³ (Berenfus, Casas, & Rezk, 2006)

Inicialmente como muestra la figura anterior se considera que cada variable influye en el comportamiento de la otra. Las perturbaciones de la temperatura en el nivel y del nivel en la temperatura luego en un modelo simplificado (que es utilizado para el diseño del controlador) las mismas se consideran despreciables.

1.3.1 Nivel en el tanque

⁴ El nivel en el tanque L_T es controlado por la válvula de control V_{c2} (válvula de agua fría). La función de transferencia que relaciona el caudal Q_{c2} con el nivel L_T (en pequeña señal) es:

$$\frac{L_T(s)}{Q_{c2}(s)} = H_{L_T}(s) = \frac{1}{sA_T}$$

Donde A_T es el área del tanque.

1.3.2 Temperatura a la salida

⁵ La temperatura a la salida T_3 es controlada por la válvula de control V_{c1} , abriendo y cerrando la misma gradualmente se obtienen diferentes caudales y se regula la temperatura al valor de referencia.

Para la función de transferencia de la temperatura se considera que el nivel dentro del tanque se mantiene constante, esto implica que el caudal de salida Q_{o4} tiene un valor fijo. Esta consideración se debe a que la respuesta a la temperatura es mucho más lenta que la del nivel. De esta forma la función de transferencia $\frac{T_3}{Q_{c1}}(s)$ obtenida (en pequeña señal) es la siguiente:

$$H_{T_3 Q_{c1}}(s) = \frac{\frac{(T_H - T_C)}{\bar{L}_T A_T}}{s + \frac{\bar{Q}_{o4}}{\bar{L}_T A_T}}$$

Donde:

- \bar{Q}_{o4} es el caudal medio de salida (aproximadamente 1.70 l/min)
- T_H es la temperatura de agua caliente
- T_C es la temperatura de agua fría
- \bar{L}_T es el nivel medio del tanque (aproximadamente 0.3m, medidos desde la base)
- A_T es al sección del tanque ($0.0314m^2$, tanque de 20cms de diámetro)

⁴ (Berenfus, Casas, & Rezk, 2006)

⁵ (Berenfus, Casas, & Rezk, 2006)

1.4 Objetivos del Proyecto

El objetivo principal es mejorar y optimizar la implementación previa de la Planta Piloto e implementar aplicaciones de control para poder realizar demostraciones y prácticas a ser utilizadas en la enseñanza de las asignaturas del Departamento de Sistemas y Control.

Para efectivamente mejorar la implementación de la primera versión del proyecto Planta Piloto se plantearon los siguientes objetivos específicos:

- Optimizar al menos dos de los siguientes sensores y actuadores:
 - a. Caudalímetros
 - b. Sensor de nivel
 - c. Válvulas de control

Evaluar según costo y disponibilidad la instalación de válvulas ON-OFF para de ésta manera elevar el grado de automatización de la planta.

- Evaluar la posibilidad de cambiar el tanque o las cañerías de forma de acelerar las respuestas de las variables de interés (nivel y temperatura).
- Realizar una interfaz amigable entre el sistema DCS 800xA y Matlab de manera de poder utilizar dicho programa para implementar algoritmos de control en tiempo real y realizar futuras prácticas.
- Implementar autotuning para los lazos de control de nivel y temperatura del tanque.
- Estudiar la posible automatización de la calibración de los instrumentos que integran la Planta Piloto.
- Implementar una aplicación con técnicas de control avanzado (por ejemplo: MR, autosintonía, control adaptativo, MBPC: “Model Based Predictive Control”) para lograr que el nivel o la temperatura del tanque sea capaz de comportarse como modelos de 1er y/o 2do orden definidos por el usuario del sistema
- Armar prácticas de laboratorio y demostraciones, para la enseñanza en las asignaturas del Departamento de Sistemas y Control, en particular para la asignatura ITC.

1.5 Resumen

1.5.1 Estudio del proyecto original

El primer paso para la realización del proyecto fue el estudio de la implementación de la Planta Piloto, para ello se empezó por conseguir la documentación original. La misma se fue recibiendo por partes. A poco más de un mes de empezado este proyecto se obtuvo la última parte de la documentación que tenía en su poder el grupo anterior. La misma no estaba completa pues nunca no se pudo localizar la última versión del código utilizado para la programación del PLC ni el último SCADA realizado.

1.5.2 Planta y Sistema

A partir del tercer mes del proyecto se tuvo acceso al Laboratorio de Control y en definitiva a la planta. Es en esa instancia que se comienza con el reconocimiento de la implementación física y se la compara con la documentación recibida. Posteriormente, se realizan pruebas de los elementos de la planta, encontrándose los elementos, que se detallan en el capítulo 2, en mal estado o sin funcionamiento.

Uno de los primeros objetivos planteados fue conseguir un sistema de HMI (Human-Machine-Interface) que no tenga las limitaciones de los sistemas FIX y Scada Portal y que permita mostrar claramente los distintos comportamientos de los modelos de las distintas plantas. Para ello se iniciaron, a poco de empezado el trabajo, tratativas con la empresa ABB para la donación del DCS 800xA con su respectiva llave hardware (licencia). Las mismas culminaron exitosamente pues se consiguió la donación del software y la licencia para el uso del sistema en facultad en hasta 10 máquinas.

Para garantizar el funcionamiento del sistema 800xA se hizo necesario que la facultad adquiriese una PC con determinados requisitos mínimos, hecho que se vio retrasado en varias ocasiones, llegando la PC recién a los siete meses del inicio.

Mientras se esperaba por la PC se empezó con el estudio de técnicas avanzadas de control y además se comenzaron a simular en Simulink los diferentes controles que se implementaron a posteriori. Una vez obtenida la PC se procedió a instalar el DCS y el software necesario para su utilización (el procedimiento se detalla en el Anexo E).

Con el DCS funcionando se crearon los objetos (sensores, actuadores y controladores) que forman parte del control de la planta y se conectaron las variables del programa del PLC con sus respectivos pares en el DCS (vía Modbus) y éstas últimas con los correspondientes en Matlab (vía OPC). Adicionalmente se crearon diversos Faceplates (pantallas) que funcionan como interfaz amigable con el usuario, facilitando así la utilización del sistema.

1.5.3 Calibración de sensores y actuadores

Como un objetivo secundario del proyecto se propuso mejorar la calibración (capítulo 4) poco flexible de los instrumentos de la planta, en la cual, una vez obtenidas las constantes por medio de experimentos, se debían programar las mismas en el PLC.

En este proyecto se propuso realizar una calibración automática de los instrumentos. Luego de probado el código de la misma, por las razones que se expondrán más adelante se decidió que la calibración no fuera automática. Para compensar este hecho se crearon un faceplate específico de calibración, una hoja de Excel (dónde se guardan los resultados de la calibración completa y además se pueden cargar los datos directamente al sistema presionando un botón), y un procedimiento detallado de calibración de todos los sensores y actuadores.

En paralelo se continuó con el estudio y la implementación de distintas estrategias de control, y se realizaron mejoras no previstas, por ejemplo un nuevo tanque para dinamizar los procesos.

1.5.4 Control PID

En el capítulo 5 se centra el análisis en los controladores PID. A modo de introducción se explica su funcionalidad y se describen varias representaciones alternativas. En particular se describe el PID con ‘Ponderación de la Referencia’ que tiene un buen desempeño cuando hay altos niveles de ruido en las señales. Luego se estudian las debilidades del PID teórico y se detallan maneras de mejorar su desempeño en la práctica, en particular se tratan los efectos de Windup Integral y la técnica anti-windup denominada: ‘Back Calculation’. Además se adiciona el requerimiento extra de que sea un controlador Bumpless.

Otro punto de particular interés es el estudio de técnicas de autosintonía de dichos controladores y la implementación de la técnica de autosintonía denominada: ‘Método del Relé’ de Aström.

1.5.5 Control Adaptivo

Una vez estudiadas en detalle las distintas partes del sistema y solucionados la mayoría de los inconvenientes encontrados en el camino, los esfuerzos se concentraron en comenzar a estudiar posibles implementaciones de control avanzado (capítulo 6). Con el tutor se trazó como meta lograr que el nivel o la temperatura en el tanque de la planta se comporte de acuerdo a un modelo de segundo orden deseado.

Como inicialmente las pruebas realizadas a los controladores por modelo de referencia no daban buenos resultados se decidió implementar la Identificación de forma separada al controlador y estudiar su funcionamiento.

Para esto se detalla la teoría de Identificación (capítulo 7) para una planta (con función de transferencia propia) de orden ‘n’ y se estudian en particular las identificaciones para plantas de segundo y primer orden.

Al obtener resultados exitosos en las pruebas realizadas se decidió retomar con más ímpetu el estudio de controladores por modelo de referencia.

Luego de un estudio de varios meses, que se centró en los motivos por los cuales falló la primera versión de controlador por modelo de referencia (Modelo de referencia con error a la salida y orden relativo 1), se logró concluir exitosamente la implementación del Modelo de Referencia con error a la Entrada.

1.5.6 Prácticas de Laboratorio

Se realizaron un conjunto de prácticas de laboratorio (basadas en la Planta Piloto v2) de manera que resulten un complemento de las prácticas actuales de la asignatura Introducción a la Teoría del Control.

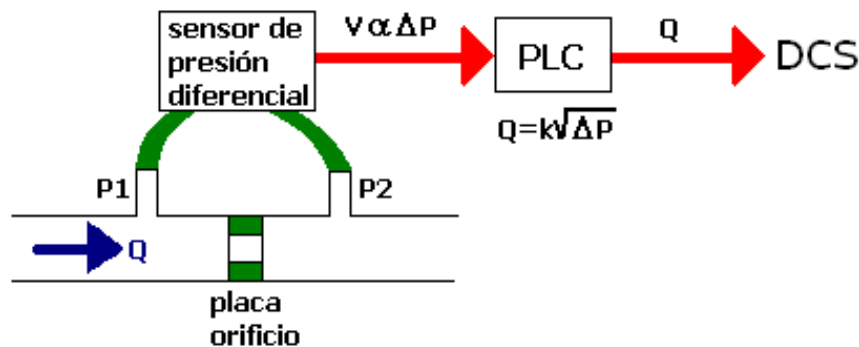
2 Mejoras de sensores y actuadores

2.1 Introducción

Una vez estudiada la implementación de los distintos actuadores y sensores que dispone la planta se propusieron (a nuestro tutor) y llevaron a cabo las mejoras que se detallan a continuación. Además se repararon averías debidas principalmente a la no utilización de planta por un período prolongado.

2.2 Caudalímetros

Estos componentes se implementaron con sensores de presión diferencial. En el mercado los caudalímetros que se encuentran son mayoritariamente de éste tipo⁶.



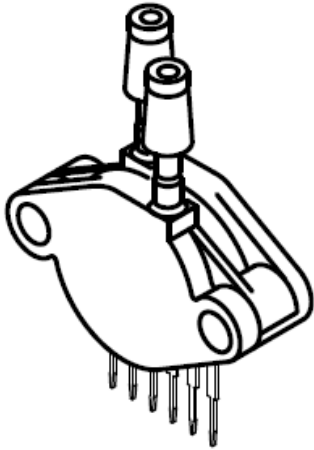
⁷ Esquema de implementación de un Caudalímetro

Los sensores utilizados, MPX5010DP, se mantuvieron ya que su valor ronda los 70 dólares (considerando los gastos de envío) mientras que un caudalímetro industrial compatible con los requerimientos del proyecto cuesta unos U\$S 500.

En su implementación previa los sensores se encapsularon en unas cajitas que se mantuvieron cerca del lugar dónde se medía la diferencia de presión. Una debilidad encontrada en la conexión PLC-sensores fue que los mismos se conectaban a una pequeña placa, un filtro, mediante un pedazo de socalo, quedando los pines del sensor poco sujetos. Esto ocasionaba que de vez en cuando se perdiera la lectura de algún caudal. Por este motivo se decidió centralizar todos los filtros en una placa. La misma se ubicó en las cercanías del PLC por lo que el filtrado se realiza más cerca del mismo. También se optó por conectar los sensores a dicha placa mediante cables que en sus extremos tienen un conector hembra de 3 pines. De esta forma se mejoró el contacto de los pines ya que se les redujo el movimiento.

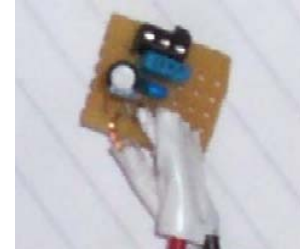
⁶ (Berenfus, Casas, & Rezk, 2006)

⁷ (Berenfus, Casas, & Rezk, 2006)



PIN NUMBER			
1	V_{out}	4	N/C
2	Gnd	5	N/C
3	V_S	6	N/C

NOTE: Pins 4, 5, and 6 are internal device connections. Do not connect to external circuitry or ground. Pin 1 is noted by the notch in the Lead.



Filtro y sócalo al que se conectaba cada sensor

MPX5010DP utilizado, 3 de los pines NC

En la documentación original se menciona que el caudalímetro de salida no funciona, pero no se especifica el por qué. Se realizó una prueba del dispositivo, resultando en que el integrado se encontraba quemado. Se consultaron empresas por una posible donación de integrados y como no se obtuvieron respuestas, se lo decidió quitar pues no es de vital importancia.

2.3 Sensor de nivel

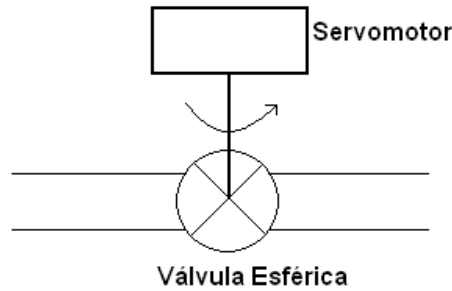
El sensor de nivel está implementado con un sensor diferencial de presión en el cual se mide directamente la diferencia de presión entre el fondo del tanque y la atmósfera (esto último debido a que el tanque es abierto).

Como el sensor de nivel se implementó de manera similar a los caudalímetros, para mejorarlo, se procedió de manera análoga. Además vale la pena mencionar que debido al costo de un sensor de nivel industrial, unos U\$S 500, se optó por mantener el ya implementado.

2.4 Válvulas de control

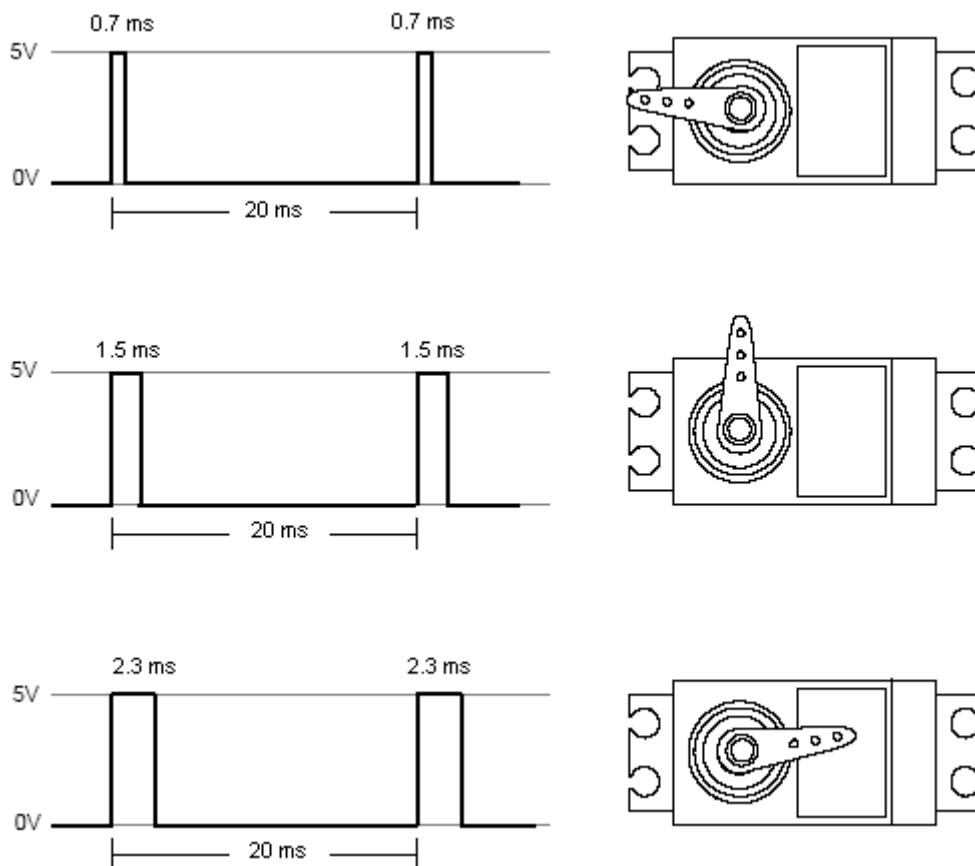
Dado el elevado costo de las mismas (U\$S 300), y dado que no se recibieron donaciones de las empresas consultadas, se optó por realizar mejoras a las válvulas existentes.

Las válvulas de control originalmente se construyeron con una válvula esférica manual y un servomotor que la comanda.



Esquema de una válvula de Control de fabricación casera.

Como actuadores se utilizaron servomotores pues los mismos tienen la capacidad de lograr y mantener una posición, que se le indica por medio de una señal de control. Si la señal de control de entrada (pulso) está entre 0.7ms y 1.5ms el servo se posiciona moviéndose hacia un lado, mientras que si la señal se encuentra entre 1.5ms y 2.3ms el motor se mueve hacia el otro lado, como se muestra en la figura:



• Relación Caudal Vs Apertura e Histéresis

En un principio se atacó el problema de la no linealidad que existe entre el caudal que circula por la válvula y el grado de apertura de la misma. Por recomendaciones de personas que ya antes habían pensado soluciones al problema fue sugerida la siguiente solución:

Se utilizan válvulas esféricas de 1/2", se remueve la esfera, se la rellena con una barra de nylon y se la torneó de forma tal que el nylon se adapte a la curvatura de la esfera. Una vez completado el torneado, de un lado de la esfera se perfora un triángulo equilátero hasta la mitad de la bolita, del otro lado lo mismo pero con el triángulo invertido de forma tal que el nuevo caudal de la válvula sea de 1/4" (como fue proyectado inicialmente). De ésta forma se logra llegar a una curva más lineal de apertura-caudal.

La primera válvula (100% metálica) se construyó a partir de un bulón, del cual se extrajo la parte sin rosca. La pieza resultante se cortó al medio longitudinalmente, para luego extraerle el contenido y formar medio triángulo en cada parte. Posteriormente se soldaron ambas partes y quedó construido un cilindro con la forma del triángulo invertido. Luego la pieza se torneó y se introdujo dentro de la bola de la válvula. Para testearla se la incluyó en una de las entradas de agua de la planta, pero dado el precario torneado, la misma se atascaba. Como forma de evitar que el motor se trancara se la dejó floja, lo que a la larga se tradujo en una molesta pérdida de agua. Finalmente, por este motivo, se optó por re-acondicionar una de las válvulas del proyecto anterior e incluirla en el lazo de menor importancia, el de recirculación.

En total, con la solución del triángulo invertido propuesta, se armaron 3 válvulas de paso horizontal y de 1/2" de caudal. Para las dos válvulas restantes se utilizó nylon como relleno.

En el caso de las válvulas que se solucionaron con nylon, se introdujo la varilla de nylon en la esfera, se la torneó (se agradece a Ángel Coímbra del IQ su esfuerzo en el torneado de las mismas) y se realizó una perforación longitudinal de un poco menos de 1/4" (1/4" es la sección de la cañería de la planta). Luego se talló a mano, la forma de los triángulos en cada cara, actividad que demandó muchas horas (dada la dureza del material).

En la siguiente figura se resalta el triángulo tallado:

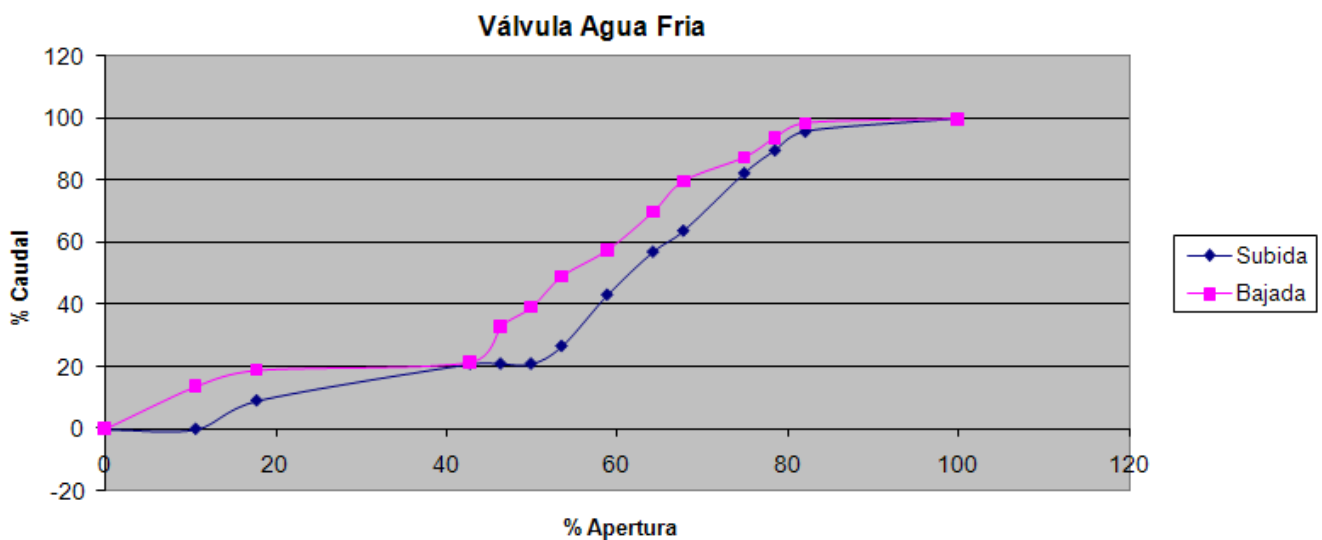


Tallado del Triángulo

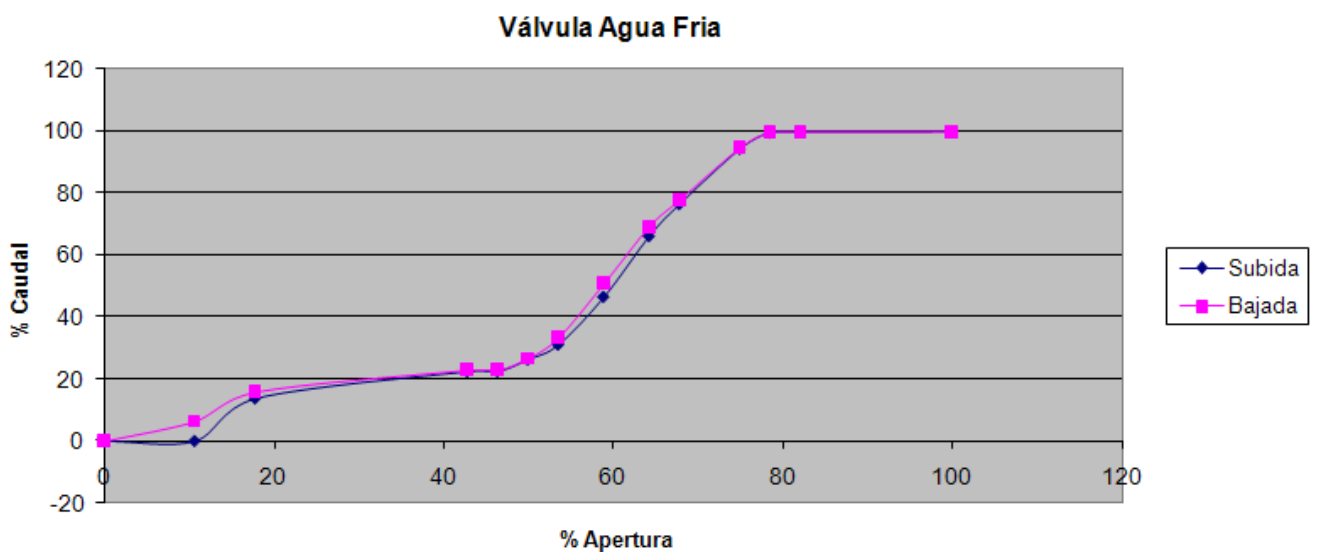
Las otras dos válvulas de control que si se diseñaron para éste proyecto se incluyeron en la entrada de agua fría y en la de agua caliente.

A continuación se muestran los resultados de los relevamientos de las curvas apertura-caudal. Vale la pena mencionar que a la señal de control de apertura de la válvula se le sumó una onda cuadrada de alta frecuencia y baja amplitud, para disminuir la histéresis (solución propuesta en el proyecto anterior).

Para agua fría con una amplitud de onda cuadrada muy baja, unas 50⁸ cuentas PLC, lo que equivale a aproximadamente un 5% de apertura, se obtiene:

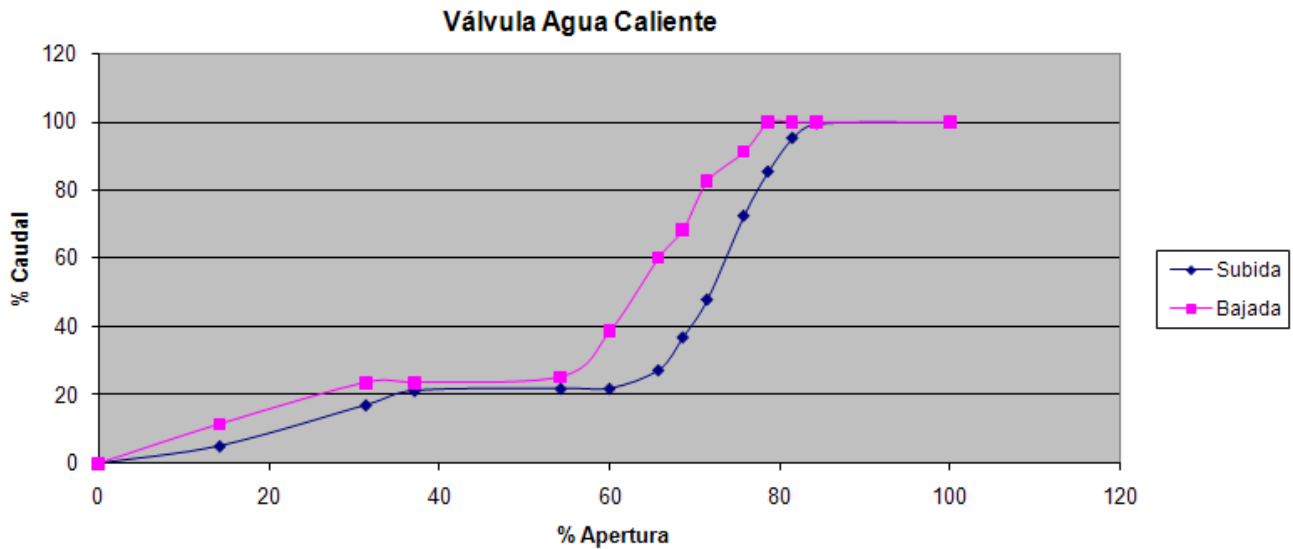


Para agua fría con una amplitud de onda cuadrada del triple del experimento anterior se obtiene:

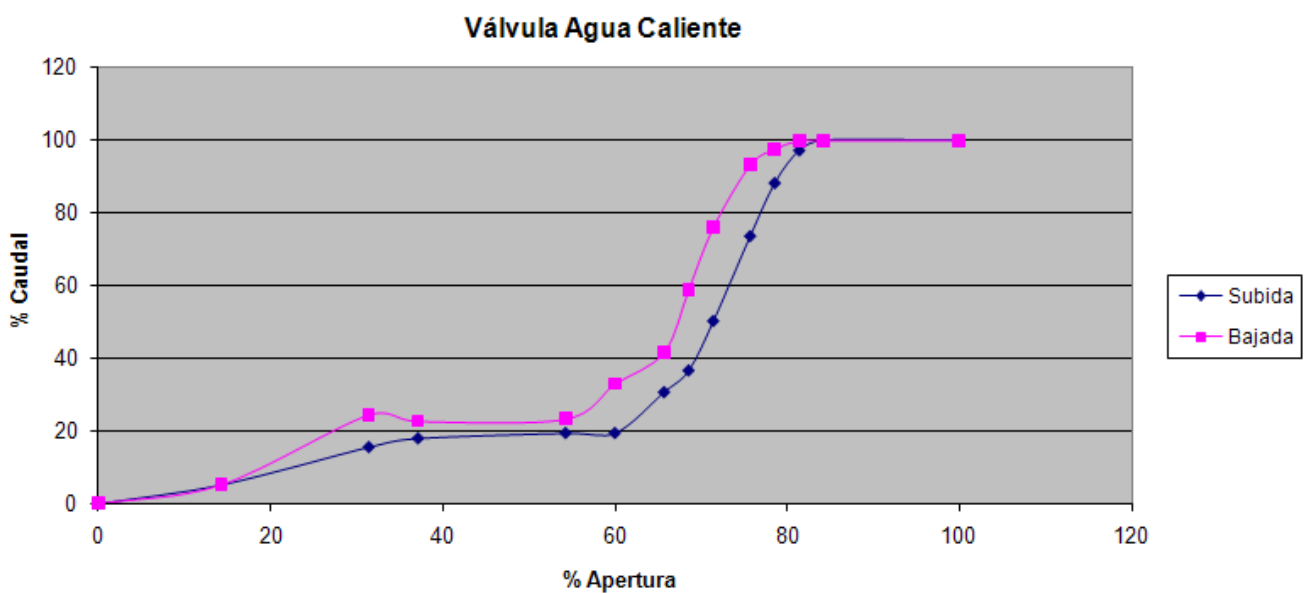


⁸ (Canetti, 2008)

Para el caso de la válvula de agua caliente, como la misma no quedó bien tallada y tampoco muy bien torneada el efecto de histéresis es bastante mayor por lo que se decidió en principio sumarle una onda cuadrada de una amplitud equivalente al 20% de apertura; obteniéndose la siguiente gráfica:



Posteriormente se volvió a relevar la curva pero con una amplitud de la onda cuadrada de 300 cuentas y se obtuvo la siguiente curva:

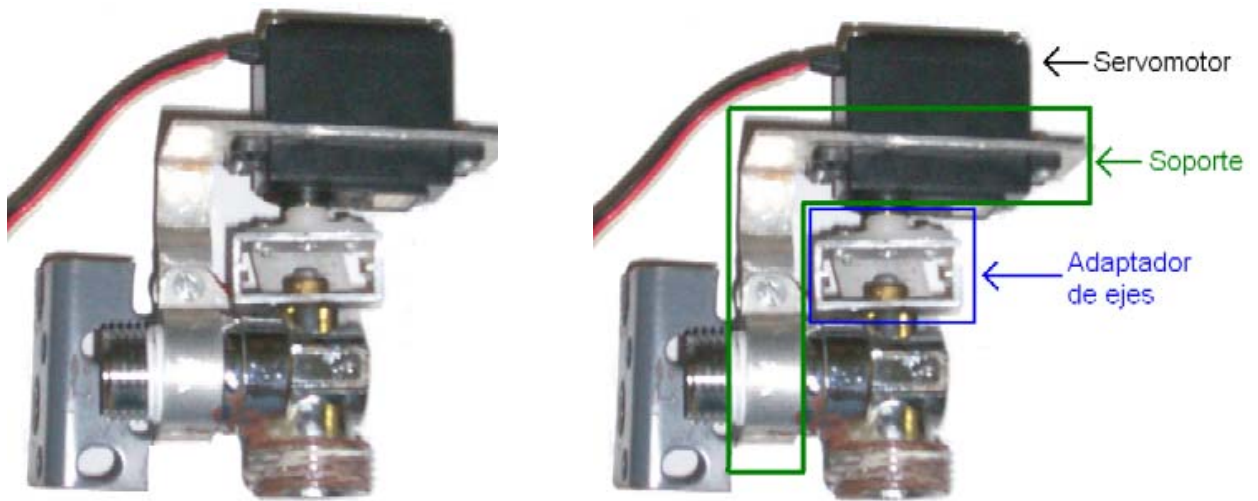


Finalmente se configuró la amplitud de la onda cuadrada en 350 cuentas (aprox.: 35% apertura) obteniéndose una curva con histéresis despreciable.

Un hecho interesante de resaltar es que la resolución que se obtuvo (rango en “cuentas PLC”) para válvulas con la solución de triángulo invertido propuesta, en comparación con la misma válvula sin la solución, es aproximadamente cinco veces mayor. O sea que en las nuevas válvulas, los servomotores se desplazan un ángulo útil cinco veces mayor que en el caso de válvulas de esfera comunes.

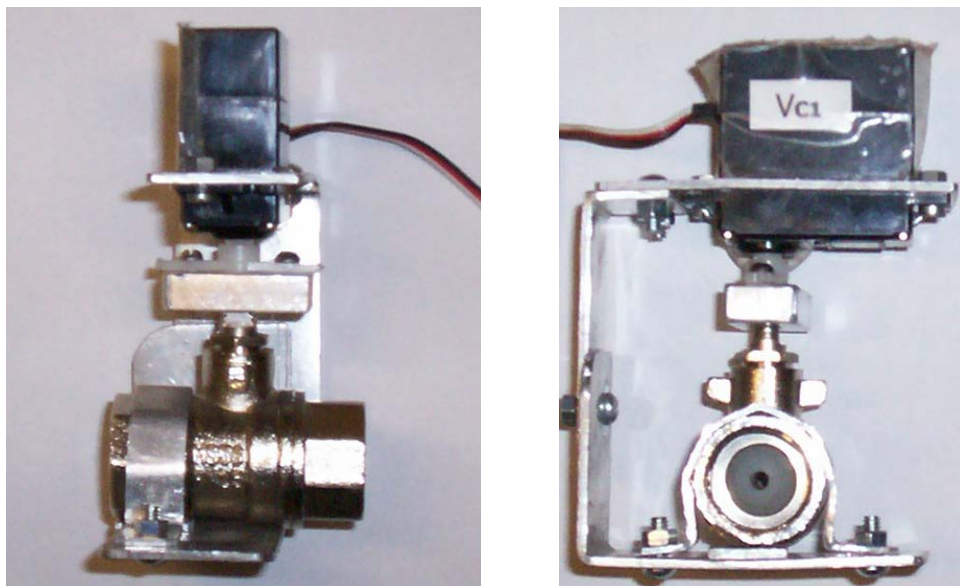
Robustez del soporte

Las válvulas originalmente fueron implementadas de la siguiente manera:

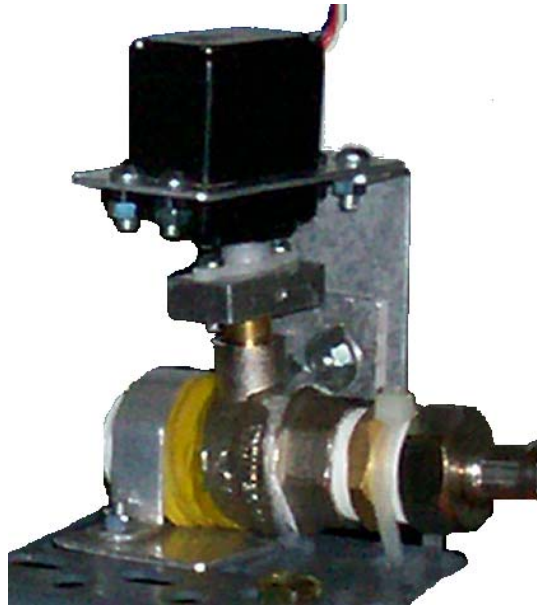


En la primera revisión de la planta se pudo comprobar que los soportes diseñados presentaban cierta debilidad ya que fácilmente se doblaban con una mano. Proyectado a futuro, dentro de unos años los mismos estarían tan doblados que harían que el eje del servo y el de la válvula se desalineen a un punto tal que sea tanto lo que se sobre-exija al servo que los daños ocasionados al mismo serían irreparables. Otro motivo por el cual es deseable reforzar el soporte es disminuir la histéresis de las mismas. Teniendo en cuenta esto y que las nuevas válvulas son de mayor tamaño se consultó a los talleristas del instituto de eléctrica para que fabriquen un nuevo soporte.

La implementación previa al “tallado” de la esfera interior se puede observar en la siguiente figura:



Si se comparan ambas implementaciones, se ve que la válvula original era una canilla de 1/4" cuya forma exterior es cilíndrica. Esto dificulta sujetarla de manera fija con el agarre ideado. La nueva válvula, por el contrario, tiene una forma exterior octogonal, lo que facilita su fijación al soporte.



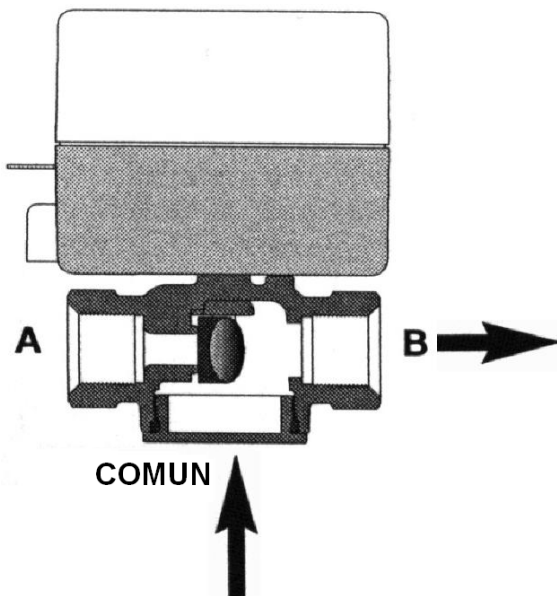
Válvula, con la solución del triángulo invertido, fijada a la planta y funcionando

2.5 Válvula ON-OFF

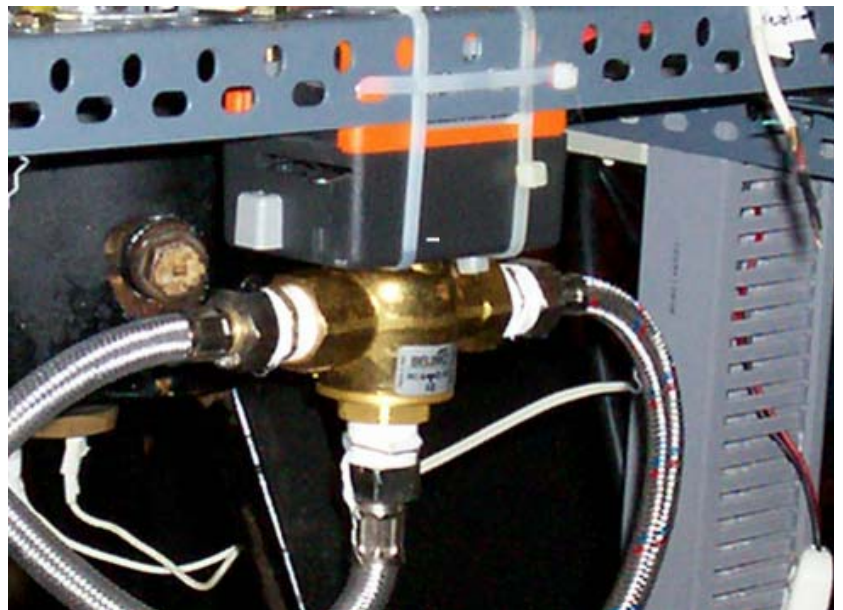
Se consiguió a préstamo una válvula ON-OFF solenoide de diafragma (sin hoja de datos). La misma fue probada a la salida del tanque y no funcionó. Luego de realizar varias consultas se nos informó que éste tipo de válvula necesita una diferencia de presión mínima de 3m (en nuestro caso de agua) para que se abra el diafragma. Como el tanque posee una altura máxima de 40cm se procedió a su devolución.

Consultando en el mercado, se encontró en Tenisol (empresa avocada al rubro Aire Acondicionado), una válvula ON_OFF de 3 vías de 30 segundos de respuesta, la cual fue adquirida. Otras empresas consultadas prometieron recibir válvulas ON-OFF de presión diferencial cero, pero no llegaron a cumplir en el plazo límite fijado (Bako, Aprahamian).

La actual válvula ON-OFF comunica siempre 2 de las 3 vías. La misma posee un motor que al energizarlo gira una especie de tapón que se encuentra en la intersección de las 3 vías y comunica la entrada común con la otra entrada (la “no-normal”). Para aclarar ideas puede verse la siguiente figura:



Detalle Interno de la válvula ON-OFF



Válvula ON-OFF instalada en la planta

Una desventaja de ésta válvula es que demora en llegar a la posición no-normal 30 segundos, pero el recorrido inverso lo realiza en menos de 5 segundos.

2.6 Nuevo Tanque de agua

Con el caudal de agua que se manejaba la respuesta del nivel del tanque era muy lenta para presentar lo que se deseaba mostrar. Por este motivo la primera opción evaluada fue la de incrementar el diámetro de las cañerías. Al estudiar el tema se concluyó que no era una buena opción ya que la temperatura también tiene un respuesta lenta, además de estar limitada por la cantidad de agua caliente que puede entregar el calefón sin que el agua del mismo empiece a disminuir su temperatura de salida.

La decisión tenía que pasar por un cambio que ayudara a disminuir los tiempos de respuesta tanto de temperatura como de nivel. La medida más natural para acelerar la respuesta del nivel fue la de construir un tanque con menor diámetro. Además con esto se disminuye el volumen del mismo y la respuesta de la temperatura resulta más rápida. Por este motivo, para el cálculo del tanque, sólo se tuvo en cuenta el lazo de nivel.

Diámetro del Tanque:

Como objetivo se planteó que el nivel, con la canilla de salida cerrada y la válvula de entrada de agua fría abierta 100% (aprox. 3.0 l/min.), pasara de 0 a 100% en un tiempo entre 20 y 30s. Esto implica que si se mantiene la altura del tanque actual, unos 40 cms, el volumen “útil” (cubriendo la resistencia, unos 18cms*área) estaría en el rango comprendido entre 1 y 1.5 litros.

En un caño de acero inoxidable de diámetro exterior 4 pulgadas, el diámetro interno es 10 cm, lo que da un área de: $\frac{\pi * 0.1^2}{4} = 0.00785m^2$. Si se utiliza este caño como nuevo tanque, tomando como nivel útil del para el control unos 18 cms, se tiene un volumen útil de 1.41 litros, lo cual e se encuentra dentro del rango deseado ([1,1.5] litros).

Sección de salida:

La misma no sufrió modificaciones y adicionalmente se dejaron 2 salidas para tener mayor flexibilidad en proyectos futuros (si se deseara manejar el caudal de salida).

Tanque Actual:

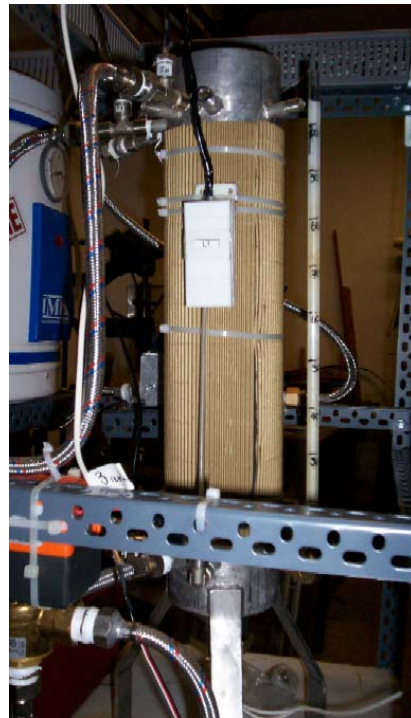
Actualmente el tanque mide unos 40 cms de largo, tiene un diámetro interno de unos 10cms y está fabricado en acero inoxidable.

Comparando los volúmenes del tanque anterior y del nuevo, se mejora la respuesta (en rapidez) de las variables de nivel y temperatura en un 75% (el tanque anterior era de 20 cms de diámetro).

A continuación se presentan los tanques de agua del proyecto:



Tanque Viejo



Tanque Nuevo

En el nuevo tanque se aprecia una aislación removible que consta de una cobertura de goma-eva y por encima cartón corrugado. De ésta manera se pretenden disminuir las pérdidas de calor por convección, aunque podría resultar interesante remover ésta aislación durante algún experimento.

2.7 Pruebas y reparaciones realizadas a la electrónica

2.7.1 Sensores de temperatura (RTD)

Se inspeccionó el estado en el que se encontraban las RTDs utilizadas. Para eso se procedió a extraerlas de la planta, luego se limpiaron (se les quitó el sarro acumulado con limpiador de contactos) y utilizando el calibrador Druck-TrxII (de General Electric) se comprobó que las 4 RTD funcionaban correctamente.

2.7.2 Servos

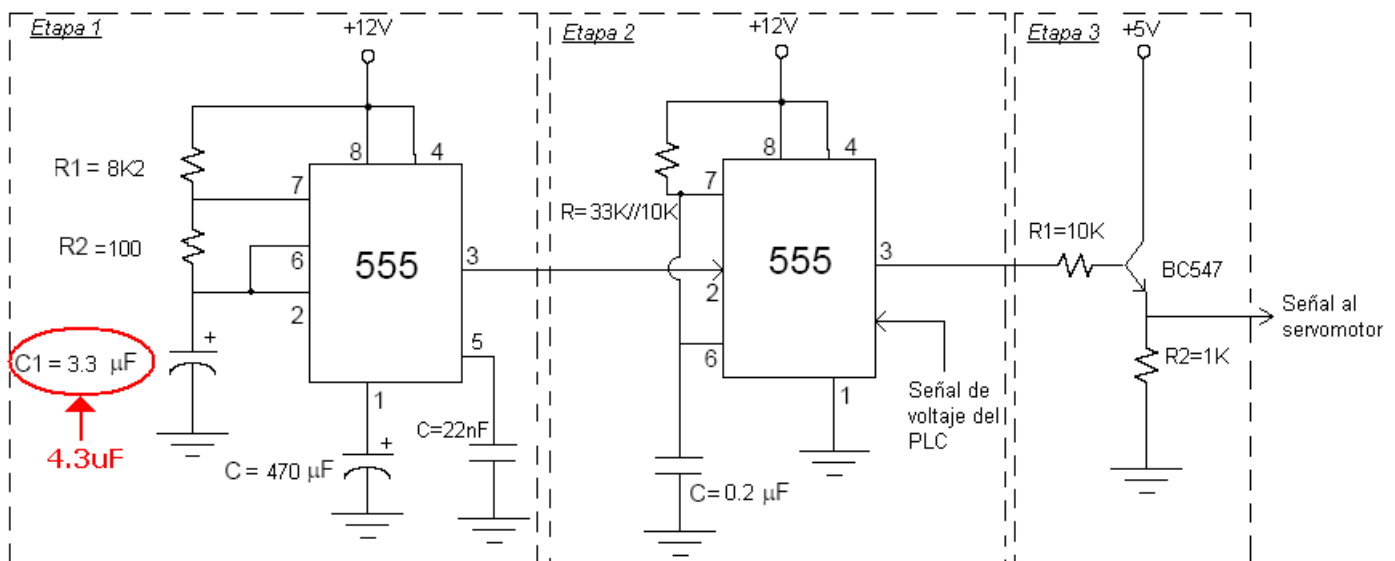
En la evaluación preliminar de daños se notó que uno de los servomotores se encontraba trancado, era el de menor importancia pues se trataba del lazo de recirculación. Este servo se desarmó, se engrasó y se volvió a armar. Esto no tuvo efecto alguno optándose por adquirir un nuevo servo de mayor torque (13kg.cm Vs 5.4kg.cm a 4.8 Volts de alimentación).

Otro de los servos al ser comandado emitía un sonido que indicaba que un engranaje estaba patinando. Por este motivo se procedió a desarmarlo y se descubrió que uno de los engranes internos, de plástico, tenía los dientes limados. Esto se debe a que dicho motor se exigió demasiado (se lo pasó de vuelta). Por dicho motivo y al no encontrar repuestos (sólo los hay para servos con engranajes de metal) se optó por adquirir un servo de mayor torque (13kg.cm Vs 7.2kg.cm a 4.8 Volts), el cual es armado con engranes de bronce lo que permite exigirlo aún más.

2.7.3 Placa Driver de Servos

Antes de proceder a comprar nuevos servos se testeó la placa que genera la señal modulada en PWM ('pulse width modulation') que los comanda.

Un esquema de la misma puede observarse en la siguiente figura:



Explicación:

La primera etapa es la encargada de generar la señal para el trigger de la segunda etapa. El trigger es un pulso bajo con una frecuencia de 50Hz, que es lo que el servo necesita.

La segunda etapa cumple la función de modular el pulso de acuerdo al voltaje ingresado en el pin número 5 del integrado. En este caso el circuito actúa como un multivibrador monoestable.

Finalmente en la tercer etapa se limita la salida al motor en un voltaje de 5V, ya que es el voltaje que admite dicho servomotor.

Pruebas de Funcionamiento:

Para determinar el funcionamiento de la placa se procedió de la siguiente manera:

1. Con un generador de señal se inyectó una señal de continua de 0 a 10V en la entrada del circuito (pin nº5 del 555 de la etapa 2, donde dice: 'Señal de voltaje del PLC').
2. Con el osciloscopio se relevó la salida (señal al servomotor).
3. Se varió el voltaje de entrada y se observó como respondía la salida del circuito.

En la inspección anterior se constataron dos fallas:

1. El 555 de la etapa 2 del generador de señal modulada que comandaba la válvula de realimentación se encontraba quemado pues su salida no era un tren de pulsos.
2. Un condensador en la etapa 1 del circuito, que se encarga de generar un pulso bajo con frecuencia de 50 Hz, se encontraba mal soldado por lo que hacía falso contacto y el período de la señal de salida variaba considerablemente.

Además de estas fallas se constató una diferencia en el condensador que se calculó en el proyecto previo y lo que realmente se soldó en la placa. En la documentación dicho condensador se calculó en 3.3uF lo que da como resultado un trigger de menos de 50Hz. En la placa se encontró un condensador de 3.3uF en paralelo con uno de 1uF (ver figura anterior) lo que sitúa la frecuencia de la señal de trigger muy cerca de los 50Hz. Este cambio no se documentó, y además gracias a que el condensador de 1uF se encontraba correctamente soldado es que a pesar del falso contacto del de 3.3uF igual se generaba una señal modulada.

Una vez notada la diferencia se procedió a soldar el condensador en cuestión y se corroboró que el período de la señal de salida fuera de 20ms y constante.

Lo siguiente que se hizo fue conectar un servomotor a la placa, variar la entrada y relevar el rango de valores de entrada en el que el mismo recorre los 180°. En la práctica se llega a que para una entrada que varía aproximadamente entre 3.7 y 9.6 volts se tiene que el servo recorre los 180° (en realidad unos más) que en teoría se mueve. Lo que es lo mismo que la señal que lo controla tiene para una entrada de 3.7V un TH=0.7ms (tiempo en 'alto' de 0.7ms) que corresponden a los 0° y para una entrada de 9.6V un TH=2.3ms que corresponde a los 180°.

2.7.4 Fuente de la placa driver

La fuente encargada de generar los 12 y 5 Volts, que precisa la placa driver de los servos, se trata de una fuente electrónica. Un testeo primario mediante un voltímetro corroboró su salida. A pesar de este hecho se dudó de su estado ya que la misma emitía un “zumbido”.

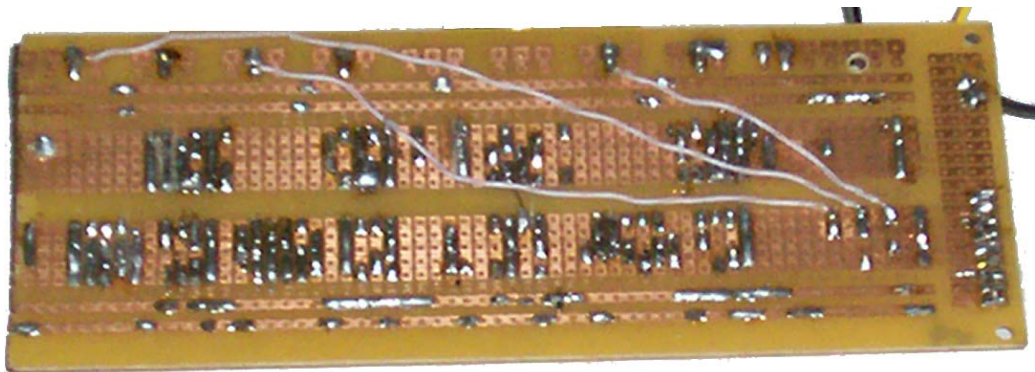
Pruebas:

Una vez reparada la placa que maneja los servos se conectó la misma a la fuente para de ésta manera poder testearla. Mediante el generador de señales se le inyectó en una de las entradas de la placa driver una señal de continua. Con el osciloscopio se relevó la salida correspondiente y se constató que al variar la entrada entre 0 y 5 volts, la salida (de la placa driver) se modula adecuadamente, pero para un voltaje de entrada superior a 5V se observaba que la misma tenía una variación del tiempo en que permanecía alta. Esto se traduce, al conectar el servo, en una oscilación respecto a la posición deseada que a medida que aumenta el voltaje de entrada se torna más apreciable.

Hipótesis de mal funcionamiento:

En este punto se plantearon dos hipótesis a las cuáles se debía la oscilación:

1. La placa driver de los servos no está adecuadamente soldada, se observan en algunas soldaduras gotas excesivas de estaño y además (dado el tipo de placa utilizado) las pistas no están en buenas condiciones. Esto podría indicar que la fuente de ruido se encuentra en dicha placa.



Soldaduras en la placa Driver de los Servos

2. La fuente presenta ruido.

Solución:

Por simplicidad se optó por empezar por la fuente (hipótesis 2) ya que hacer la placa driver desde cero insumiría mucho tiempo. Para ello se sustituyó la misma por una fuente, no electrónica, de mayor tamaño, con las mismas salidas (12 y 5 volts). Durante el testeó se corroboró un desempeño peor, la salida presentaba mucho ruido y oscilaba demasiado como para mover el servo.

Luego se optó por la prueba más simple y que descarta cualquier problema de ruido en la fuente, probar con una batería de 12V de auto. Esta vez para generar los 5V se utilizó un regulador de voltaje TIP7805. La prueba resultó exitosa, se eliminó todo ruido y al aumentar el valor de la entrada la salida varió sin oscilaciones.

Finalmente, a la luz de las pruebas realizadas, se optó por mejorar la implementación de la fuente. Para eso se cambiaron los cables que alimentan la placa, realizándose una mejor soldadura; se cambiaron los cables de salida y se arreglaron algunas pistas defectuosas. Luego se consultó la hoja de datos de los LM555, y se constató que el rango del voltaje requerido para su correcto funcionamiento es de 4.5V a 16V. Por este motivo se decidió utilizar un regulador de voltaje TIP7810 para reducir el voltaje de alimentación a la placa (de 12V a 10V) y de alguna manera “filtrar el ruido”, pero además, para alimentar el circuito con un voltaje más estable aún, se colocó un condensador en los bornes de salida del regulador. Hecho esto se volvieron a testear los servomotores comprobándose una reducción significativa de la oscilación presentada para voltajes de entrada mayores a los 5V.

2.8 Mejoras en la estructura de la Planta

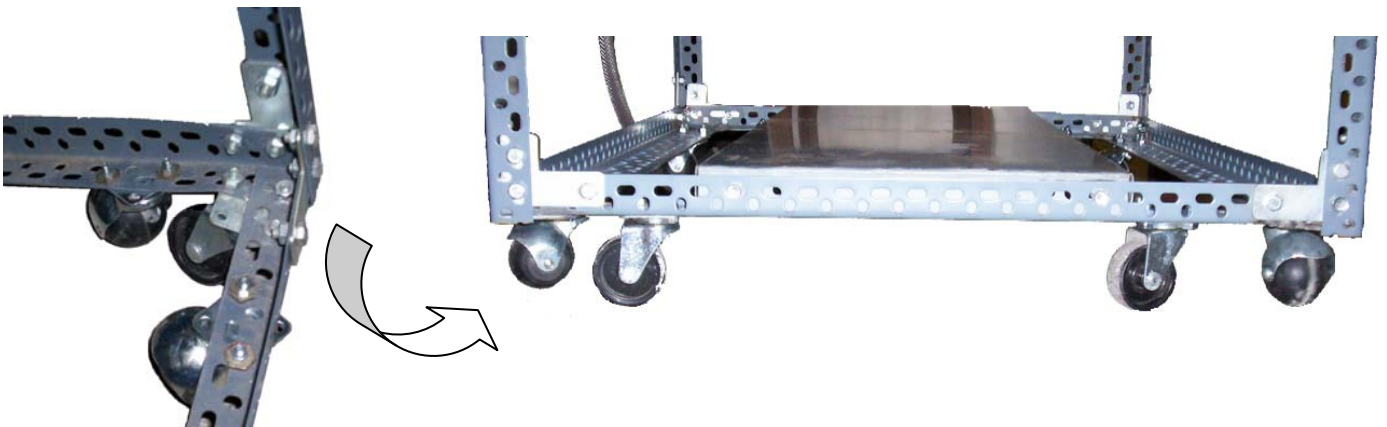
Dado que en el laboratorio de control no se cuenta con una entrada de OSE se necesita trasladar la planta desde el mismo al laboratorio de proyectos (o a donde se al opere). En los primeros intentos de movilizarla se notó lo siguiente:

- El soporte de la estructura estaba vencido a causa del peso de la planta, esto impedía un fluido movimiento de la misma.
- Las ruedas utilizadas se doblaron hacia el interior de la estructura, imposibilitando su rotación, motivo por el cual se trababan.

En primera instancia como medida provisoria se agregó un juego de 4 ruedas extras. Esta solución no fue suficiente, a causa del torque aplicado a los perfiles ocurría que los mismos se doblaban hacia el exterior.

Consultados los fabricantes de los perfiles auto perforados (Fumaya, perfiles originales 3x3 chapa 16), se nos informó que los mismos no estaban diseñados para esa función, son construidos para estanterías fijas. Además la distancia entre el soporte del intercambiador y el piso era mayor a la permitida (80cm Vs 40cm), por lo cual la base se vencería indefectiblemente. Por este motivo se fabricó una nueva base con perfiles de mayor resistencia (5x3cm chapa 14) soldados en sus esquinas y con las ocho ruedas disponibles colocadas.

Luego de enderezar los perfiles originales estos fueron colocados, como refuerzo, en la zona más débil: a la mitad de distancia entre la base y el piso. Además de eso, en la base se agregó un estante para colocar la PC y evitar que los perfiles de los laterales se doblen.



2.9 Problemas encontrados en el rearmado final

En las pruebas realizadas a la planta se encontraron algunos inconvenientes que se solucionaron, entre ellos:

- El cable de alimentación de la resistencia del tanque estaba desconectado, el mismo fue soldado y sujetado correctamente.
- Se encontraron obstrucciones en las cañerías, principalmente en el intercambiador de calor (sección de agua caliente). Se limpiaron las conexiones (obstruidas debido a la acumulación de sarro y óxido) y se conectó la entrada del intercambiador directamente a la línea de agua para así obtener una presión de entrada suficiente para limpiarlo.
Luego de lidiar con continuas disminuciones del caudal debidas a obstrucciones se decidió cambiar las cañerías afectadas por el óxido del tanque viejo y del intercambiador.
Finalmente todas las cañerías se colocaron por dentro de la estructura para que de ésta manera no se engancharan con objetos en su traslado.
- Se detectaron pérdidas de agua principalmente porque varias de las conexiones carecían de teflón. Además unas colillas estaban pinchadas (la goma interna se había reseca) lo que motivó su reemplazo. Actualmente no se notan pérdidas
- Debido a reparaciones que se llevaron en las líneas de agua del IIE la presión de agua del laboratorio de proyectos quedó muy disminuida (desde Junio al día actual). Además de eso se encontraron obstrucciones en la línea de alimentación de la planta debido a mugre que quedó acumulada en las cañerías del laboratorio luego de los dichos trabajos. Todo esto motivó el traslado de la planta hacia el Taller del instituto, donde se cuenta con una entrada de agua directa de OSE, con buena calidad de agua (“limpia”) y presión suficiente.

3 Interfaces

3.1 Introducción

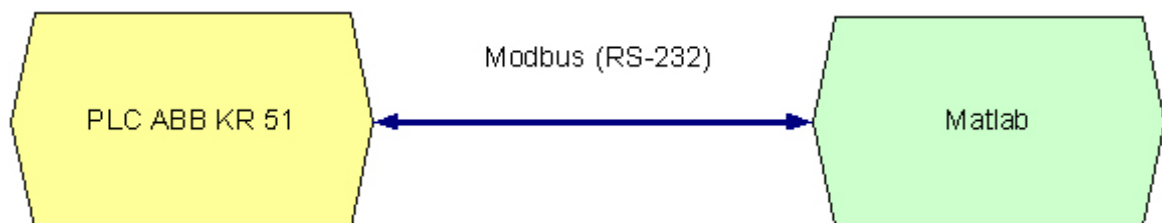
Una de las principales dificultades que presentaba el proyecto era poder realizar la comunicación en “tiempo real” entre el PLC ABB y Matlab, además de:

- tener que crear un sistema que almacene los valores históricos de las distintas variables de interés de manera óptima
- lograr una HMI (Human Machine interface) amigable, fácil de configurar y que pueda ser reutilizable con otros dispositivos inteligentes.

El primer paso para poder vencer estas dificultades fue estudiar las distintas posibilidades de comunicar información que tienen incorporados el PLC ABB KR 51 y Matlab. Las conclusiones después de este estudio preliminar fueron las siguientes:

- 1- El PLC ABB solamente puede comunicar información mediante Modbus (ver Comunicación Modbus, Anexo A), limitando en primera instancia las opciones a elegir.
- 2- Matlab es capaz de manejar una amplia gama de formas de comunicación de información, entre ellas TCP/IP, OPC, Serial.

A continuación se muestra como podría ser la comunicación entre el PLC ABB y Matlab de acuerdo a la información recabada.



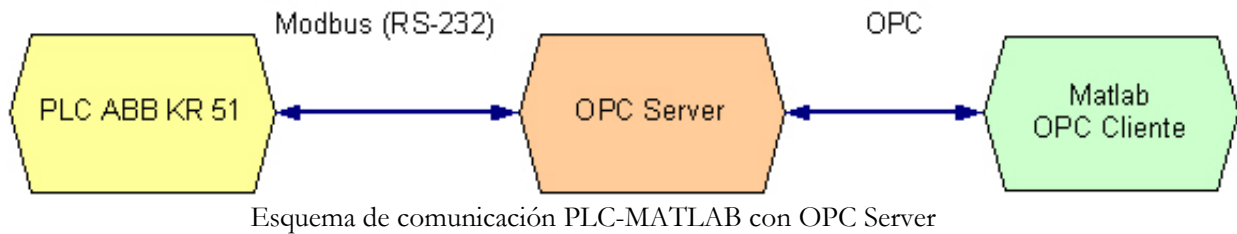
Esquema de comunicación PLC-MATLAB vía Modbus

Con este esquema de comunicación era prácticamente imposible cumplir con los requerimientos de tener un sistema que pueda almacenar los históricos de las variables de forma sencilla, tener un HMI amigable y fácil de configurar. Esto es debido principalmente a la complejidad para realizar menús interactivos y a limitaciones propias de Matlab (en lo referido a comunicación), pues este programa a pesar de ser una gran herramienta no fue pensado para su uso en la industria.

Por el motivo anterior se decidió buscar nuevas opciones en la forma de comunicar información entre el PLC y Matlab. Con tal fin se empezó por estudiar las distintas formas de comunicar información de los sistemas industriales.

Una de las formas ampliamente utilizada para comunicar información de los distintos dispositivos con los sistemas HMI industriales es el Standard OPC (Ver OPC: Anexo B). Actualmente la gran mayoría de los dispositivos inteligentes son capaces de comunicar información por medio de OPC.

Con el Standard OPC el esquema de comunicación quedaría de la siguiente forma:



En este esquema se cumplen todos los requerimientos correspondientes a la comunicación entre Matlab y los dispositivos inteligentes. Pero aún quedan por levantar las restricciones de una interface HMI amigable y crear un sistema que almacene valores históricos de variables de forma óptima.

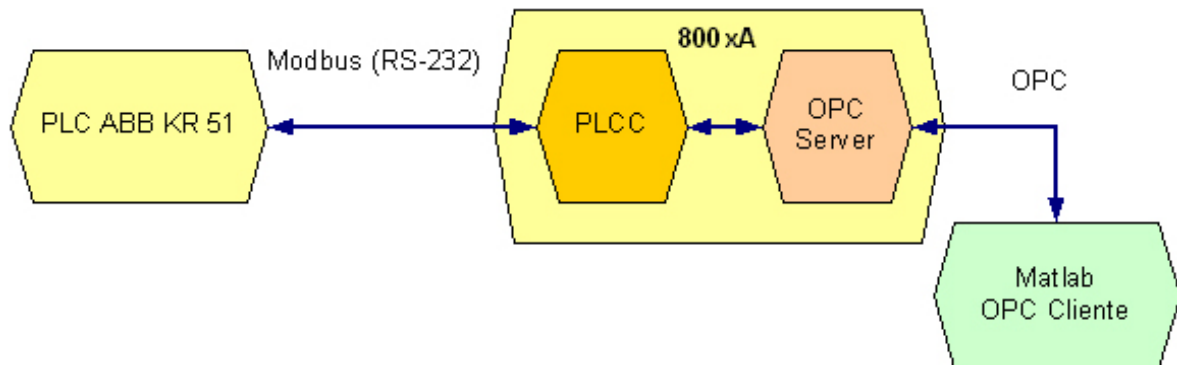
Después de investigar como los sistemas industriales resuelven el problema de tener una interface amigable para el operario y almacenar los históricos de variables se concluyó que se debía incorporar al esquema de comunicación un sistema SCADA o DCS.

El IIE cuenta con dos sistemas SCADA con las siguientes desventajas:

- 1- **FIX:** Este sistema SCADA tiene la gran desventaja de que su licencia permite correr el software por un período acotado de tiempo.
- 2- **SCADA Portal ABB:** La licencia que tiene no es legal y además tiene problemas en el despliegue de los históricos (no se puede cambiar la resolución). Este SCADA fue el utilizado por el proyecto previo y ya desde aquel entonces el despliegue de históricos era un problema (por este motivo se definió el objetivo de cambiar el SCADA).

Para sortear los inconvenientes de los sistemas SCADA con los que contaba el IIE se recurrió a ABB, ésta empresa donó un sistema DCS, de última generación, utilizado en el mundo entero. Este sistema (DCS), llamado 800xA, cumple ampliamente con todos los requerimientos necesarios de la interface HMI que se requería para este proyecto además de agregar una gran cantidad de herramientas útiles a la hora de armar dicha interface.

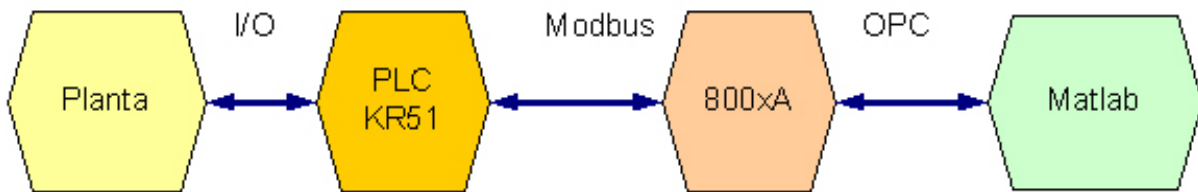
Con el DCS 800xA el esquema de la comunicación se transforma en el siguiente:



Esquema de comunicación PLC-DCS-MATLAB

3.2 Definición de las distintas interfaces y funcionalidades

Una vez solucionado el problema de la comunicación entre el PLC ABB KR51 y Matlab se prosiguió con la definición de las distintas interfaces y funcionalidades de las diferentes partes del sistema. A continuación se muestra un esquema:



Para poder organizar el trabajo se dividió el sistema en cuatro partes bien diferenciadas.

- 1- **Planta:** Es la planta piloto diseñada y construida para simular un proceso industrial. La misma cuenta con instrumentos y actuadores diseñados y contruidos para cumplir esta función.
- 2- **PLC KR 51:** Su principal función es adquirir la información de los instrumentos de medida y controlar los actuadores por medio de I/O. Como los instrumentos y actuadores no son industriales el PLC KR51 también debe adaptar las señales de estos. Por ejemplo los caudalímetros están armados por sensores de presión diferencial que no están pensados para ser usados en la industria, por lo que las medidas de estos sensores están afectadas por offsets (que pueden variar). También ocurre que las medidas de estos sensores respecto al caudal se puede suponer cuadrática, por lo tanto hay que afectarlas por una raíz cuadrada. Con las válvulas sucede algo muy parecido, si bien quedaron bastante lineales se adicionaron bloques que linealizan su curva característica, además de otros bloques para reducir el efecto de histéresis. A grandes rasgos las funciones del PLC son:
 - comunicar la información adquirida desde los sensores al DCS 800xA
 - enviar comandos, que llegan desde el sistema, a los actuadores de la Planta.
- 3- **Sistema 800xA:** Entre las funciones del sistema DCS están:
 - i. Actuar como SCADA
 - ii. Despliegue de gráficos que muestren la planta.
 - iii. Almacenamiento y despliegue de históricos en forma de tendencias.
 - iv. Despliegue de datos en tiempo real
 - v. Ser el nexo entre el PLC y Matlab.
 - vi. Ser el soporte de la HMI diseñada
- 4- **Matlab:** Es la plataforma en la cual se hacen las aplicaciones de control avanzado ya que cuenta con una variedad de herramientas muy poderosas a la hora de crear, simular y testear sistemas tanto lineales como no lineales. Para este proyecto en particular se decidió trabajar con Simulink (ver Anexo D) que

es un entorno gráfico de programación propio de Matlab que tiene una gran cantidad de ventajas respecto a la programación clásica de Matlab (programación en archivos con extensión .m).

Para definir los datos que se intercambian entre PLC-DCS-Matlab se diseñó una planilla Excel. En la misma está bien definida la función y rangos de todas las variables que intervienen en cualquier parte del sistema así como su interacción. Ésta se actualizó a medida se fue avanzando en el proyecto.

3.3 PLC

El PLC utilizado para este proyecto es el 07KR51 (familia AC31, serie 50) de la empresa ABB. Se utiliza una CPU (unidad central) y dos extensiones analógicas de entrada y salida: XM06B5 que cuentan, cada uno, con cuatro entradas y dos salidas analógicas.

3.3.1 Resumen

Inicialmente el objetivo del proyecto era implementar el controlador de temperatura en el PLC. Luego de avanzar en el estudio del mismo, y conocer más a fondo las características del PLC y de las herramientas que se disponen se decidió implementar dicho controlador en MATLAB. Esto se debe a que se consideró una mejor alternativa que consiste en centralizar el control en Matlab de manera de lograr una mayor flexibilidad a la hora de hacer cambios. De ésta manera se delegaron al PLC las funciones de adquisición, envío y adaptación de señales, como también la implementación de ciertos procedimientos en caso de alarma.

A grandes rasgos se utiliza al PLC como interfaz entre el DCS y la Planta (sensores y actuadores).

3.3.2 Programación

3.3.2.1 Configuración de la comunicación PLC-DCS

Por defecto la comunicación del DCS con el PLC se da a 9600bps (Half Duplex). Por este motivo la cantidad de Bytes que se pueden transmitir por segundo es de

$$\left(\frac{9600bps}{8}\right) = 1200Bps.$$

En la comunicación MODBUS la información transmitida por el maestro contiene lo siguiente⁹:

- La dirección de MODBUS del esclavo interrogado (1 byte)
- El código de función que define la petición del maestro (1 byte)
- La información a intercambiar (N bytes)
- El código de control CRC16 (2 bytes)

Por lo tanto, si las transferencias se realizan de a palabras, para transmitir (leer o escribir) 16 bits (una Word) son necesarios 4 bytes extras, dando una transmisión efectiva

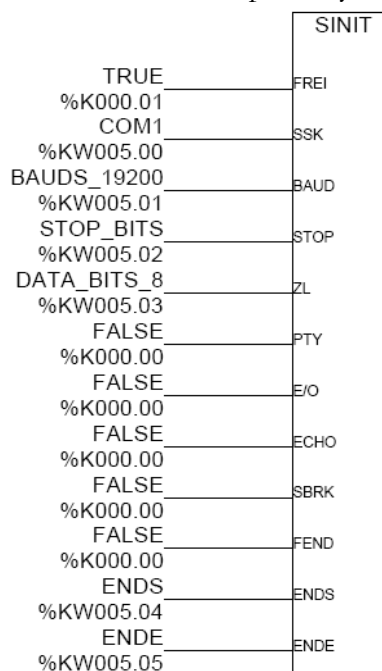
⁹ (ABB, 2001)

máxima de $\frac{1200Bps}{6Bytes} = 200 \text{ palabras/seg}$. Si las mismas son por bloques en una única

transferencia se pueden intercambiar un máximo de 100 datos tipo palabra¹⁰. Así se logra un mejor throughput (cadencia de datos), pues hay menos sobrecarga de cabezales y se pueden intercambiar casi 1200 Bytes/seg (o expresado de otra forma: 600 palabras/seg). En las primeras pruebas a 9600 Baudios las señales se retrasaban apreciablemente hasta un punto tal que los servicios de comunicación consumían tantos recursos que la PC no respondía.

En un inicio (con transferencias por palabra), dado el número de señales que se comunican entre MATLAB y el PLC y la cadencia con la que lo hacían, el límite de 200 palabras/segundo se veía fácilmente sobrepasado, en especial cuando se pretendía realizar un control lo más aproximado a “en tiempo real” (altas tasas de refresco de datos). Por este motivo, luego de un detallado estudio del PLC con el que se cuenta, se pudo comprobar que dicha velocidad de transmisión se puede duplicar, contando de ésta manera con una transferencia efectiva máxima de 400 palabras/seg (en modo bloque casi 1200 palabras/seg). Para este fin se debió incluir en el programa del PLC un bloque de inicialización de comunicación que cuando se utilizan los valores de comunicación por defecto no es necesario incluir. Además de esta medida, ya que hay casi 90 palabras que se intercambian entre el PC y PLC, se tuvo que configurar el DCS para que el intercambio de información con el PLC fuera de a bloques de palabras. **De ésta manera tomando una tasa de refresco uniforme de 0.1 segundo, lo cual es bastante exagerado, se estarían intercambiando unas 900 palabras/seg (en un peor caso) lo que da un porcentaje de utilización de los recursos de comunicación del 75% (900*100/1200).**

Para configurar las propiedades de la comunicación de los puertos del PLC se utiliza el bloque SINIT (ver siguiente figura), en particular se utilizó para configurar la comunicación en el puerto COM1 en 19200 Baudios, con 1 bit de parada y sin chequeo de paridad.



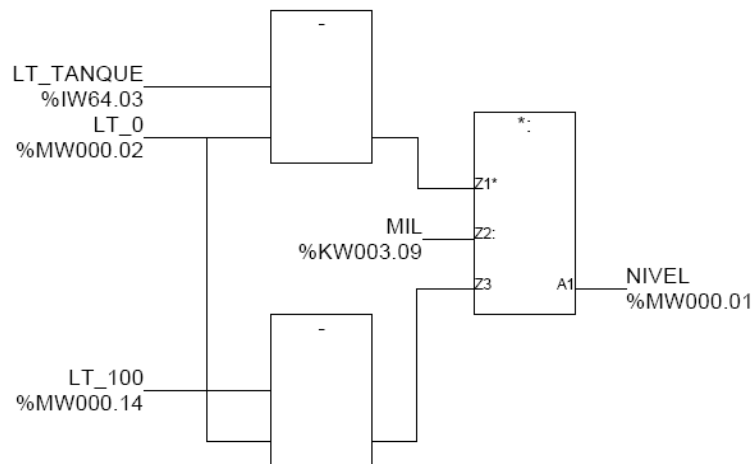
¹⁰ (ABB, 2001)

3.3.2.2 Nivel del Tanque

El PLC mide un valor de voltaje o corriente que se representa por un número entero entre 0 y 32767. La relación que existe entre el valor real de la variable y lo medido por el PLC es lineal, por lo que si se determinan experimentalmente dos puntos de la recta se determina la correspondencia para cada valor medido por el PLC.

Para el caso del nivel de agua en el tanque se decidió, escalar en el PLC, su lectura de 0 a 1000 de manera de tener una buena resolución. Esto se debe a que éste resultado se envía al DCS (y posteriormente a Matlab) para la realización del control del lazo de nivel.

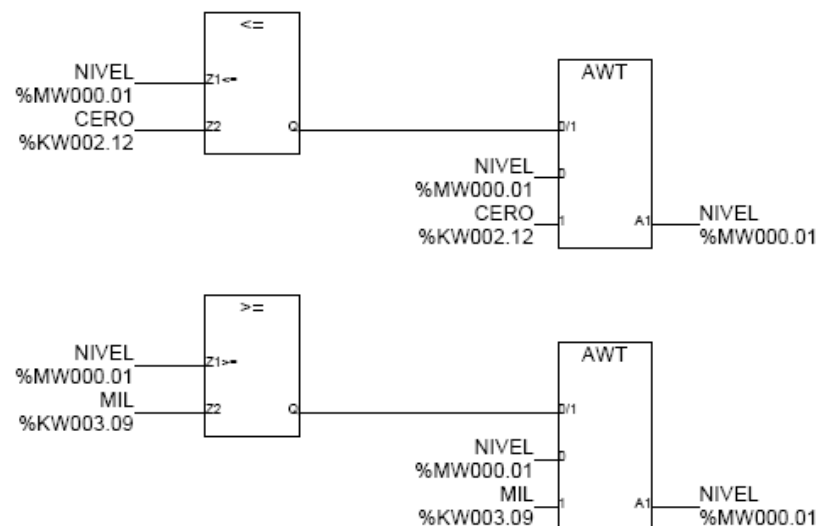
El siguiente es el bloque implementado:



Básicamente lo que se hace es una interpolación lineal. Una vez realizada la calibración de los sensores, las variables LT_0 y LT_100 quedan escritas con la lectura del sensor de nivel en la referencia y en el tope que se fije (LT_100) respectivamente. Esto es correcto pues en el rango de utilización del sensor diferencial el mismo trabaja en su zona de comportamiento lineal y además la relación entre ΔP y nivel en el tanque es también lineal.

Además se diseñó una lógica adicional para el caso de que el escalado del Nivel sea menor que 0 o mayor que 1000, pues si se dan esos casos, en el DCS y Matlab ocurren errores que enlentecen el “tiempo real”. Para ello se hace lo siguiente:

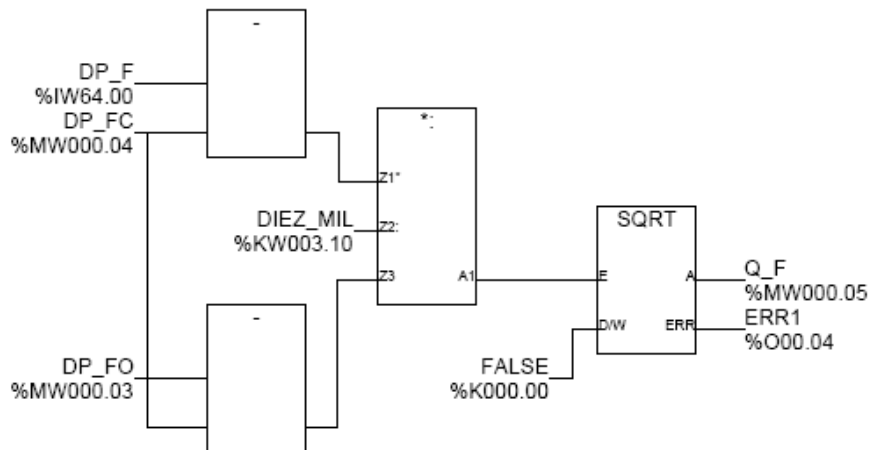
- Si $NIVEL \leq 0 \Rightarrow NIVEL = 0$
- Si $NIVEL \geq 1000 \Rightarrow NIVEL = 1000$



3.3.2.3 Caudalímetros

Como convención se optó por que los caudales sean escalados (en el PLC) de 0 a 100 ya que no interesa que presenten mucha resolución pues sólo se utilizarán para ser desplegados en pantalla.

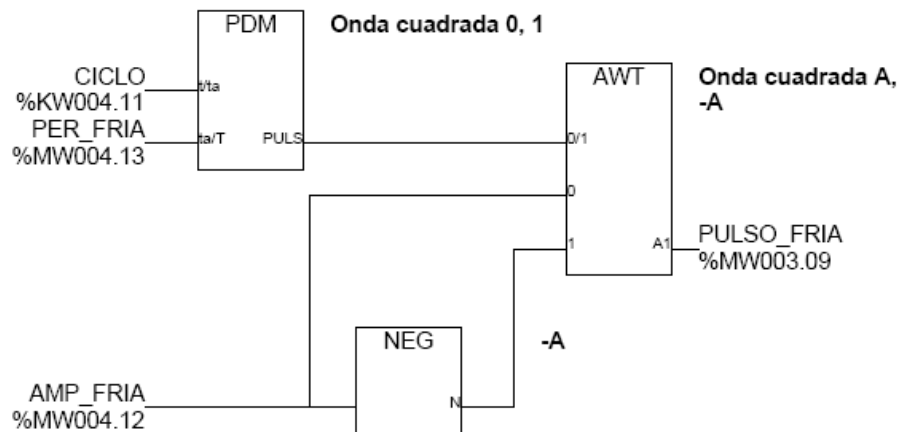
Para este fin se diseñó el siguiente bloque utilizado tanto para el caudal de agua fría como para el de agua caliente:



En una primera etapa la lectura del sensor diferencial (en este caso DP_F) se escala al rango [DP_FC, DP_FO], donde DP_FC es la lectura del sensor cuando no pasa caudal alguno y DP_FO es la lectura cuando la válvula de entrada correspondiente está abierta y pasa el caudal máximo. Luego se ajusta el resultado a un valor entre 0 y 10000 para que finalmente, al tomar la raíz cuadrada ($Q \propto \Delta P$), el caudal correspondiente varíe entre 0 y 100.

3.3.2.4 Válvulas

Para disminuir el efecto indeseable de histéresis, presente en la curva Q-CV de las válvulas de control, se decidió sumarle a cada señal que comanda dichas válvulas una onda cuadrada de período y amplitud variable, parámetros que se pueden setear desde el DCS. Para la válvula de agua fría la señal puede diferir de la de la válvula de agua caliente (en particular porque no son idénticas) hecho que fue contemplado agregando dos generadores de onda cuadrada como el que se muestra a continuación:

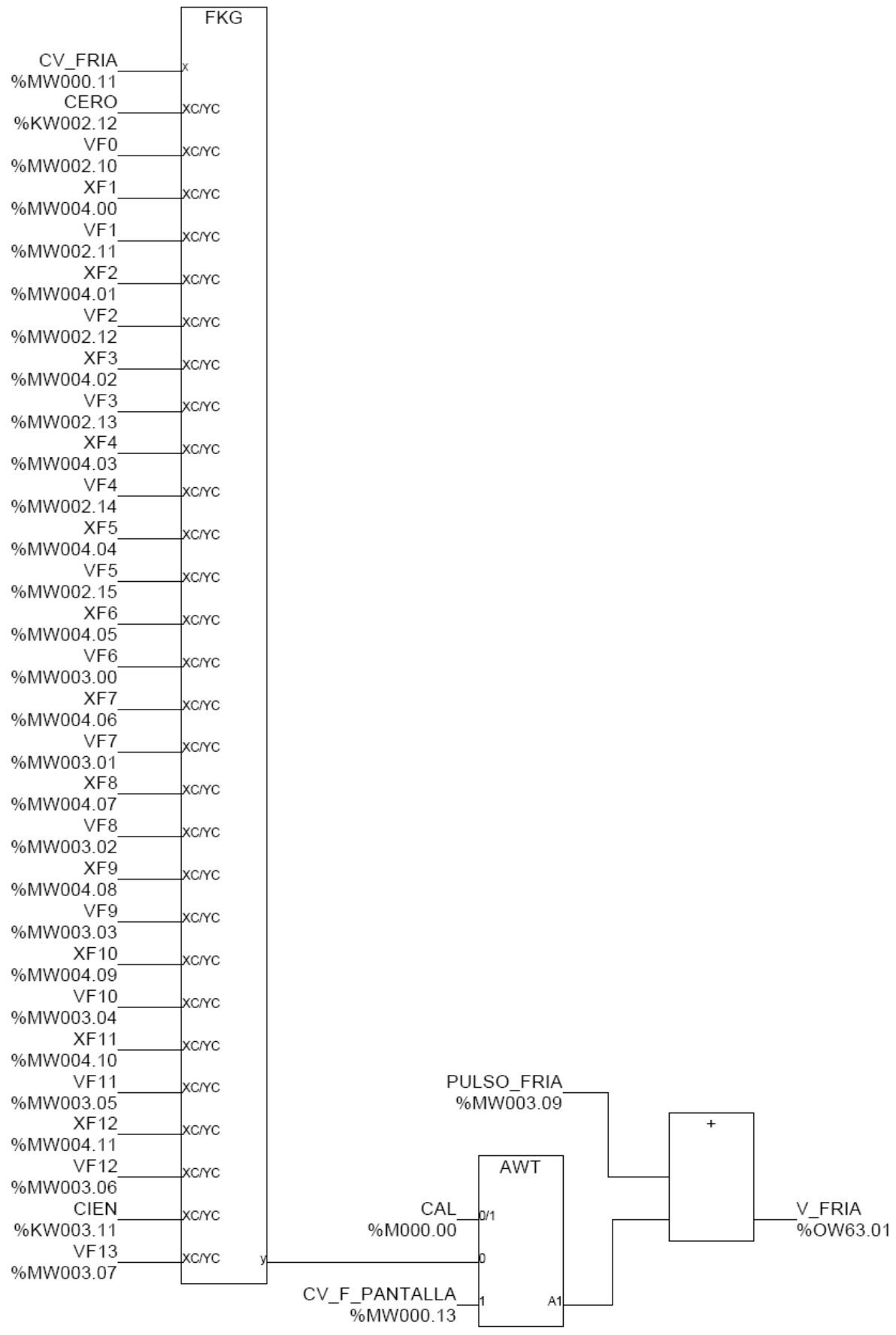


Otro aspecto que se tuvo que contemplar es el hecho de que no son válvulas 100% lineales. Por este motivo se decidió incluir bloques de linealización (ver bloque FKG de la siguiente figura), de esta forma se asegura un mejor desempeño de las mismas.

Las variables de entrada del bloque FKG se actualizan en el momento de la calibración, ingresándose desde el DCS. Lo que se hace es, a partir de una apertura de la válvula deseada (CV_FRIA en este caso), que varía entre 0 y 100, realizar una interpolación lineal de ‘apertura válvula’ VS el ‘voltaje de salida a la placa Driver de los Servos’.

Antes de la salida a cada válvula de control se incluye un selector de forma de manejar las válvulas directamente desde DCS (cuando se ejecuta la calibración) y además se le suman a la salida una onda cuadrada de alta frecuencia (su período se sitúa en 300ms) y de baja amplitud (aproximadamente entre un 5% y un 35% del rango) para de ésta manera disminuir la histéresis que presentan.

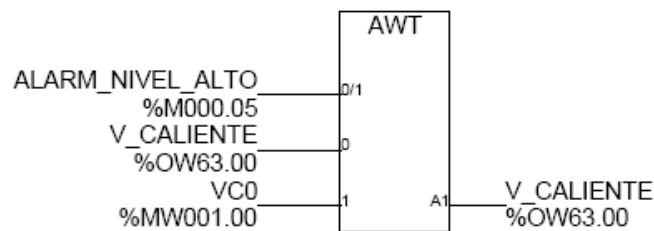
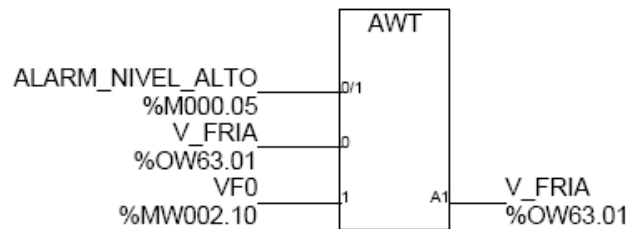
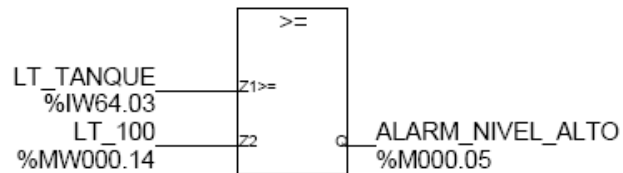
Vale la pena mencionar que se eligieron bloques FKG que interpolan entre 14 puntos (lo que puede resultar exagerado) para que el control sea más exacto.



3.3.2.5 Alarmas Hardware

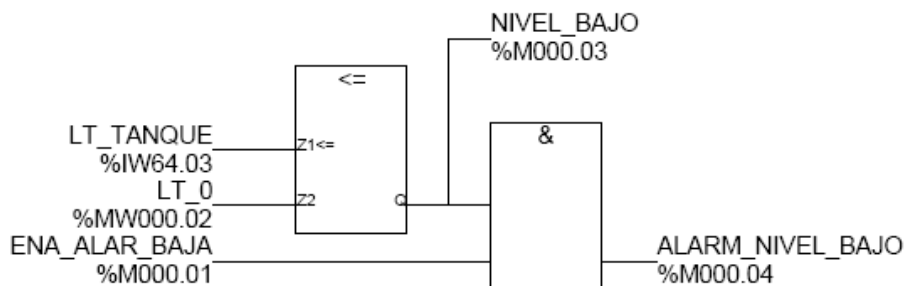
Nivel Alto

Debido a malas experiencias, por no tener previsto en una etapa inicial posibles fallos que terminaran con el sobrepaso del nivel máximo tolerado por el tanque, se decidió la implementación de la alarma de nivel alto, la cual al generarse cierra las entradas de agua caliente y fría.



Nivel Bajo

De forma análoga, se implementó una alarma para el caso de que el nivel sea menor al mínimo, condición en la cual solamente se deshabilita el encendido de la resistencia dentro del tanque.

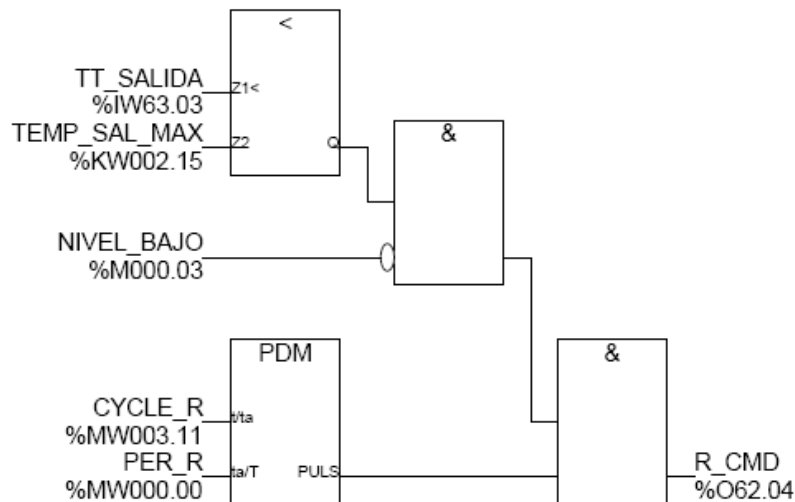


3.3.2.6 Resistencia del Tanque

Cuando se proyectó la planta, se decidió la inclusión de una resistencia en el tanque para compensar las pérdidas de calor, especialmente las resultantes de que el tanque sea abierto a la atmósfera. Además hay pérdidas de calor debidas a la carencia de aislación térmica de las cañerías de agua caliente.

Como la resistencia con la que se cuenta es de 1200W, ya desde el proyecto anterior, se decidió encenderla y apagarla de forma periódica, con un período de 2 segundos. Además el encendido de la misma se produce a partir de que el nivel en el tanque supera aproximadamente el 60% de la altura respecto a su base (condición de que la resistencia se cubra de agua).

Se debe mencionar que también se incluyó lógica de manera tal que la resistencia encienda sólo si la temperatura es menor que cierta temperatura de salida máxima admisible y el nivel mayor que el nivel de referencia.



Los parámetros para configurar el tren de pulsos que comanda la resistencia son escritos desde el DCS. Desde allí se puede variar el tanto el período como el Duty Cycle del mismo. Los parámetros nominales calculados en el proyecto anterior son Período = 2 segundos y Duty Cycle= 50%.

3.4 Comunicación DCS - PLC

Esta comunicación utiliza el protocolo de **MODICON MODBUS RTU** para intercambiar datos entre el PLC y el DCS.

3.4.1 Modo de direccionamiento:

Para linkear las variables en el PLC con las del DCS se hace imprescindible direccionar correctamente las mismas. Aquí es el usuario el que debe encargarse de escribir lugares de memoria habilitados para su escritura o sea no escribir entradas o salidas, además es el responsable de indicar si lee o escribe un BIT, WORD o DWORD y en qué dirección Modbus esto comienza.

Aquí se presenta el modo de direccionamiento¹¹ que se utiliza:

VAR 00.00 \longrightarrow ADDR 0
(VAR = tipo I, O, S, M, IW, OW, MW, KW) (Dirección de la primera variable seleccionada en decimal)

$$\text{VAR XX.YY} = \text{ADDR 0} + (16 * \text{XX}) + \text{YY}$$

(VAR = tipo MD, KD) (Dirección de la primera variable seleccionada en decimal)

$$\text{VAR XX.YY} = \text{ADDR 0} + (32 * \text{XX}) + (2 * \text{YY})$$

Ejemplo: encontrar la dirección MODBUS de las variables O62.15, M232.01 y MD002.07:

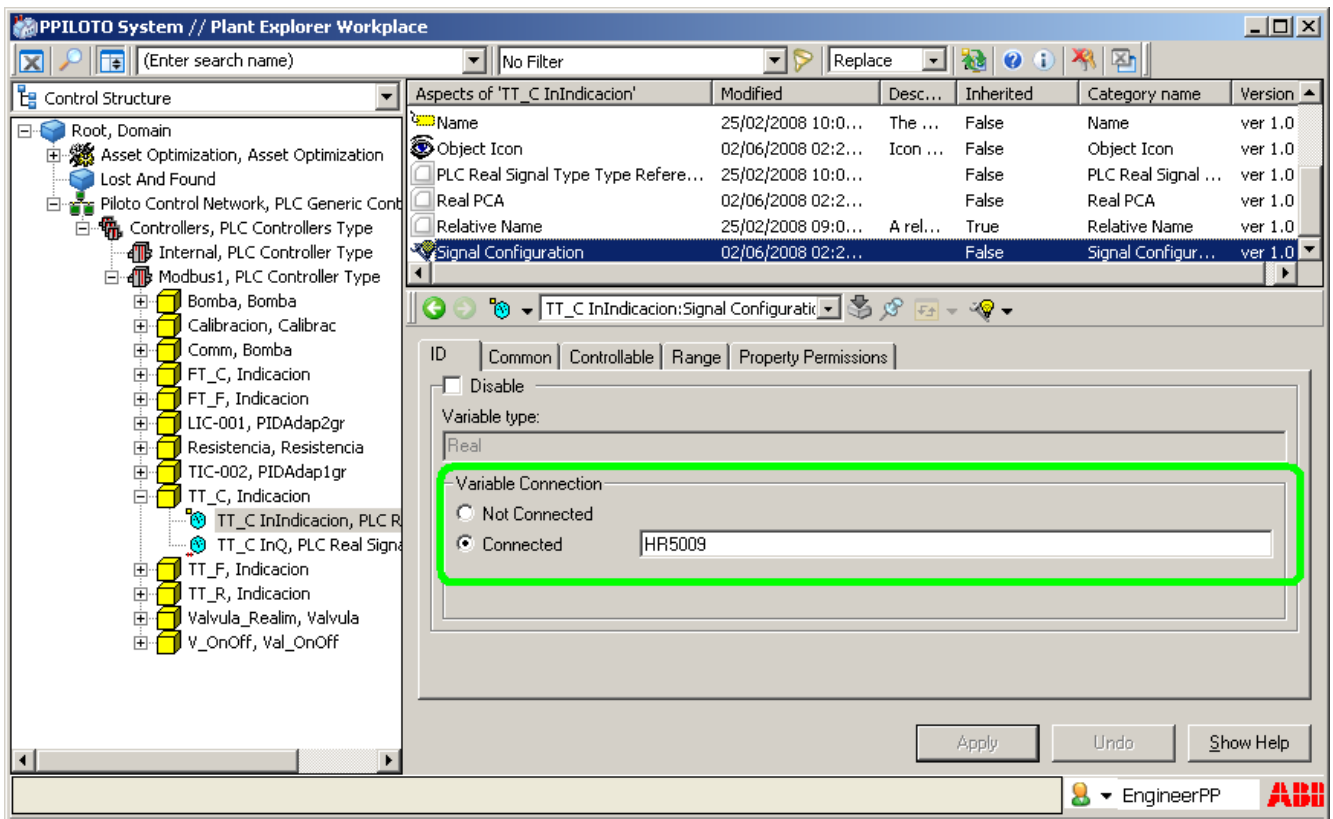
$$\begin{aligned} \text{O62.15} &= 4096 + (16 * 62) + 15 = \mathbf{5103} \\ \text{M232.01} &= 8192 + (16 * 232) + 1 = \mathbf{11905} \\ \text{MD002.07} &= 4000 + (32 * 2) + (2 * 7) = \mathbf{16462} \end{aligned}$$

En el DCS se asignaron a las variables que se desean comunicar su dirección en el PLC según las siguientes reglas:

IW: HR[dirModbus + 4001]
Ej: IW63.00 = HR[1008 + 4001] = HR5009
MW: HR[dirModbus + 4001]
Ej: MW00.01 = HR[8193 + 4001] = HR12194
M: CA[dirModbus]
Ej: M00.00 = CA[8192] = CA8192
O: C[dirModbus + 1]
Ej: O62.00 = C[5088 + 1] = C5089

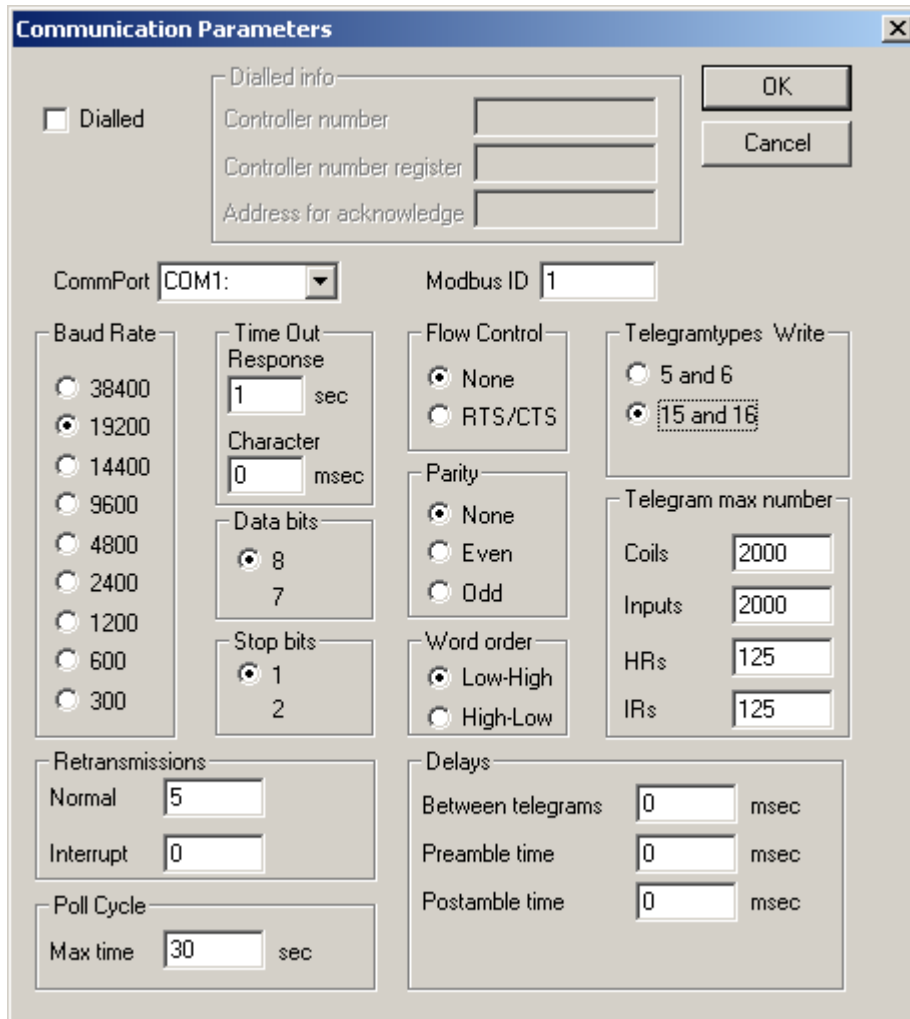
¹¹ (ABB, 2001)

En el ejemplo a continuación se muestra la conexión de la señal de temperatura del agua caliente, con su respectiva variable en el PLC (IW63.00). Para ello se seleccionó en la estructura de control el objeto 'TT_C', del tipo 'Indicación'. Luego como lo que se lee es una temperatura se debe acceder a la señal 'InIndicacion' y se selecciona el aspecto 'Signal Configuration'. Finalmente en la pestaña 'ID' se configura la señal como 'Connected' y a continuación se escribe la dirección de la variable utilizando las reglas de direccionamiento explicadas.



3.4.2 Configuración de la comunicación

Para que la comunicación entre el sistema y el PLC fuera exitosa se tuvieron los cuidados de configurar la comunicación en 19200bps y que las funciones de transferencia se realicen en bloques (*) de palabras (funciones 15 y 16 del protocolo Modbus) de forma de aprovechar al máximo la capacidad de transferencia de datos. Sino, por sobrecarga de cabezales (“overhead”), se desperdiciaría mucho ancho de banda.

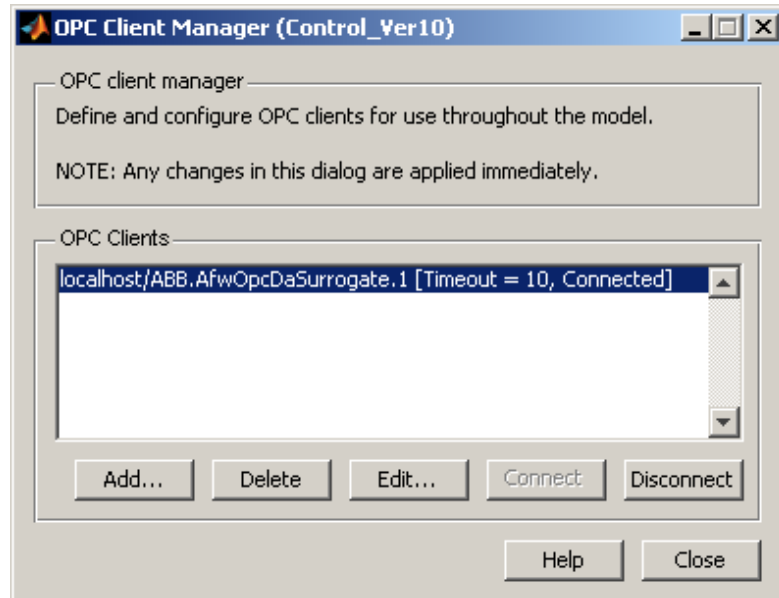


Configuración de la comunicación con el PLC

(*) Para que las transferencias efectivamente se realicen en bloques se deben usar las variables adyacentes en el PLC, de otra manera pierde la gracia utilizar éstas funciones. En la programación del PLC se tuvo este hecho en consideración.

3.5 Matlab-DCS

Para comunicar variables en el DCS con variables en Matlab se debió configurar y conectar, desde Simulink, el cliente OPC de ABB:



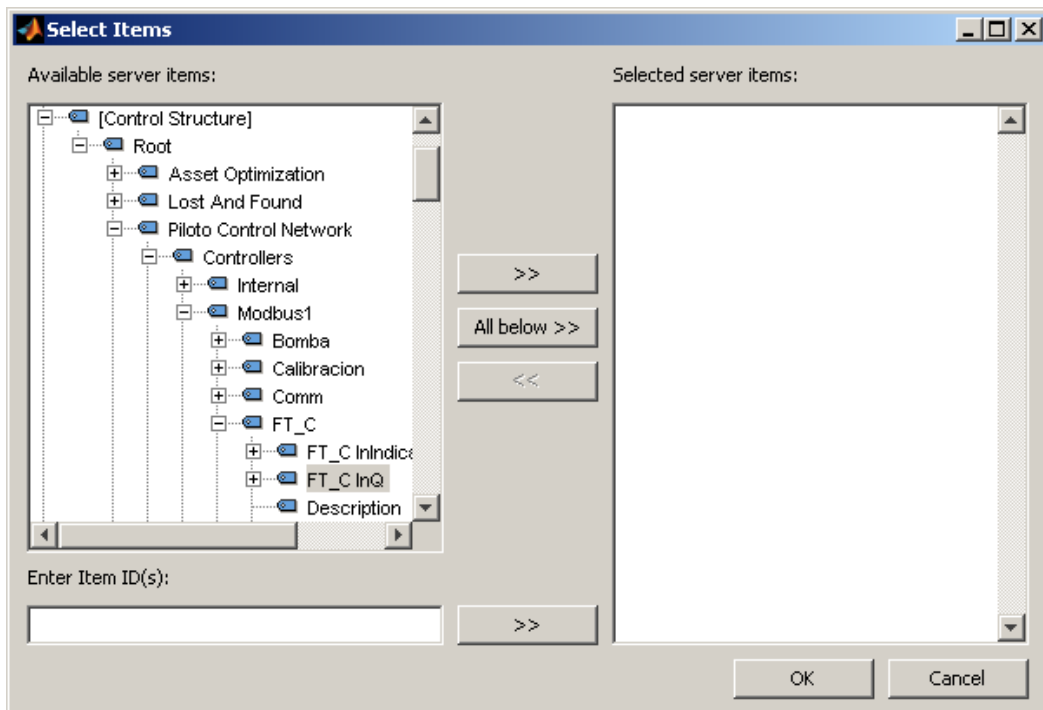
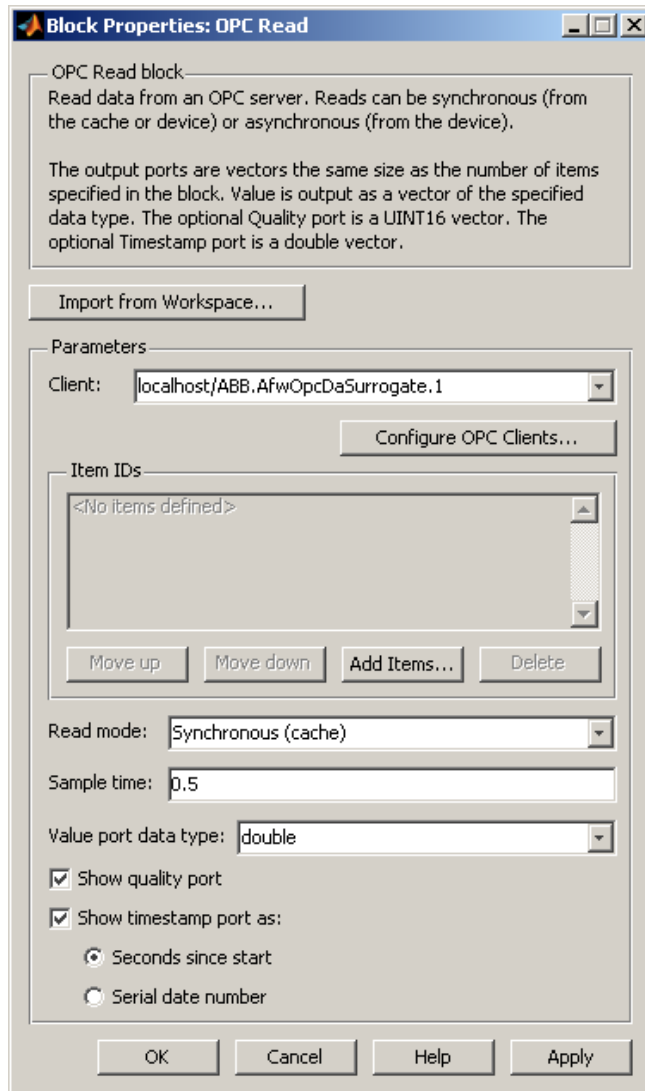
Configuración del Cliente OPC

Una vez finalizada la configuración del OPC sólo resta buscar las variables que se desean conectar y agregarlas. La única consideración que hay que tener en cuenta es el tiempo de “sample”, en este proyecto se utilizó, para variables de importancia, un tiempo de 100ms (‘sample time’ = 0.1):

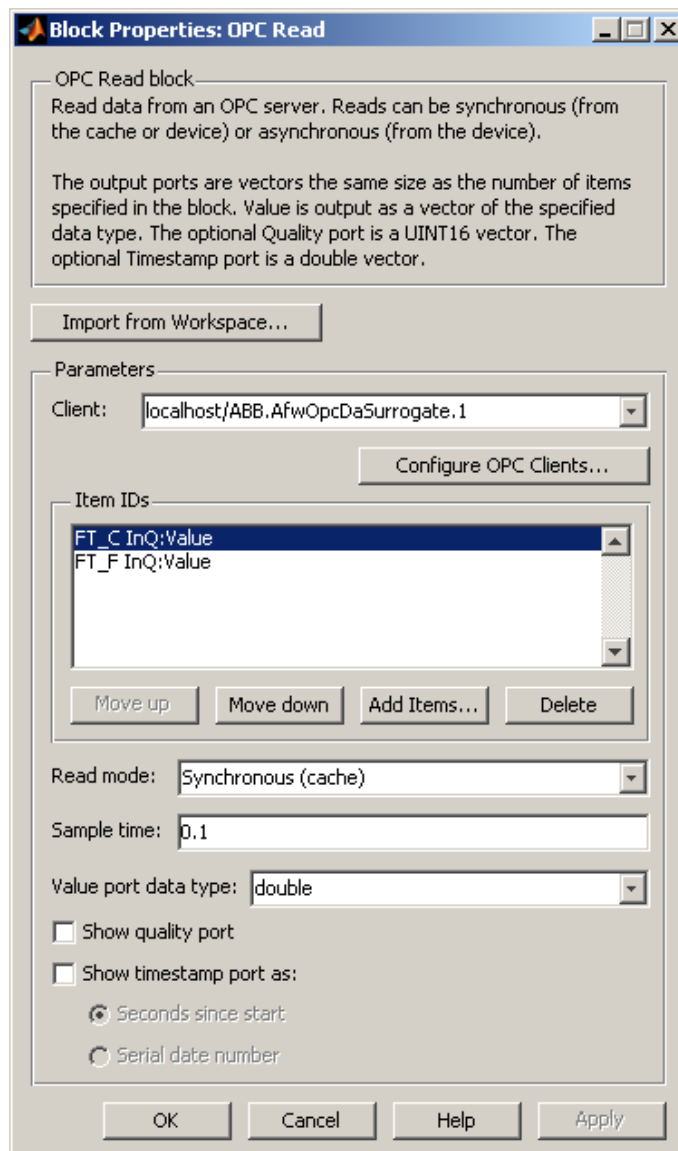
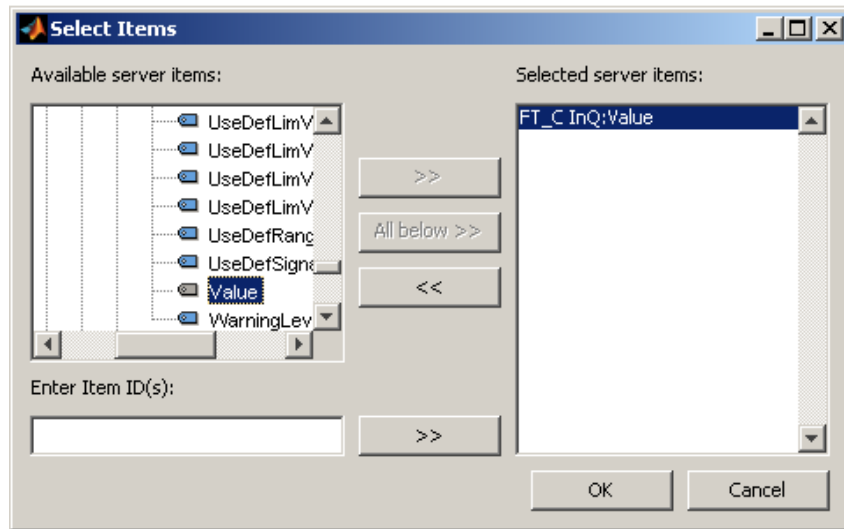
Ejemplo de conexión de una variable:

A continuación se presenta un ejemplo de conexionado de una variable desde el DCS a Simulink:

Para leer desde Simulink el caudal de agua caliente primero se debe agregar un bloque ‘OPC Read’. Después se debe clicar en ‘Add Items’ y buscar la ruta, en la estructura de control, de la señal en cuestión, seleccionar el atributo ‘Value’ y agregarlo a la lista de ‘Selected server items’.



Finalmente, luego de agregadas las señales que el bloque de OPC lee, se debe de dar 'OK' y con eso se terminó de configurar la comunicación:



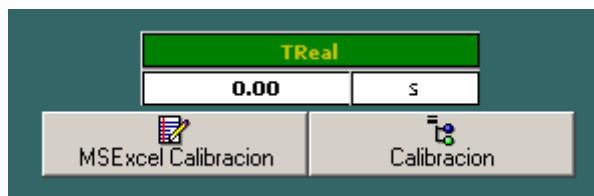
Consideraciones:

Una vez realizada la programación del DCS, Matlab y el PLC, se procedió a su prueba. Aquí surgió un primer error de comunicación que hacía aumentar la memoria utilizada por el servicio “AdsCsCommSrv.exe” de manera tal que al llegar a los 120Mb ocupados en RAM, el sistema se colgaba. Esto era debido a que:

- la velocidad de comunicación PLC-DCS estaba inicialmente seteada en 9600bps (configuración por defecto) y las transferencias no estaban configuradas para realizarse en bloques
- la tasa de refresco utilizada en OPC era demasiado grande, por eso se tomó la decisión de que la tasa de refresco máximo para las señales importantes sea: (10 veces)/seg

En definitiva éste problema se solucionó con las siguientes medidas:

- PLC-DCS: se configuró la velocidad de comunicación PLC-DCS en 19200bps y que la transferencias de datos se realizara de a bloques.
- DCS-MATLAB. se corrigió la tasa de refresco en OPC. Se estaba trabajando con frecuencias muy altas que en la simulación funcionaban pero en la práctica no, provocando que se sobrecargara el servicio de comunicación. Por dicho motivo se tomó la decisión de que la tasa de refresco máximo para las señales importantes sea de 10veces/segundo. Además actualmente en la pantalla principal del DCS hay una indicación que muestra el error que se comete respecto a lo que sería “tiempo real”



Indicación del Error respecto a la simulación en Tiempo Real

Por lo general el error es muy cercano a cero segundos, pero si por ejemplo se corren varias sesiones de Matlab al mismo tiempo se puede observar un retraso (en ese caso el error es negativo y se expresa como: -(tantos) segundos).

4 Calibración de Instrumentos

Como un objetivo secundario del proyecto se propuso mejorar la calibración poco flexible de los instrumentos de la planta, en la cual, una vez obtenidas las constantes por medio de experimentos, se debían programar las mismas en el PLC.

En este proyecto se propuso realizar una calibración automática de los instrumentos. Luego de probado el código de la misma, por las razones que se expondrán más adelante se decidió que la calibración no fuera automática. Para compensar este hecho se crearon un faceplate específico de calibración (una pantalla de calibración), una hoja de Excel (dónde se guardan los resultados de la calibración completa y además se pueden cargar los datos directamente al sistema presionando un botón), y un procedimiento detallado de calibración de todos los sensores y actuadores.

4.1 Auto Calibración

4.1.1 Introducción

Las condiciones en las que se realizan los experimentos pueden cambiar. Si se utiliza la planta conectada a una toma de agua con diferente presión de la prevista, se tendrían unos resultados, si la temperatura ambiente es diferente de la nominal otros y si le sumamos a ello el envejecimiento de las componentes otros resultados. Por lo tanto las constantes de calibración de los sensores y actuadores de la planta pueden y van a variar. Esto si no se prevé puede llevar a que a largo plazo se cometan gruesos errores en las mediciones y en las acciones.

El procedimiento de calibración manual completa es bastante tedioso y por este motivo se decidió hacer un programa, que con la asistencia de un operario, la lleva a cabo en forma ordenada (en su versión más simplificada) en menos de 15 minutos. Sin embargo luego de varias pruebas se llega a la conclusión que una buena calibración demoraría cerca de dos horas.

4.1.2 Procedimiento de Calibración

En este punto se describen los pasos que, dejando a un lado momentáneamente el programa de autocalibración, se deberían seguir en una calibración.

Se asume que esta todo en un estado correcto (presión de OSE constante sin pérdidas en la toma, válvulas manuales abiertas, etc.).

1- Calibrar el 0 % y 100% de las válvulas

Válvula de agua Fría:

- a. Dejar la válvula de agua fría en una posición intermedia (ni abierta ni cerrada)
- b. Abrir de a pasos la misma
- c. Cuando la lectura del sensor diferencial de la línea de agua fría se estabilice:
 - i. Registrar el valor del instrumento como DP_FO (Delta de presión del agua fría con válvula abierta (Open)) y registrar el voltaje de comando del servomotor como V_F0
 - ii. Volver a ubicar la válvula a una posición intermedia
- d. Cerrar de a pasos la válvula
- e. Cuando la lectura del sensor diferencial de la línea de agua fría se estabilice:
 - i. Registrar el valor del instrumento como DP_FC (Delta de presión del agua fría con válvula cerrada (Closed)) y registrar el voltaje de comando del servomotor como V_F100
 - ii. Dejar la válvula cerrada

Válvula de agua Caliente: Procedimiento análogo válvula de agua fría.

2- Calibrar el sensor de nivel manualmente

- a. Empezar a vaciar el tanque, cuando el nivel dentro llegue a la posición de referencia guardar la lectura del sensor de nivel en la variable LT_0
- b. Llenar el tanque, cuando el nivel llegue al máximo deseado, guardar la lectura del sensor de nivel en LT_100

Vaciar el tanque

3- Calibración de los caudalímetros:

Caudalímetro de agua fría:

- a. Abrir al 100% la válvula de agua fría durante 30 segundos, luego volver a cerrarla.
 - i. Registrar la diferencia de nivel que se produjo durante esos 30 segundos como DL_F (Delta de nivel de agua fría)
 - ii. Calcular el caudal máximo mediante la siguiente ecuación:
$$FF_Max = \frac{DeltaLT \cdot NivelM(m) \cdot \acute{A}reTanque(m^2) \cdot 1000}{0.5 \text{ min}} \quad [Lt/min]$$
- b. Para el cálculo de cualquier caudal se debe aplicar el siguiente algoritmo
 - i. $\text{Sqrt}[(DP_F / (DP_FO - DP_FC)) * 10000]$ * FF_Max

Caudalímetro de agua caliente: proceder de forma análoga al anterior

Vaciar tanque

4- Calibrar válvulas de control

Válvula de agua fría:

- a. Abrir válvula al 25% (o algún punto cercano) y registrar la lectura del sensor diferencial de agua fría como FF_25
- b. Abrir válvula al 50% (o algún punto cercano) y registrar la lectura del sensor diferencial de agua fría como FF_50
- c. Abrir válvula al 75% (o algún punto cercano) y registrar la lectura del sensor diferencial de agua fría como FF_75
- d. Calcular OFF_25 como $(FF_{25}/FF_{Max}) * 100$
- e. Calcular OFF_50 como $(FF_{50}/FF_{Max}) * 100$
- f. Calcular OFF_75 como $(FF_{75}/FF_{Max}) * 100$
- g. Armar una función en la que se interpolen linealmente los siguientes puntos [x,y]: [0,0], [OFF_25,25], [OFF_50,50], [OFF_75,75], [100,100], esta función vincula la apertura de las válvulas con los “voltajes” que hay que enviar a la placa driver de los servos que las comandan.

Válvula de agua caliente: procedimiento análogo al anterior

4.1.3 Programación de la Autocalibración

4.1.3.1 Introducción

Para la programación de la autocalibración de sensores en MATLAB se utilizó la herramienta StateFlow de Simulink. Esta herramienta se utiliza para construir máquinas de estados. Como el procedimiento anteriormente expuesto es secuencial y en cierto orden, es bastante intuitivo implementarlo por medio de ésta poderosa herramienta.

4.1.3.2 Implementación

Al iniciar la calibración una variable denominada CAL toma el valor real ‘1’ (en el PLC, DCS y Matlab), lo que indica que se está en proceso de calibración (valores de CAL mayores que 1 indican en qué parte de la máquina de estados se encuentra la ejecución, esto se utiliza en el DCS para habilitar las opciones y visualizar el estado actual). Enseguida después se deshabilita el encendido de la resistencia del tanque y se abre la válvula ON-OFF de salida.

1- Calibrar el 0 % y 100% de las válvulas de control

Ahora se está en condiciones de comenzar con la calibración del 0% y 100% de apertura de las válvulas.

Luego de unos segundos se comanda la válvula de agua fría a una posición intermedia entre cerrada y abierta y se incrementa de a pasos su apertura hasta que la lectura del sensor diferencial de presión (de la línea de agua fría) entre dos pasos consecutivos sea menor que cierta tolerancia. Cuando esta condición se cumple se está aproximadamente en el 100% de la apertura de la válvula y se registra la lectura del sensor como DP_FO (dif. presión fría

open) y el voltaje de salida a la válvula V_F100. Para calibrar el 0% el procedimiento es análogo, se manda a la válvula a una posición intermedia y se la cierra de a pasos hasta que la lectura del sensor de “estabilice”. En este caso se registran DP_FC (dif. presión fría closed) y V_F0 (Voltaje fría al 0% o cerrada). El procedimiento se repite para la calibración del 0% y 100% de la válvula de agua caliente.

2- Calibración del sensor de nivel

Como primer paso se cierran las válvulas de entrada de agua fría y caliente y luego se abre la válvula ON-OFF de salida. Después el operario debe esperar a que el nivel de agua en el tanque descienda hasta un punto que se tomará como referencia, en ese momento se hace click izquierdo sobre un botón en la pantalla de autocalibración en el DCS, eso provoca que se guarde la lectura del sensor de nivel en la variable LT_0 (nivel de referencia), además al mismo tiempo se manda cerrar la válvula ON-OFF de salida.

Lo siguiente es calibrar el 100% del nivel, para ello se abre la válvula de agua fría. El operario al notar que el nivel alcanzó el 100% debe realizar un click izquierdo sobre el botón correspondiente en la pantalla autocalibración, esto registra el valor (actual) del nivel en la variable LT_100 (nivel con el “tanque lleno”).

Luego de calibrado el rango de variación para el nivel se procede a vaciar el tanque. Esto implica cerrar la válvula de agua fría y abrir la válvula ON-OFF de salida. Una vez que el nivel está a cierta tolerancia del nivel de referencia LT_0 se pasa a la calibración de los caudalímetros.

3- Calibración de los caudalímetros

Caudalímetro de agua fría:

En el comienzo se cierran todas las válvulas automáticas y se registra el nivel actual en la variable Nivel_aux. Posteriormente se abre la válvula de agua fría y se esperan 30 segundos. Cuando expira el tiempo se cierra la válvula y se registra el nivel actual. Luego se calcula el delta de nivel como $\Delta LT = \text{Nivel} - \text{Nivel_aux}$ y se calcula el caudal máximo como

$$FF_{\text{Max}} = \frac{\Delta LT \cdot \text{NivelM}(m) \cdot \text{ÁreaTanque}(m^2) \cdot 1000}{0.5 \text{ min}} \text{ [Lt/min]}, \text{ donde el área del tanque se}$$

mide en m^2 y NivelM es la diferencia de nivel medido por el operario en metros (ésta variable se ingresa vía DCS).

Caudalímetro de agua caliente: el procedimiento es análogo al anterior.

Finalmente se vacía el tanque y se cierra la válvula ON-OFF de salida.

4- Calibrar válvulas de control

En este paso el objetivo es relevar varios puntos de la curva: ‘Voltaje Servos’ – ‘Caudal’ (que también se podría interpretar como: ‘Apertura válvula’ – ‘Caudal’). En primera

instancia se envía al servo que comanda la válvula de agua fría un voltaje V_F25. Este voltaje corresponde a un punto en el cual, si la válvula fuera totalmente lineal, la misma estaría abierta en un 25%. Con la válvula en ese estado se esperan 5 segundos para que se estabilice el caudal y se registra el mismo como OFF_25(valor en porcentaje).

El procedimiento se repite para un voltaje que a la “válvula ideal” la dejaría en el 50% de su apertura y finalmente otro en el cual la misma estaría abierta en un 75%.

Además ya se disponían del voltaje de 0% y 100% de apertura (obtenidos en el paso 1), con lo cual ya se tienen 5 puntos de la curva de la válvula.

Para la válvula de agua caliente el procedimiento es análogo.

Luego de terminado este punto se vuelve a habilitar la resistencia y se setean las variables de interés en los bloques que comunican dichas variables con el PLC.

4.1.3.3 Evaluación Autocalibración

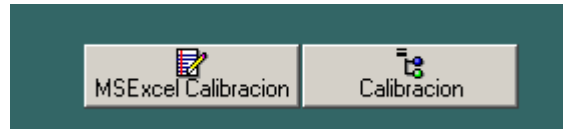
Para probar la autocalibración programada en Matlab se asoció al mismo un faceplate en el DCS (ver siguiente figura) en el cual se siguen “en línea” los pasos de la autocalibración.



Luego de realizadas las pruebas (con la planta), se llegó a la conclusión que la autocalibración funciona, pero demora más de lo previsto. El hecho de que una calibración consuma tanto tiempo y que el programa de Matlab esté siempre corriendo (aunque la autocalibración no sea habilitada) y por ende consumiendo CPU y memoria RAM fueron los principales motivos para concluir que la misma no era práctica. Por este motivo se retira y se deja un procedimiento manual.

4.2 Calibración Manual

Para calibrar la planta no es necesario ingresar al PLC para cargar las constantes de calibración, sino que todo se hace a través de la interfaz programada sobre el sistema. Cuando se deba o se desee realizar una recalibración la manera más simple es ir a la pantalla principal, y clicar en los dos botones que aparecen en la esquina inferior izquierda:



Uno de los botones abre una planilla de Excel que sirve para cargar automáticamente las constantes obtenidas de las calibraciones. Esto tiene la ventaja de que si por ejemplo se lo dejó al PLC mucho tiempo sin alimentación y se borran las constantes de su memoria, las mismas permanecen en la planilla lo que facilita la tarea de re-calibración. Además las celdas de dicha hoja se conectaron directamente con variables del sistema, de forma tal que con simplemente presionar 'F9' (en el teclado) se carguen todas las constantes en el PLC.

El otro botón abre un faceplate especialmente creado para hacer dichas calibraciones y observar el estado de la antigua calibración. En dicho faceplate se observa el estado de la memoria del PLC.

A continuación se explica el faceplate de calibración:

Calibracion : Main Faceplate

Calibracion

Calibracion

Caudal Fria		Caudal Caliente	
DP	8877	DP	648
DP_C	560	DP_C	595
DP_O	16104	DP_O	12815
CV	18200	CV	18500

ON
OFF

Nivel		Voltaje Sensores	
Nivel	4762	Fria	2.71 V
LT_0	3300	Caliente	0.20 V
LT_100	5830	Nivel	1.45 V

Histeresis Fria		Histeresis Cal		Resistencia	
Amp	150	Amp	350	DC %	10
Per ms	300	Per ms	300	Per ms	10000

Valvulas

	Y Fria	X Fria %	Y Caliente	X Cal %
XY0	18200	0	18500	0
XY1	18500	4	19000	5
XY2	18700	15	19600	20
XY3	19400	22	19800	20
XY4	19500	22	20400	21
XY5	19600	26	20600	26
XY6	19700	32	20800	36
XY7	19850	48	20900	49
XY8	20000	67	21000	64
XY9	20100	77	21150	84
XY10	20300	94	21250	93
XY11	20400	99	21350	98
XY12	20500	99	21450	100
XY13	21000	100	22000	100

En la solapa izquierda se puede observar que sobre un fondo verde hay dos botones con las indicaciones de OFF y ON. Si la indicación está en OFF el control de las válvulas de agua fría y caliente lo realiza Matlab, si se presiona el botón ON el control lo tiene el faceplate de calibración.

Notación:

Para hacer referencia a un campo del faceplate se adopta la siguiente notación:

‘Título→Campo’

donde ‘Título’ es el título de la sección en el faceplate y ‘Campo’ es el campo al que se hace referencia.

A la hora de calibrar se ingresan en: Caudal Caliente→CV y Caudal Fría→CV los valores de la salida real del PLC, que están en el rango (aproximado) de 18000 a 22000 cuentas, para abrir o cerrar dichas válvulas (se aconseja utilizar el rango de la tabla, que se encuentra en el faceplate, a la derecha).

4.2.1 Caudalímetros

Para calibrar los caudalímetros se deben seguir los siguientes pasos:

- 0.-En el faceplate anterior setear el botón de calibración en ON.
- 1.-Abrir completamente las válvulas de control de agua fría y caliente (a través del faceplate, utilizar como guía la tabla a la derecha del mismo). Para eso se ingresan en ‘Caudal Caliente→CV’ y ‘Caudal Fría→CV’ unas 23000 cuentas (se ingresan 1000 cuentas más de lo indicado en la tabla para tener seguridad de que las válvulas estén totalmente abiertas, aunque se pueden ingresar hasta aproximadamente 24000 cuentas).
- 2.-Abrir y luego cerrar las válvulas manuales hasta que ‘Voltaje Sensores→Fría’ y ‘Voltaje Sensores→Caliente’ casi saturen (los voltajes de saturación son para el sensor diferencial de presión de agua fría: 4,92 V y para el sensor diferencial de presión de agua caliente: 3,82V)
- 3.-En el momento en que saturan los sensores se debe ingresar en ‘Caudal Fría→DP_O’ el valor de ‘Caudal Fría→DP’ y en ‘Caudal Caliente→DP_O’ el valor de ‘Caudal Caliente→DP’
- 4.-Cerrar las válvulas de control de agua fría y caliente. Para eso se ingresan en ‘Caudal Caliente→CV’ y ‘Caudal Fría→CV’ unas 18000 cuentas (ver tabla Válvulas).
- 5.-Ingresar en ‘Caudal Fría→DP_C’ el valor de ‘Caudal Fría→DP’ y en ‘Caudal Caliente→DP_C’ el valor de ‘Caudal Caliente→DP’.
- 6.-Setear el botón de calibración en OFF.

Se recomienda escribir los valores en cuestión, en la hoja de Excel y presionar ‘F9’ para que se actualicen en el DCS y por consiguiente en el PLC.

4.2.2 Sensor de nivel

En el caso del nivel se deben ingresar los niveles LT_0 y LT_100 utilizados para desplegar el porcentaje de nivel en pantalla y para configurar los límites de acción de las alarmas de nivel bajo y alto respectivamente. Para ello se deben seguir estos pasos:

- 0.-Setear el botón de calibración en ON.
- 1.-Cerrar las válvulas de control de agua fría y caliente
- 2.-Desagotar el tanque
- 3.-Cerrar la válvula de salida del tanque completamente (también mantener en posición normal la válvula ON-OFF, estado OFF)
- 4.-Abrir la válvula de agua fría un porcentaje no demasiado alto, se aconseja 50%
- 5.-Inmediatamente después que la resistencia se haya cubierto totalmente de agua cerrar la válvula de agua fría.
- 6.-Ingresar en 'Nivel→LT_0' el valor de 'Nivel→Nivel'
- 7.-Abrir nuevamente el agua fría.
- 8.-Cerrar la válvula de agua fría una vez tanque este casi lleno (sin desbordar)
- 9.-Ingresar en 'Nivel→LT_100' el valor de 'Nivel→Nivel'
- 10.-Setear el botón de calibración en OFF.

Se recomienda escribir los valores en cuestión, en la hoja de Excel y presionar 'F9' para que se actualicen en el DCS y por consiguiente en el PLC.

Esta calibración es de suma importancia ya que se programaron los siguientes bloqueos en el PLC:

- 1.-Por bajo nivel: deshabilita la función de la resistencia para cuando no está totalmente cubierta de agua.
- 2.-Por alto nivel: cierra las válvulas de control de agua fría y caliente cuando se supera el 100% del tanque evitando desbordes de este.

4.2.3 Acción de la resistencia

Para configurar el ciclo de trabajo de la resistencia se debe ingresar en 'Resistencia→DC %' el valor del ciclo en porcentaje y en 'Resistencia→Per ms' el período de trabajo en milisegundos.

4.2.4 Calibración de las válvulas

- 0.-Setear el botón de calibración en ON.
- 1.-Ingresar en 'Histéresis Cal→Amp': 350 (cuentas) y en 'Histéresis Cal→Per ms': 30
- 2.-Ingresar en 'Histéresis Fría→Amp': 150 (cuentas) y en 'Histéresis Fría→Per ms': 30
Los pasos 1 y 2 ya fueron relevados, en caso de NO confiar de estos valores hay que volver realizar ensayos y volver a calcularlos.
- 3.-Abrir la válvula de salida del tanque
- 4.-Presionar el botón de ON en el faceplate de calibración

- 5.-Cerrar la válvula de agua caliente ingresando el valor 18500 o el que corresponda en ‘Caudal Caliente→CV’
- 6.-Cerrar la Válvula de agua fría ingresando el valor 18400 o el que corresponda en ‘Caudal Fría→CV’
- 7.-Leer el valor de ‘Voltaje Sensores→V_Fría’ e ingresar al Excel de calibración
- 8.-Ingresar en el Excel el valor de ‘Caudal Fría→CV’
- 9.-Incrementar el valor de ‘Caudal Fría→CV’ en aproximadamente: paso=2500/14. Tener en cuenta que la válvula cierra aproximadamente en 21000 y hay que ingresar 14 datos en el Excel que luego serán cargados al DCS y este posteriormente los cargará al PLC en la función FKG. Es muy aconsejable, que el tamaño del paso sea variable, así se obtiene mejor resolución en la zona crítica.
- 10.-Ingresar en la celda correspondiente en el Excel el valor de ‘Voltaje Sensores→Fría’
- 11.-Cerrar la válvula y luego incrementar el valor de ‘Caudal Fría→CV’, en n*paso (dónde ‘n’ es el punto actual de relevamiento)
- 12.-Ingresar en la celda correspondiente en el Excel el valor de ‘Voltaje Sensores→Fría’
- 13.-Repetir los pasos 11 y 12 hasta que se complete la tabla, tener en cuenta que en el Excel hay que llenar una columna con datos de subida y otra con datos de bajada. En el caso de los datos de bajada siempre se pasa de un estado con la válvula 100% abierta a otro con la válvula abierta según el punto que se esté relevando en la curva.
- 14.-El procedimiento para la válvula caliente es análogo al del agua fría.
- 15.-Una vez terminado el procedimiento presionar F9 (con la planilla del Excel seleccionada) para que carguen los datos en el DCS y PLC
- 16.-Presionar el botón ON en el faceplate de calibración para que Matlab tome el control de las válvulas.
- 17.-Setear el botón de calibración en ON.

Un ejemplo de calibración se puede ver en la siguiente tabla de Excel:

	FKG Fria							FKG Caliente						Nivel	
	Ys (V)	Xs(DP)	Xb(DP)	Xp(DP)	X%	Y%		Ys (V)	Xs(DP)	Xb(DP)	Xp(DP)	X%	Y%		
0	18200	0,17	0,17	0,17	0	0	0	18500	0,20	0,23	0,22	0	0	LT 0	3300
1	18500	0,17	0,19	0,18	5	11	1	19000	0,21	0,24	0,23	5	14	LT 100	5830
2	18700	0,26	0,29	0,29	16	18	2	19600	0,29	0,45	0,37	20	31	Caudal Caliente	
3	19400	0,41	0,42	0,42	23	43	3	19800	0,32	0,42	0,37	20	37	DP_CC	595
4	19500	0,41	0,42	0,42	23	46	4	20400	0,34	0,43	0,39	21	54	DP_CO	12815
5	19600	0,50	0,51	0,51	27	50	5	20600	0,34	0,63	0,49	27	60	Caudal Fria	
6	19700	0,63	0,70	0,67	32	54	6	20800	0,55	0,87	0,71	37	66	DP_FC	560
7	19850	1,20	1,40	1,30	49	59	7	20900	0,70	1,51	1,11	49	69	DP_FO	16104
8	20000	2,25	2,45	2,35	68	64	8	21000	1,14	2,37	1,76	64	71		
9	20100	2,95	3,05	3,00	77	68	9	21150	2,22	3,44	2,83	84	76		
10	20300	4,40	4,43	4,42	95	75	10	21250	3,10	3,75	3,43	93	79		
11	20400	4,88	4,89	4,89	100	79	11	21350	3,72	3,92	3,82	99	81		
12	20500	4,90	4,91	4,91	100	82	12	21450	3,92	3,92	3,92	100	84		
13	21000	4,91	4,91	4,91	100	100	13	22000	3,92	3,92	3,92	100	100		

5 Controladores PID

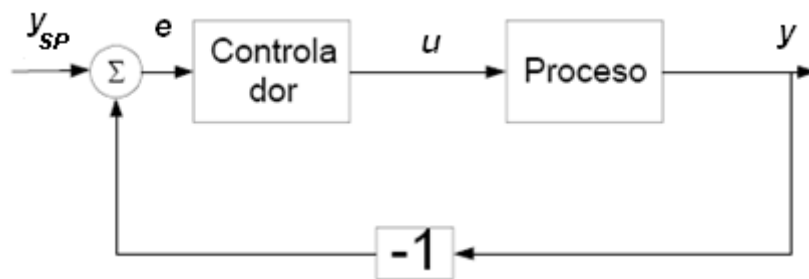
En éste capítulo se centra el análisis en los controladores PID. A modo de introducción se explica su funcionalidad y se describen varias representaciones alternativas. En particular se describe el PID con ‘Ponderación de la Referencia’ que tiene un buen desempeño cuando hay altos niveles de ruido en las señales. Luego se estudian las debilidades del PID teórico y se detallan maneras de mejorar su desempeño en la práctica, en particular se tratan los efectos de Windup Integral y la técnica anti-windup denominada: ‘Back Calculation’. Además se adiciona el requerimiento extra de que sea un controlador Bumpless.

Otro punto de particular interés es el estudio de técnicas de autosintonía de dichos controladores y la implementación de la técnica de autosintonía denominada: ‘Método del Relé’ de Aström.

Finalmente se detallan las pruebas realizada para validar la implementación del controlador diseñado.

5.1 Introducción

Un esquema común de un sistema de control se muestra en la siguiente figura:



El sistema está caracterizado por la formación de un error, que es la diferencia entre la referencia y la salida del proceso ($y_{sp} - y$). El controlador genera una señal de control (u) que se obtiene mediante el procesamiento del error y es aplicada luego al proceso. Este sistema es llamado “sistema con error realimentado” debido a que el controlador opera sobre la señal de error. Una estructura más flexible se obtiene mediante el tratamiento de la referencia y de la salida del proceso de manera separada.

5.2 Control PID

Un PID (Proporcional Integral Derivativo) es un mecanismo de control por realimentación que se utiliza en sistemas de control industriales. Un controlador PID corrige el error, entre un valor medido y el valor que se quiere obtener, calculándolo y luego sacando una acción correctora que puede ajustar el proceso acorde. El algoritmo de cálculo del control PID se da en tres parámetros distintos: el proporcional, el integral, y el derivativo. El valor Proporcional determina la reacción del error actual. El Integral determina la reacción basada en la suma de errores recientes y el Derivativo determina la reacción del tiempo en el que el error se produce. La suma de estas tres acciones es usada para ajustar al proceso vía un elemento de control como por ejemplo la posición de una

válvula de control o la energía suministrada a un calentador. Ajustando estas tres constantes en el algoritmo de control del PID, el controlador puede proveer un control diseñado para lo que requiera el proceso a realizar. La respuesta del controlador puede ser descrita en términos de respuesta del control ante un error, el grado el cual el controlador llega al "set point", y el grado de oscilación del sistema. Nótese que el uso del PID para control no garantiza control óptimo del sistema o la estabilidad del mismo.

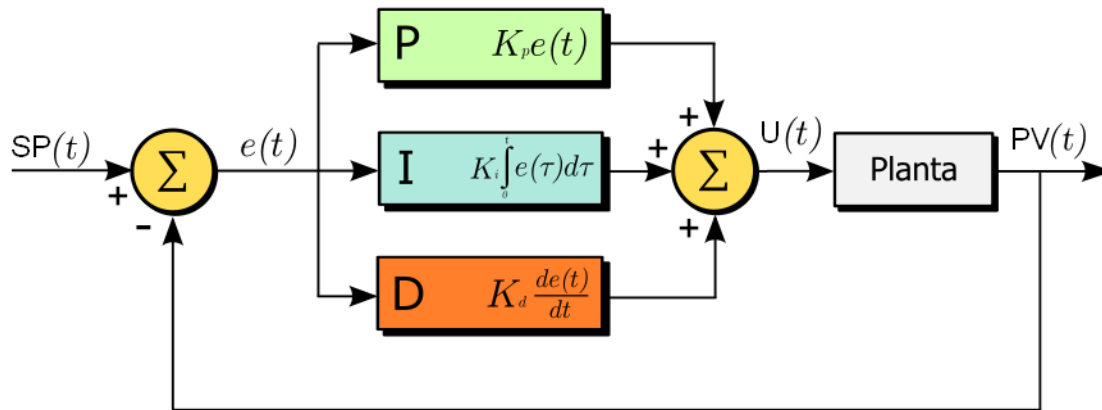


Diagrama en bloques de un control PID.

¹² “El control proporcional tiene la desventaja de que, en la mayoría de los casos, resulta en un error estático o de estado estacionario diferente de cero. Los algoritmos de control usados en la práctica son, por tanto, normalmente más complejos que el del controlador proporcional. Se puede mostrar empíricamente que el llamado ‘controlador PID’ es una estructura útil. Dentro de la banda proporcional el comportamiento del algoritmo PID en su versión de “libro de texto” se puede describir como:

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

Donde ‘ u ’ es la variable de control (la acción del PID sobre algún actuador) y ‘ e ’ es el error de control dado por $e = SP - PV$ (siendo PV el valor actual de la variable de proceso y SP es el valor deseado para la variable en cuestión). De esta manera, la variable de control es una suma de tres términos: el término P, que es proporcional al error; el término I, que es proporcional a la integral del error; y el término D, que es proporcional a la derivada del error. Los parámetros del controlador son: la ganancia proporcional K , el tiempo integral T_i y el tiempo derivativo T_d .”

¹² (Åström, 2002)

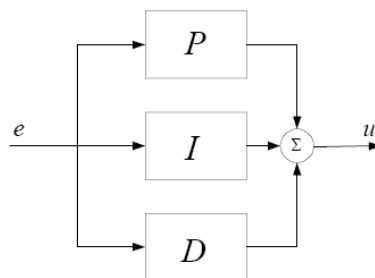
5.3 Representaciones Alternativas

Este punto está basado en el texto “*Apuntes de Control PID*” (Capítulo 3), publicado en 2001, de M. Moreno.

5.3.1 Controlador PID Paralelo

El algoritmo dado en la ecuación (A) puede ser representado por la siguiente función de transferencia y el siguiente diagrama de bloques:

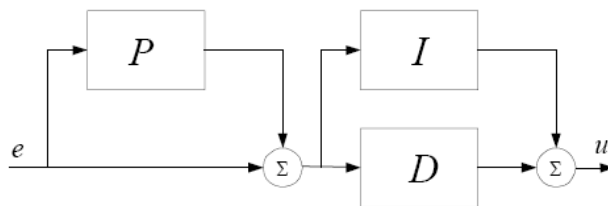
$$G(s) = K \left(1 + \frac{1}{sT_i} + sT_d \right) \quad (B)$$



5.3.2 Controlador PID Serie

Una versión ligeramente diferente es más común en muchos controladores comerciales. Este controlador está descrito por:

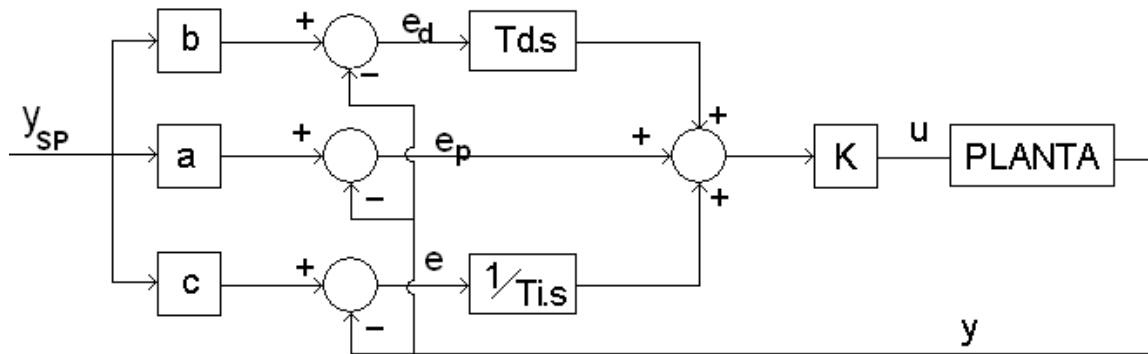
$$G(s) = K' \left(1 + \frac{1}{sT'_i} \right) (1 + sT'_d) \quad (C)$$



El controlador dado por la ecuación (B) se llama “no interactivo” y el dado por la ecuación (C) “interactivo”. La razón para esta nomenclatura es que en el controlador (B) el tiempo integral T'_i no influye en la parte derivativa, y el tiempo derivativo T'_d no influye en la parte integral, de esta forma, las partes no interactúan entre sí. En el controlador interactivo, el tiempo derivativo T'_d influye en la parte integral. Por tanto, las partes son interactivas.

5.3.3 PID con Ponderación de la Referencia

En esta variación del PID clásico el esquema del controlador es el siguiente:



donde el error en la parte proporcional es:

$$e_p = by_{sp} - y$$

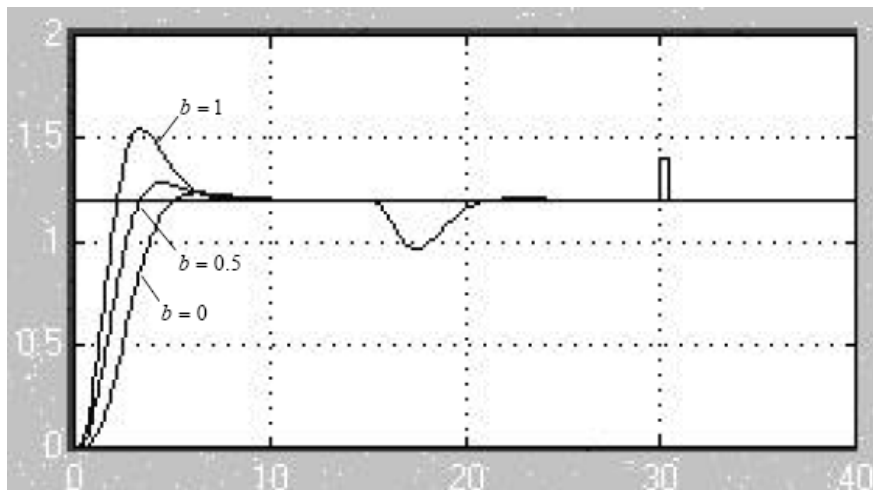
y el error en la parte derivativa es:

$$e_d = cy_{sp} - y$$

Para evitar errores de control en estado estacionario, el error en la parte integral debe ser el verdadero error de control:

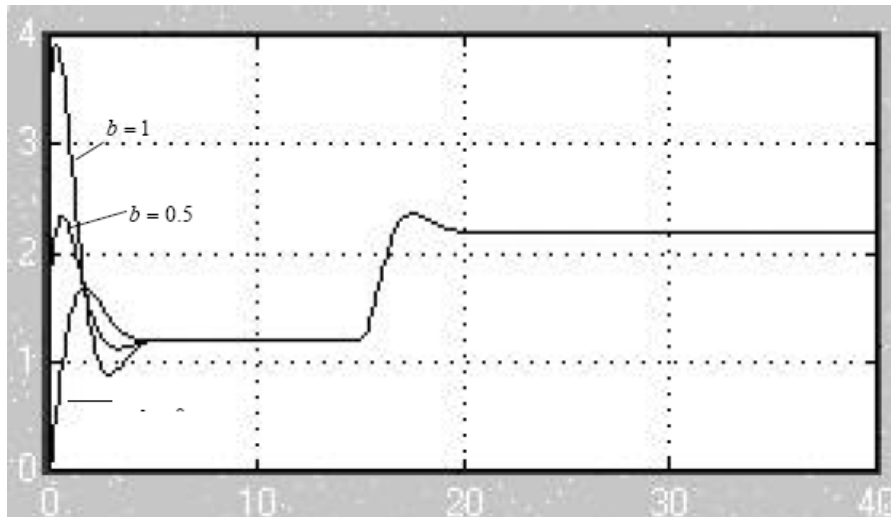
$$e = y_{sp} - y$$

Los controladores obtenidos para diferentes valores de 'b' y 'c' responden a perturbaciones de carga y ruido de medición de manera similar. La respuesta a los cambios en la referencia dependerá, sin embargo, de dichos valores. Esto se ilustra en las siguientes figuras, que muestra la respuesta de un controlador PID con respecto a cambios en la referencia, perturbaciones de carga y errores de medición, para diferentes valores de 'b'.



¹³ Respuesta de un controlador PID con respecto a cambios en la referencia para diferentes valores de 'b'

¹³ (Moreno, 2001)



¹⁴ Respuesta con respecto a cambios en la referencia, perturbaciones de carga y errores de medición para diferentes valores de la ponderación de la referencia 'b'

Las figuras muestran claramente el efecto de cambiar 'b'. El sobrepaso con respecto a los cambios en la referencia es más pequeño para $b = 0$, que es el caso donde la referencia sólo es introducida en el término integral y se incrementa cuando se incrementa 'b'. Una simulación como la de las figuras anteriores es muy útil para hacer una rápida evaluación de las respuestas del sistema en lazo cerrado con respecto a los cambios en la referencia, las perturbaciones de carga y los errores de medición.

El parámetro 'c' normalmente se elige igual a cero para evitar grandes transitorios en la señal de control debido a cambios repentinos en la referencia. Una excepción es cuando el controlador es el controlador secundario en un acoplamiento en cascada. En este caso, la referencia cambia suavemente, debido a que es dada por la salida del controlador primario.

El controlador con $b = 0$ y $c = 0$ se llama algunas veces controlador I-PD y el controlador con $b = 1$ y $c = 0$ se llama algunas veces controlador PI-D. En general, se prefiere el genérico uso de PID y dar los parámetros de 'b' y 'c'.

¹⁴ (Moreno, 2001)

5.4 Windup

5.4.1 Introducción

Gran parte de este punto está basado en una traducción libre del texto de K.J. Astöm y T. Hägglund: “*PID Controllers: Theory, Design, and Tuning*” (Capítulo 3.5), publicado en 1995,

¹⁵“A finales de los años 50 y principios de los años 60, cuando se empezaban a establecer poco a poco los sistemas de control, se empezaron a observar comportamientos anómalos y no lineales que debían considerarse. Estos comportamientos procedían de saturaciones de los actuadores (las válvulas no podían abrirse o cerrarse más del 100%, los motores no podían girar a más revoluciones de sus nominales, etc.), modos de conmutación de manual a automático, cambios de parámetros.”

“Por ello, y dado que todos los actuadores tienen limitaciones físicas, se producían una serie de consecuencias para el control, que no afectaban cuando el lazo de control estaba cerrado. Pero dicho lazo de control se rompía cuando el actuador se saturaba, debido a que la salida del elemento saturado ya no quedaba controlada por su entrada. Este modo inestable, provocaba valores de parámetros muy grandes en el controlador, aportaciones negativas referentes al slew-rate. Sin embargo, cuando el actuador se desaturaba, se sucedían un conjunto de transitorios durante un cierto tiempo que incluso producían que el actuador oscilara entre un nivel alto y un nivel bajo antes de que el sistema se recuperase.”

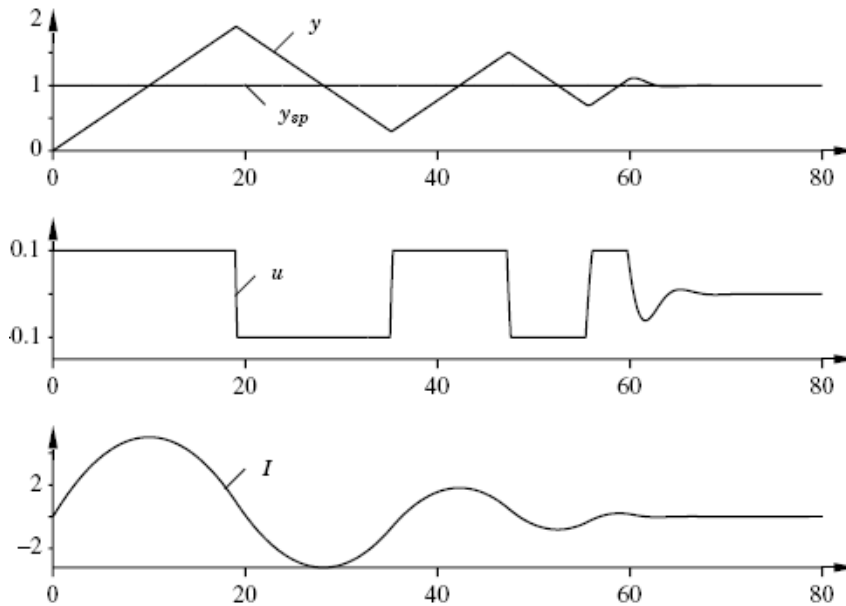
¹⁶“El fenómeno windup puede verse, por tanto, en la siguiente figura, que muestra un sistema de control con un controlador de tipo PI. El cambio inicial de la consigna (SP) es tan grande, que el actuador satura en el nivel alto. El término integral aumenta inicialmente porque el error es positivo y alcanza su valor más grande en un tiempo $t=10\text{seg}$. Cuando el error va hacia cero, la salida queda saturada en ese punto a causa del gran valor del término Integral, que no deja el límite de saturación hasta que el error ha sido negativo durante el suficiente tiempo que hace que la parte Integral caiga a un nivel bajo.”

“Es importante destacar que la señal de control oscila entre sus límites varias veces. El efecto que esto tiene sobre la red es una gran sobre oscilación donde la señal de control se dirige desde un extremo a otro. La salida finalmente se aproxima al valor de la consigna y el actuador no satura. El sistema se comporta al cabo de un tiempo como lineal y estable.”

“En este ejemplo se muestra el windup del integrador que es generado por un cambio en el valor de referencia. El fenómeno del windup, también puede ser debido a grandes perturbaciones o a un mal funcionamiento del equipo.”

¹⁵ (J.M. espinosa, 2006)

¹⁶ (Astrom, 2002)
(Levine, 1996)



17 Ilustración del windup del integrador. El diagrama muestra la señal de salida del proceso y , la consigna y_{sp} , la señal de control u , y la parte Integral I .

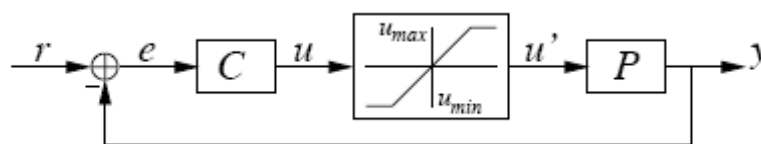
Este fenómeno, se observó mayoritariamente en los reguladores de tipo PID dado que éstos resolvían desde aquel entonces hasta nuestros días el 90% de los problemas de control.

Con todo ello, debido a la saturación en el actuador, la señal de control satura inmediatamente cuando se le aplica la entrada escalón unidad. La señal de control entonces entra en saturación y se rompe la realimentación.

Con la aparición del fenómeno de windup, la parte Integral del regulador PID aumenta, causado por el error positivo. Dicha parte integral, empieza a decrementarse sólo cuando la salida es igual al valor de consigna, pero la salida permanece saturada a causa del gran valor de dicha parte Integral.

La salida finalmente decrece cuando la parte Integral desciende lo suficiente. La consecuencia inmediata de esto, como se ha comentado anteriormente, es una situación de gran oscilación y que el sistema tarda mucho en recuperarse. Cuando ocurre el fenómeno de windup en el integrador de un controlador PID se lo denomina como “integrator windup”.

Desde la época en que se descubrió dicho fenómeno con la fuerte industrialización que existía hasta nuestros días (años 50, 60, 70, 80 y 90...) se han desarrollado muchos “esquemas antiwindup” para evitar que esto se produzca.



Esquema de control con saturación

17 (Åström, 2002)

5.4.2 Técnicas Anti-Windup

5.4.2.1 Limitación de la referencia

¹⁸La forma más intuitiva de evitar el efecto windup del integrador es evitando la saturación de la variable de control. Esto puede realizarse limitando o suavizando los cambios de set-point en el controlador. Este proceso requiere un esfuerzo adicional en el diseño del controlador y por sobretodo puede implicar un decremento de la performance inaceptable. Por lo tanto este enfoque no es recomendable en casos prácticos.

5.4.2.2 Integración condicional

Una metodología clásica y efectiva es la llamada integración condicional. Consiste en apagar el integrador (en otras palabras, el error a integrar es puesto a cero) cuando cierta condición es verificada. Las siguientes opciones pueden ser implementadas:

- limitar el termino integral a un valor predefinido
- detener la integración cuando el error es mayor a una banda predefinida, o sea cuando el error está lejos del valor de set-point
- detener la integración cuando la variable de control satura
- detener la integración cuando la variable de control satura y el error y la variable de control tienen el mismo signo

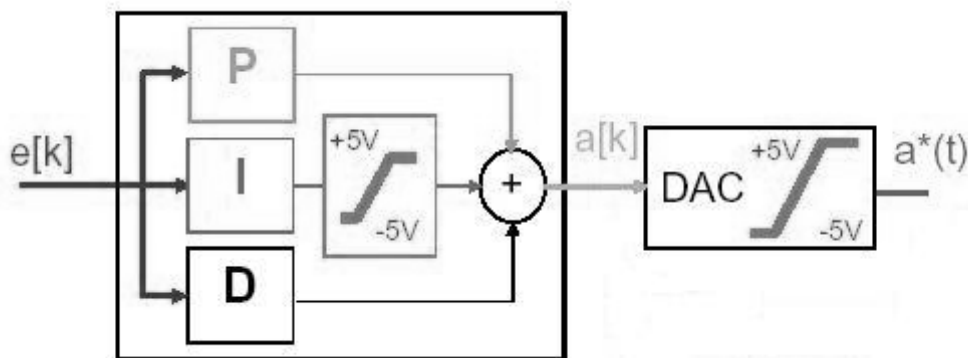
Los dos primeros métodos, parecen ser particularmente adecuados para el inicio de algoritmos, tienen la desventaja que pueden producir un resultado con error en tiempo estacionario. En el primer caso la limitación del término integral puede no prevenir el alcance del valor de set-point y en el segundo caso debe ser evitado que el controlador se atasque en un valor de tiempo estacionario de manera que el error de control sea mayor que la banda predefinida.

Parece que ambos métodos requieren un parámetro de diseño adicional que debe ser cuidadosamente seleccionado, para poder manejar apropiadamente el intercambio entre la necesidad de evitar el windup y asegurar un error en tiempo estacionario cero.

¹⁸ (Åström, 1995)

5.4.2.3 ¹⁹Limitación estática del término Integral

Saturar el término Integral, se utilizó entre los años 1950-60.



Limitación estática del término Integral

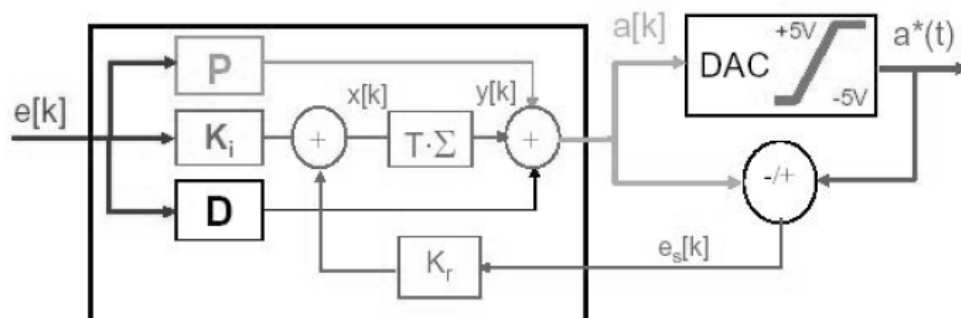
“Con este esquema se tiene un sistema con controlador tipo PID, invariante y no lineal. Además, puede llegar en ciertos momentos a superar la saturación física del DAC. Posee unos límites fijos para un actuador. Y como inconveniente, se observa que se permite que la acción quede fuera del rango del actuador. Este esquema es fácil de implementar, a costa de cargar al controlador con discontinuidades.”

5.4.2.4 ²⁰Seguimiento Integral

“En este caso la ganancia Integral es dependiente del error (Años 1950-60). Esta configuración, que consta de un controlador PID, variante y lineal, es de las más eficientes dado que añade un lazo de realimentación dentro del propio PID. Está considerada como una de las soluciones más elegantes ya que cuando existe saturación, se modifica el valor de I y cuando no existe saturación, el lazo no actúa.”

“Es de destacar el valor y significado de la constante K_r que indica: con qué rapidez se hace la corrección.”

“La forma más habitual de calcular su valor es: $K_r = 1 / [\sqrt{(T_i T_d)}]$ ó $[(T_i + T_d)]$
Como otra característica de esta configuración es que, si la salida del DAC no se puede medir se estima a través de un modelo.”



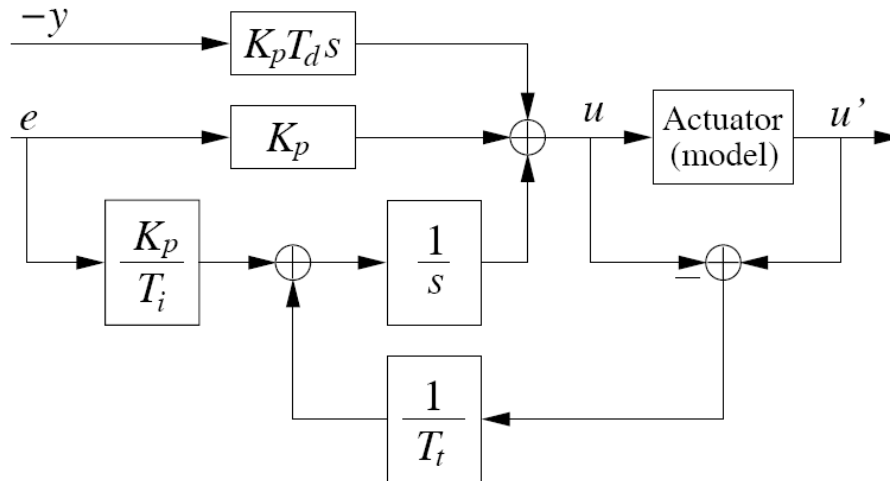
Seguimiento Integral

¹⁹ (Espinosa, 2006)

²⁰ (Espinosa, 2006)

5.4.2.5 ²¹ Back-Calculation

Una valiosa alternativa de aproximación a la integración condicional es la llamada Back-Calculation. La misma consiste en recalculer el término integral cuando el controlador satura. Para explicar ésta técnica y su implementación en este proyecto se utiliza un controlador PID con Ponderación de la Referencia (ver 5.3.3) eligiendo como parámetros 'a'=1, 'c'=1 y 'b'=0. Con ésta elección de parámetros el esquema del PID Anti-Windup es el siguiente:



Esquema del PID Anti-Windup

Básicamente, el valor de la integral es reducido, cuando la salida del controlador es mayor que su límite superior u_{\max} , o incrementado, cuando es menor que su límite inferior u_{\min} . Para ello se realimenta la diferencia de la señal de control saturada y no saturada, como se muestra en la figura anterior, donde T_t es llamada la constante de tiempo de seguimiento. Formalmente, denotando e_i a la entrada del integrador, queda:

$$e_i = \frac{K_p}{T_i} e + \frac{1}{T_t} (u' - u)$$

Debe ser notado que esta técnica tiene una propiedad de observador inherente. En particular, apunta a estimar el correcto estado del controlador cuando no corresponde a la entrada del proceso (debido a la saturación del actuador). El valor de T_t claramente determina la tasa en la cual el término integral es reseteado y luego esa decisión determina la performance del esquema de control global. Para ayudar al operador en este contexto, han sido propuestas reglas de afinación para la constante de seguimiento. En las mismas es sugerido que:

$$T_t = \sqrt{T_i T_d}$$

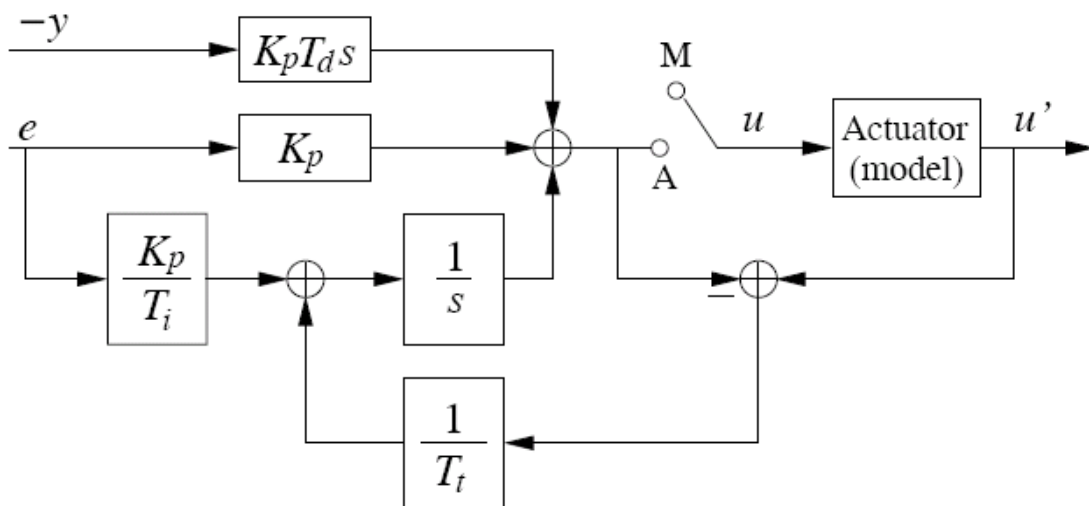
²¹ (Antonio Visioli, 2006)
(Åström, 1995)

La mayor desventaja de esta fórmula es que no puede ser adoptada en un controlador PI donde $T_d = 0$. En esos casos es sugerido que: $T_t = T_i$.

En este punto debe ser mencionado que esta técnica incluye la llamada técnica condicional, la cual representa un método de transferencia general anti-windup y bumpless. Ciertamente, la conmutación entre el modo manual y automático y el efecto windup del integrador pueden ser vistos en el contexto general de poder manejar una situación en la cual la actual entrada del proceso sea diferente de la salida del controlador. La técnica de condicionamiento asegura que cuando la diferencia entre la variable de control y la deseada desaparece, entonces la salida del proceso alcanza su valor en estado estacionario con la misma dinámica que el sistema en lazo cerrado sin restricciones. En este punto la sugerencia es asignar:

$$T_t = K_p$$

La correlación entre la transferencia anti-windup y bumpless en el pasaje de manual a automático se muestra en la siguiente figura:



²²Esquema de control Anti-Windup y Bumpless para transición entre Manual (M) y Automático (A)

El controlador puede ser interpretado teniendo un modo de control, cuando este opera como un controlador ordinario, y un modo de seguimiento, cuando el integrador sigue una señal específica (se debe de notar que el modo de seguimiento es automáticamente desconectado cuando la señal a seguir es la salida del controlador).

²² (Antonio Visioli, 2006)
(Åström, 1995)

5.5 Autosintonía del PID por el método del relé

5.5.1 Objetivos

- Comprender las consideraciones prácticas que se han de tener en cuenta para la aplicación del método.
- Analizar algunas técnicas de sintonía para el cálculo de los parámetros del PID a partir de los valores obtenidos en la experimentación del relé.

5.5.2 ²³ Introducción

La sintonía de un controlador consiste en dar los valores adecuados a sus parámetros. Así, en el caso del regulador PID se deberán determinar K_p , T_i y T_d puesto que la acción de control que se obtiene es la combinación lineal de las acciones proporcional, integral y derivativa:

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

Antes de continuar con la sintonía de un controlador, se hará una pequeña reseña histórica de la aparición de estos controladores y su sintonización primitiva.

En 1940 Taylor Instrument introdujo al mercado el primer controlador PID, el Fulscope modelo 100. Por aquel entonces no existía un procedimiento sistemático para la sintonización de los controladores industriales, esto significa que los controladores P, PI o PD se sintonizaban, básicamente, mediante procedimientos de prueba y error.

John Ziegler y Nathaniel Nichols, ingenieros de Taylor, se abocaron entonces a la tarea de realizar pruebas experimentales con varios procesos para determinar una forma simple de sintonizar el nuevo controlador. El resultado de ese trabajo, fue el método de sintonización ampliamente conocido como método de Ziegler- Nichols (Ziegler y Nichols, 1942), el cual despertó reacciones encontradas al momento de su presentación, en la reunión anual de la Sociedad Americana de Ingenieros Mecánicos (ASME) en 1941. A pesar de que han transcurrido ya más de sesenta años desde su desarrollo, el método de Ziegler y Nichols se sigue empleando, ya sea en su versión original o en alguna de sus variantes.

Para sintonizar un controlador se debe conocer el comportamiento de la planta implicada. Por lo tanto, el proceso completo consta de tres fases:

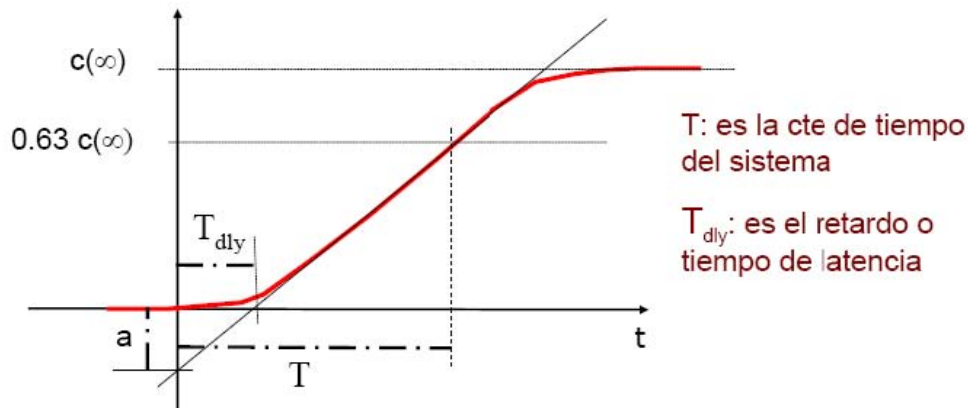
- 1 Excitación del sistema con una entrada de prueba.
- 2 Observación de la respuesta obtenida para determinar la dinámica del sistema.
- 3 Cálculo de los parámetros del controlador basándose en algún método de sintonía. Basándose en los pasos básicos de sintonía, con las técnicas de identificación de la dinámica del sistema (paso 2) se pretende la obtención de unos parámetros primarios que permitan la posterior aplicación del paso 3.

²³ (Alfaro Ruiz, 2005)

5.5.3 Métodos para la obtención de los parámetros primarios

5.5.3.1 Métodos basados en la respuesta escalón

Estos métodos se llevan a cabo en lazo abierto, por lo que su principal desventaja es que son sensibles a las perturbaciones. Además, se debe cumplir la condición de que la planta tenga respuesta sobreamortiguada.



Como se refleja en la figura los parámetros primarios son 'a' y ' T_{dly} ', a partir de los cuales se obtendrán las constantes del controlador PID (5.5.2 punto 3). Este paso puede llevarse a cabo a través de las tablas desarrolladas por Ziegler y Nichols para lazo abierto, o también, a través de los parámetros de Cohen y Coon (1953, recalculados por Witt y Waggoner en 1999) que representan una mejora con respecto a los de los primeros por ser menos sensibles a las variaciones de los datos de la curva.

5.5.3.2 Métodos de oscilación mantenida con un controlador proporcional

Estos métodos llevan al sistema hasta el límite de la inestabilidad a través del aumento de la ganancia proporcional, por lo que las acciones D e I quedan desactivadas en un PID. Se trata de la experimentación en lazo cerrado propuesta por Ziegler y Nichols en el año 1942. De la prueba se obtienen la ganancia crítica (o última), K_c , a la que se produce la oscilación, y el período que ésta posee, T_c . Con ellos, se aplican posteriormente métodos de sintonía para la determinación de los valores del PID.

5.5.3.3 ²⁴Métodos de realimentación con relé

Este procedimiento fue desarrollado por Aström y Hägglund en el año 1984, y está basado en el procedimiento anterior en lazo cerrado de Ziegler-Nichols.



Karl Johan Aström

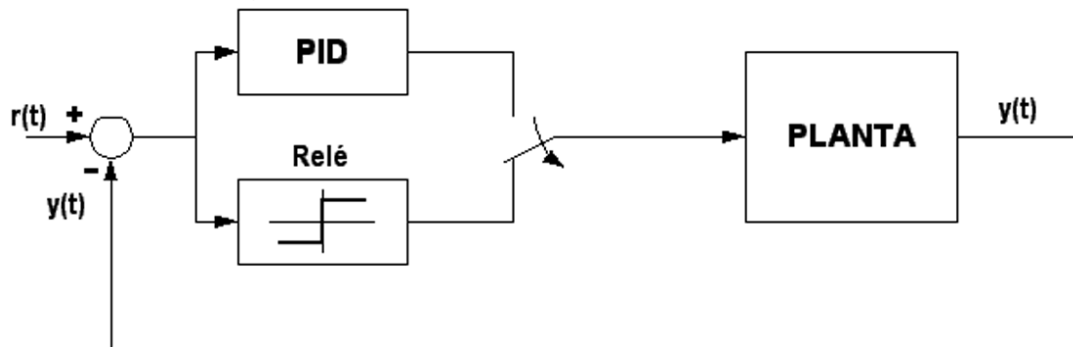


Tore Hägglund

Igualmente al método de Ziegler y Nichols, se obtienen K_c y T_c de la experimentación, siendo ésta última la principal novedad del método, ya que se introduce

²⁴ (Alfaro Ruiz, 2001)

un relé en lugar de la ganancia proporcional. Éste es el método a implementar, por lo que se presenta sólo su esquema:



Una vez estimados los parámetros primarios se deben aplicar métodos de sintonía para la determinación del controlador PID (5.5.2 paso 3). Como ejemplo se pueden destacar los métodos de Ziegler-Nichols en lazo abierto o en lazo cerrado, basados en la relación de decadencia de $\frac{1}{4}$, aunque también se han desarrollado variantes de los mismos para afinar el resultado. Además, existen otros métodos diferentes de los de Z-N.

La autosintonía de controladores por el método del relé tiene una gran aceptación industrial hoy en día ya que es un procedimiento que precisa poca información inicial del proceso ya que es el propio controlador el que lleva a cabo el desarrollo. Además, es un método que se caracteriza por su robustez porque en la experimentación en lazo cerrado se tienen en cuenta las perturbaciones en tiempo real para la obtención de los parámetros del controlador.

5.5.4 ²⁵Estudio teórico del método

5.5.4.1 Ventajas de la autosintonía por el método del relé

Con la propuesta de Aström y Hägglund se producen importantes mejoras con respecto a los métodos de Ziegler y Nichols:

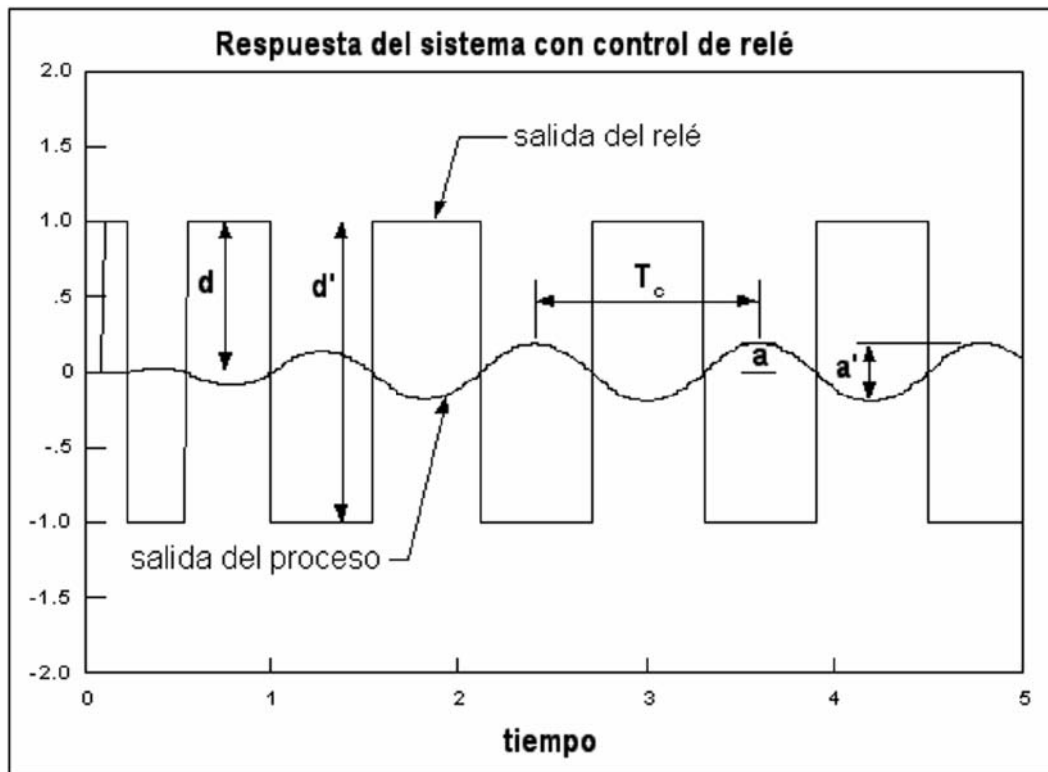
1. De sintonía manual se pasa a un proceso automático, con lo que el trabajo más arduo deja de ser realizado por un operador, que es sustituido por el controlador. Éste es uno de los principios que fomentó el desarrollo de las teorías de control y aquí es perfectamente aplicable.
2. El tiempo de sintonía es menor con los nuevos métodos, ya que con la inclusión del relé, la salida alcanza de forma más rápida la oscilación permanente.
3. En Z-N, existe una mayor dificultad de operar con el proceso en la frontera de estabilidad, ya que la oscilación puede crecer sin límites si se aumenta ligeramente la ganancia por encima de la crítica, K_c . Por el método Aström, la oscilación del relé y , en consecuencia, de la señal de salida, tiene una amplitud que puede ser controlada y mantenida con facilidad.

Así, se ejerce un mayor cuidado de la integridad de la planta ya que es un procedimiento más rápido con lo que el proceso está el menor tiempo posible en el límite de la estabilidad, y con una oscilación mantenida en el tiempo que dura el ensayo.

²⁵ (Alfaro Ruiz, 2001)

5.5.4.2 Comportamiento del relé

En primer lugar, la planta se encontraría en serie con el relé (fase de sintonía) y posteriormente, el PID tomaría los parámetros calculados y pasaría a ser él el elemento anterior a la planta, correspondiente a la fase de ejecución o control del proceso. En la siguiente figura se muestra el comportamiento del relé y de la salida de la planta al aplicar este método de autosintonía:



El comportamiento de un relé (controlador todo/nada) es:

$$y(t) > r(t) \Rightarrow OFF \Rightarrow u(t) = -d$$

$$y(t) < r(t) \Rightarrow ON \Rightarrow u(t) = +d$$

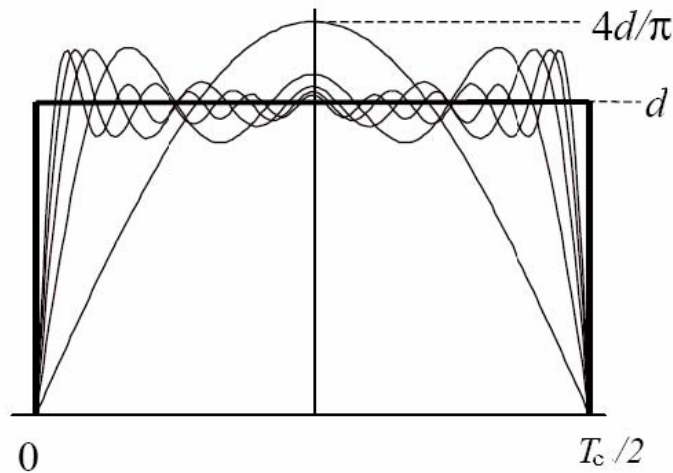
En el caso representado en la gráfica se puede deducir que la consigna es 0 ya que es el valor en torno al que oscila la salida. Por tanto, en los instantes en los que la salida es menor a la referencia (como en $t = 2$), el relé toma el valor $+d$ (en este ejemplo $d = 1$). Por el contrario, en los puntos donde $y(t)$ es mayor que $r(t)$, la salida del relé es $-d$ (-1), como sucede en $t = 2.5$.

El ajuste de la amplitud de la oscilación de salida se hace a través del valor de 'd' en el relé. Existe una relación proporcional directa, por lo que si disminuimos 'd' ocurrirá el mismo hecho en la salida.

5.5.4.3 ²⁶Obtención de los parámetros primarios, T_c y K_c

En la figura anterior se ve que se produce un transitorio, pero a partir del primer segundo la salida del relé comienza a ser estable, por lo que la variable $y(t)$ se encuentra en oscilación permanente. En este estado es cuando se puede medir directamente el período de oscilación T_c , tal y como se observa en el gráfico, entre dos puntos de la salida que se encuentren en la misma fase.

Para el cálculo de K_c , se debe estudiar la respuesta de la planta y la salida del relé. Para esto se puede únicamente tener en cuenta el primer armónico de la salida del relé (acción de control, $u(t)$) resultante de hacer la descomposición por Fourier, por lo que se obtiene una onda sinusoidal de amplitud $4d/\pi$:



Descomposición por Fourier de la salida del relé

La respuesta de la planta tendrá, por tanto, una forma senoidal de amplitud 'a' (ver gráfico de la página anterior), por lo que la ganancia crítica vendrá dada por la siguiente ecuación:

$$K_c = \frac{4d}{a\pi}$$

²⁶ (Rasmussen, 2002)

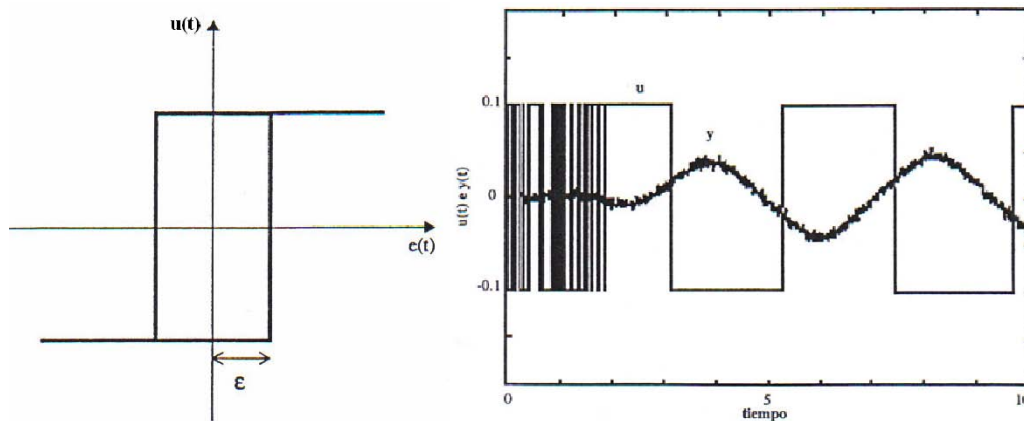
5.5.5 Consideraciones prácticas para la aplicación del método

Como se ha comentado anteriormente, la explicación teórica no es suficiente para la implementación de este método en un sistema real. Ya que es un procedimiento experimental, existen casos en los que, por ejemplo, en consecuencia del ruido de la señal, aparece una conmutación errónea del relé, con lo que la autosintonía no sería correcta o incluso no podría realizarse.

A continuación, se presentan las soluciones reales que se toman ante la aparición de efectos no deseados y ante aspectos que el estudio teórico no deja claros para su aplicación práctica.

5.5.5.1 ²⁷Relé con histéresis

En la práctica, el relé que se utiliza no es ideal, aparece una histéresis, denominada ϵ , por lo que la función de transferencia de este elemento se corresponde con la de la gráfica de la izquierda, siendo $e(t)$ la entrada al relé y $u(t)$ su salida:



Esto es así porque para una correcta conmutación del relé hay que tener en cuenta la existencia de ruido en la salida. En la figura de la derecha, se observa como en el comienzo del experimento el relé conmuta muy rápido debido al ruido existente en la señal de salida. Una vez se corrige con una histéresis en el relé, el efecto se contrarresta y en la misma figura se puede apreciar cómo se retorna al caso considerado ideal.

La anchura de la histéresis se determina sabiendo (o estimando) el ruido existente en el sistema, así que de manera práctica se toma:

$$\epsilon = 2 \cdot \text{Amplitud}_{\text{ruido}}$$

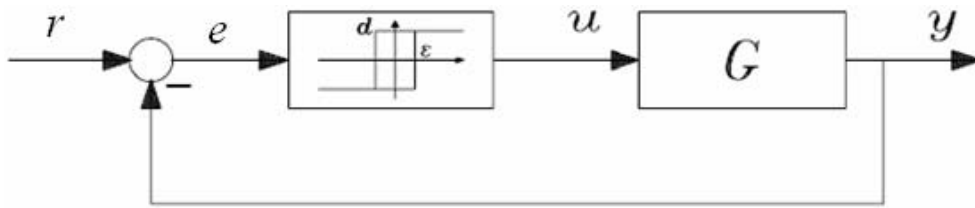
La manera de extraer el período de oscilación crítica, T_c , es idéntica al caso ideal, se mide de los resultados obtenidos en la salida, $y(t)$. Para las aplicaciones en las que se emplee un relé con histéresis se tiene que hacer el siguiente cálculo para determinar la ganancia crítica:

$$\text{crítica: } K_c|_{\text{Histéresis}} = \frac{4d}{\pi \sqrt{a^2 - \epsilon^2}}$$

Siendo ' a ' la amplitud de pico (respecto al SP) de la onda del PV durante la generación del tren de pulsos y ' d ' la amplitud de pico del relé.

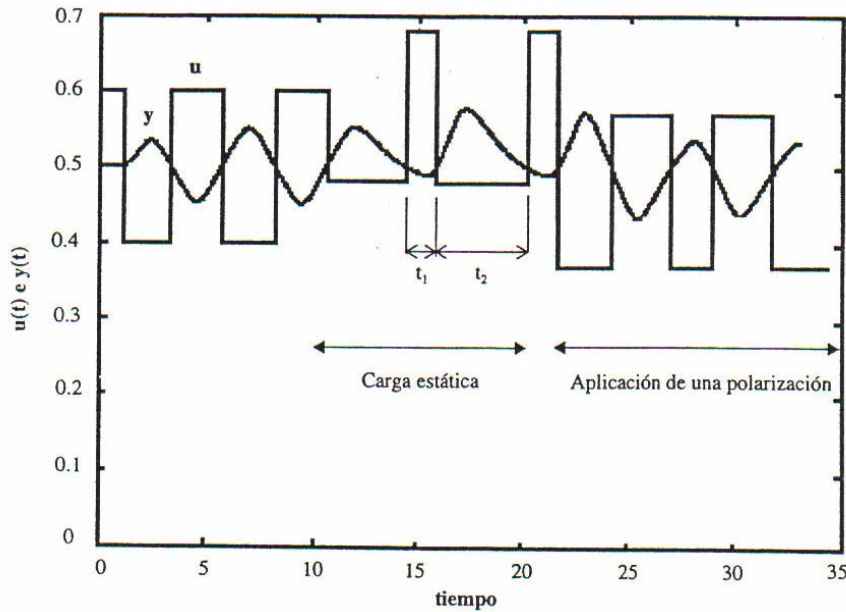
²⁷ (Rasmussen, 2002)

Por tanto, para una experimentación real, se emplea esta ecuación y el siguiente esquema:



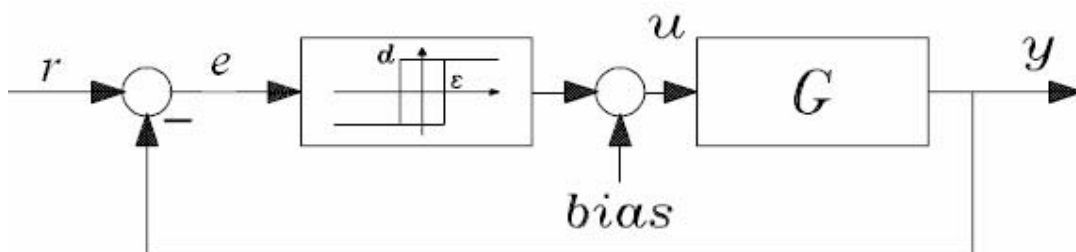
5.5.5.2 ²⁸Offset

La aparición de perturbaciones de tipo estático en la carga o de un offset, causa una asimetría en los tiempos de conmutación del relé por lo que no se puede estimar de forma correcta el valor de K_c y de T_c :



Esto conlleva a la introducción en la salida del relé de lo que se denomina una polarización. Ésta será igual al efecto de la perturbación pero con el signo opuesto para, de ésta manera, volver a establecer la simetría necesaria para la aplicación sin errores de la autosintonía por el método del relé.

Considerando lo anterior, el esquema que representa el sistema completo con la compensación del offset es el siguiente:



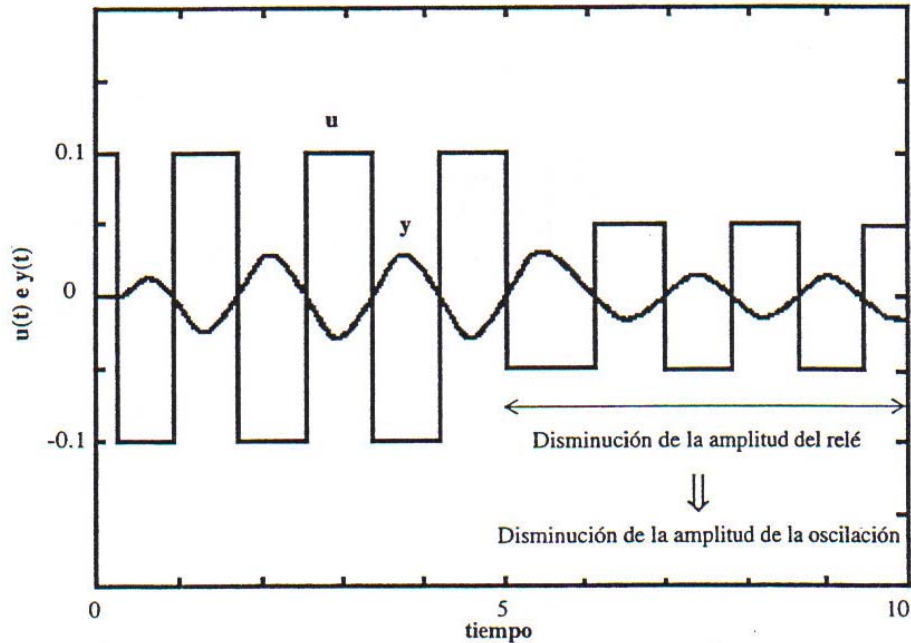
Si ' t_1 ' y ' t_2 ' son los intervalos de tiempo correspondientes a la salida ON y OFF del relé respectivamente, deberá existir una compensación igual a ésta:

$$\Delta bias = d \frac{t_1 - t_2}{t_1 + t_2}$$

²⁸ (Rasmussen, 2002)

El signo de la ecuación vendrá determinado por la relación entre 't1' y 't2'. Para el ejemplo, si 't2' es mayor que 't1', la polarización será de signo negativo, lo que supone que se producirá la eliminación de ese offset y se recuperará la simetría.

5.5.5.3 Ajuste de la amplitud de la oscilación de salida



Para el ajuste de la amplitud de la oscilación permanente se sabe que existe una relación entre ésta y la amplitud del relé. Así, por consiguiente, si se aumenta la amplitud del relé, también se incrementa la amplitud de la señal de salida, y viceversa. Este punto es otra ventaja con respecto a los métodos basados en control P como procedimientos de sintonía, porque si se aumenta la K_p en estos, el sistema se hace inestable ya que la oscilación crece de forma descontrolada. Igualmente ocurre que las disminuciones de la ganancia proporcional provocan que el sistema salga de la oscilación permanente y luego no se podrían medir los parámetros primarios.

Durante el experimento del relé hay que tener en cuenta dos factores con respecto a la amplitud de la oscilación de la variable controlada:

- Por un lado, conviene que la amplitud a la salida de la planta sea pequeña para no dañar al sistema y así evitar saturaciones.
- Pero por otro, la amplitud debe ser mayor al ruido presente en el proceso ya que de lo contrario, éste interferiría de forma negativa en el experimento porque no sería distinguible de la señal de salida.

Ambas cuestiones pueden resultar contradictorias por lo que se debe llegar a un acuerdo entre las dos partes. Tomando en cuenta dichos aspectos, en la práctica se utiliza una amplitud en el relé tal que:

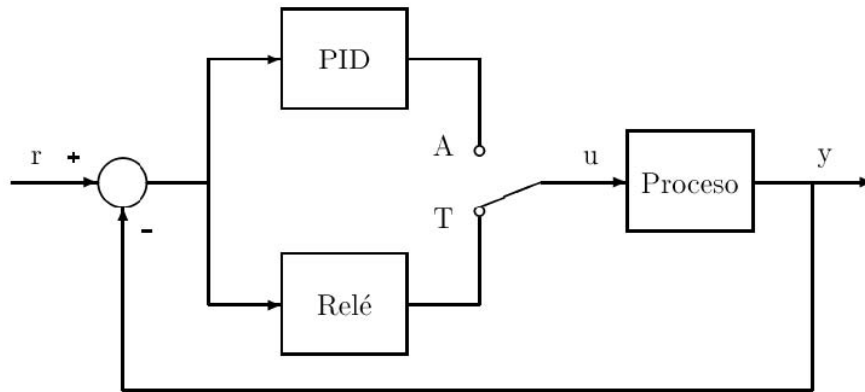
$$Amplitud_relé \Rightarrow Amplitud_oscilación_salida = 3 \cdot Amplitud_ruido$$

5.5.6 Proceso de Autosintonía

Una vez se han comentado las cuestiones prácticas que afectan al desarrollo del método, se explican los pasos para llevar a cabo la sintonización del controlador.

5.5.6.1 ²⁹Experimento con el relé para la obtención de K_c y T_c

La figura siguiente muestra el diagrama de bloques para la aplicación de este procedimiento:



Para el ajuste del PID se realizan los pasos siguientes:

1. Al activarse la opción de autosintonía del controlador, el controlador PID se desconecta y se mide el nivel de ruido. Así, éste deberá ser tenido en cuenta para el cálculo del valor de la histéresis del relé.
2. Se introduce el relé con histéresis (es decir, se está en la posición T de la figura) y se ajusta su amplitud ('d') para conseguir los niveles de la oscilación de salida adecuados según se explicó en puntos anteriores.
3. Una vez la oscilación es estable, de amplitud deseada, se estiman los valores últimos, T_c por inspección directa y K_c por la fórmula:

$$K_c|_{\text{Histéresis}} = \frac{4d}{\pi\sqrt{a^2 - \varepsilon^2}}$$

Siendo 'a' la amplitud de pico (respecto al SP) de la onda del PV durante la generación del tren de pulsos

4. En este punto se memorizan, para posteriores sintonías, los datos de anchura de histéresis y amplitud del relé además de los datos obtenidos.
5. Se calculan a partir de los parámetros límite los valores de K_p , T_i y T_d según la variante PID que corresponda, y el método de sintonía que se quiera utilizar. A continuación, se programan y se pasa al modo A de la figura, entrando en juego el controlador en el sistema realimentado.

A través de la consecución de los cuatro primeros pasos se llega a la obtención de los

²⁹ (Åström, 1995)

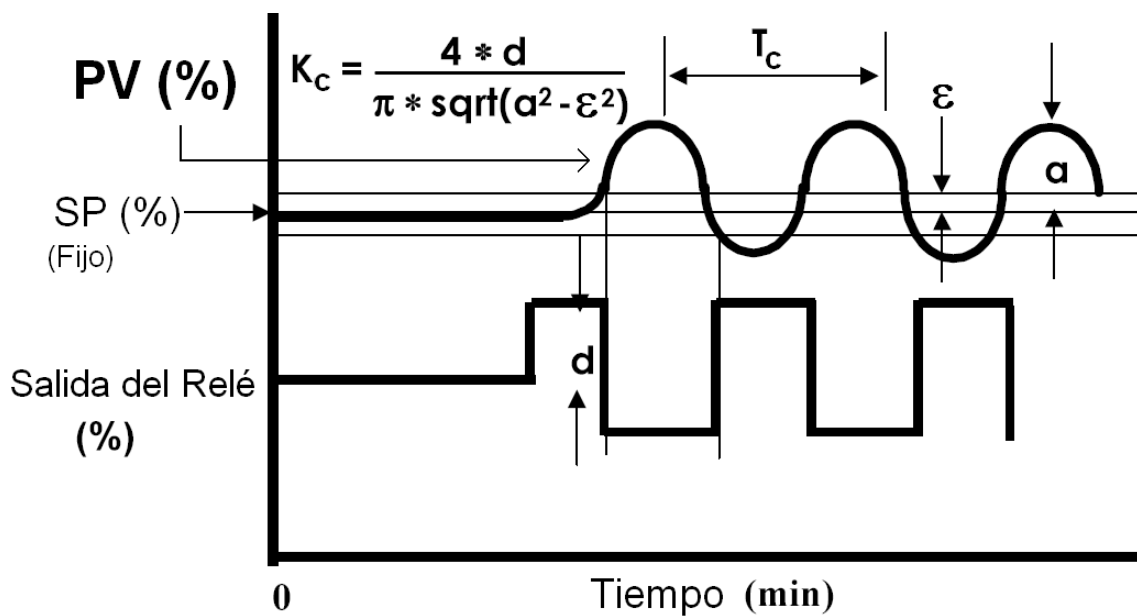
parámetros limítrofes con la estabilidad, por lo que se deduce que aquí acaba la parte de experimentación con el relé propuesta por Aström y Hägglund.

Esto supone que para realmente finalizar la autosintonía del controlador se tienen que aplicar otros métodos para el hallazgo de los parámetros directos de las variantes PID.

Luego de obtenidos K_c y T_c , se utiliza la siguiente tabla (basada en Z-N [1942]) para el cálculo que determina los valores de los parámetros del controlador PID:

Controlador	K_p	T_i	T_d
PI	$0.4 * K_c$	$\frac{T_c}{1.2}$	
PID	$0.6 * K_c$	$\frac{T_c}{2}$	$\frac{T_c}{8}$

La siguiente figura muestra un ejemplo típico de autocalibración por relé:



Autocalibración típica

5.6 Implementación práctica del PID

5.6.1 Introducción

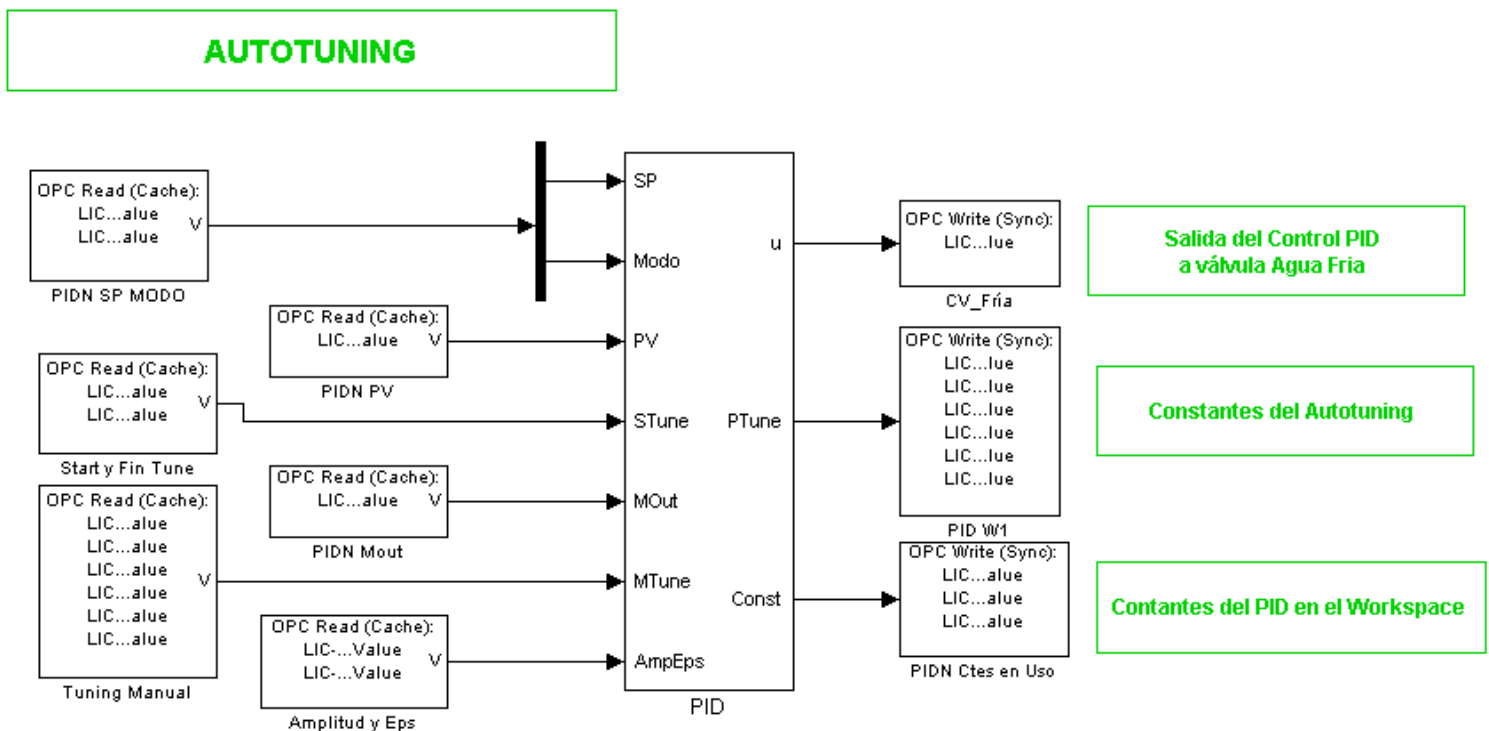
La implementación práctica del PID se basó en el estudio teórico. El PID fue implementado con las siguientes características:

- Bumpless
- Anti-Windup (técnica de ‘Back Calculation’)
- Autosintonizable (método de ‘Relé con Histéresis’)

5.6.2 Implementación en MATLAB

Debajo se muestra la programación de la interconexión entre el PID desarrollado (versión inicial, para mejor comprensión de la implementación) en Matlab y el DCS mediante comunicación OPC.

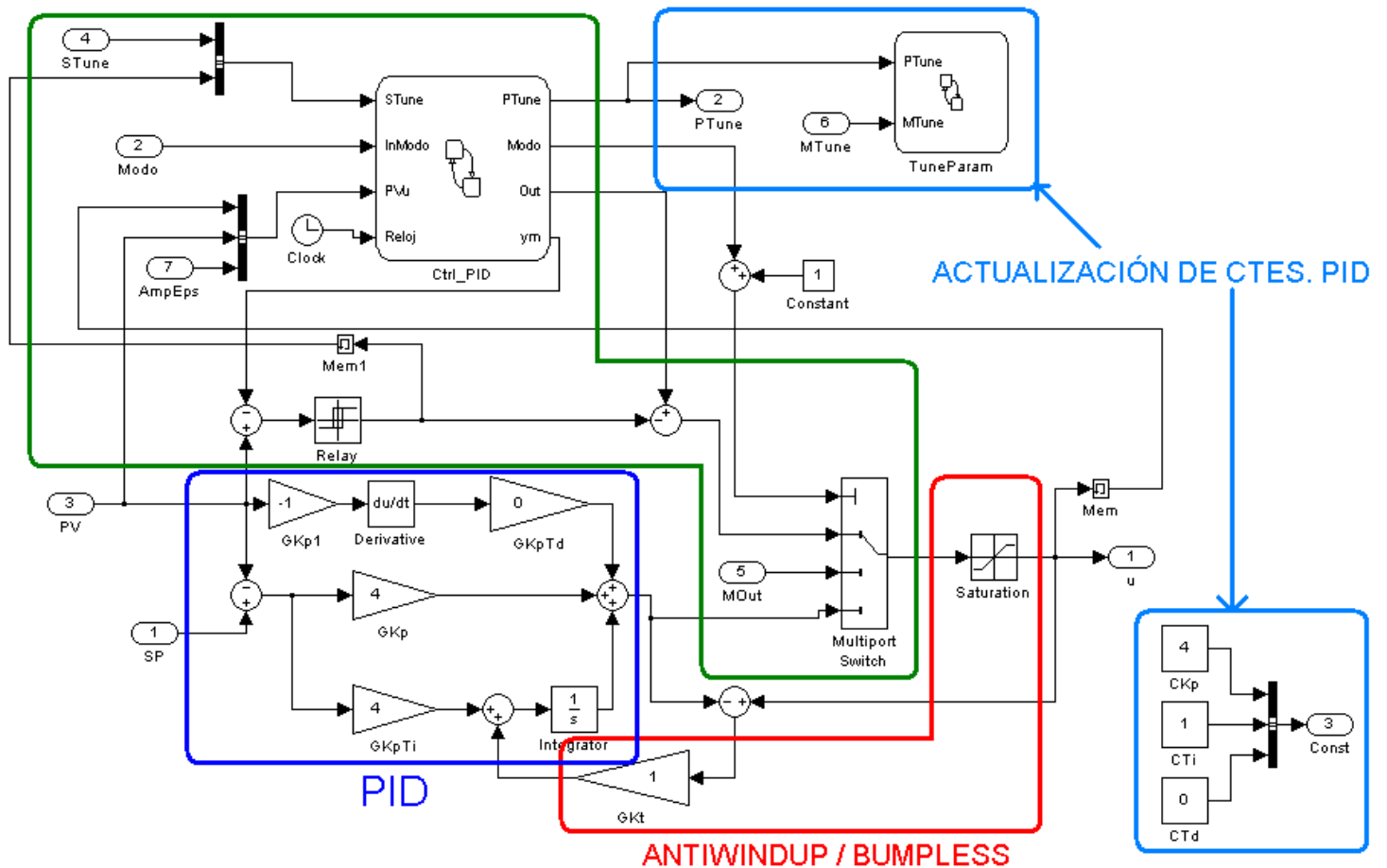
En la figura los bloques Simulink, a la izquierda del bloque PID, (bloques OPC Read) son los encargados de leer la información proveniente del DCS, por ejemplo variables tales como setpoint (SP), process value (PV), etc. Los bloques a la derecha (OPC Write) son los encargados de escribir datos al sistema DCS, por ejemplo apertura de válvula y constantes del Autotuning.



Interfaz de un PID con Antiwindup, Bumpless y con autosintonía de parámetros

En la figura a continuación se muestra la implementación en Matlab del controlador PID propuesto (estructura interna del PID).

CONFIGURACIÓN PID / AUTOSINTONÍA

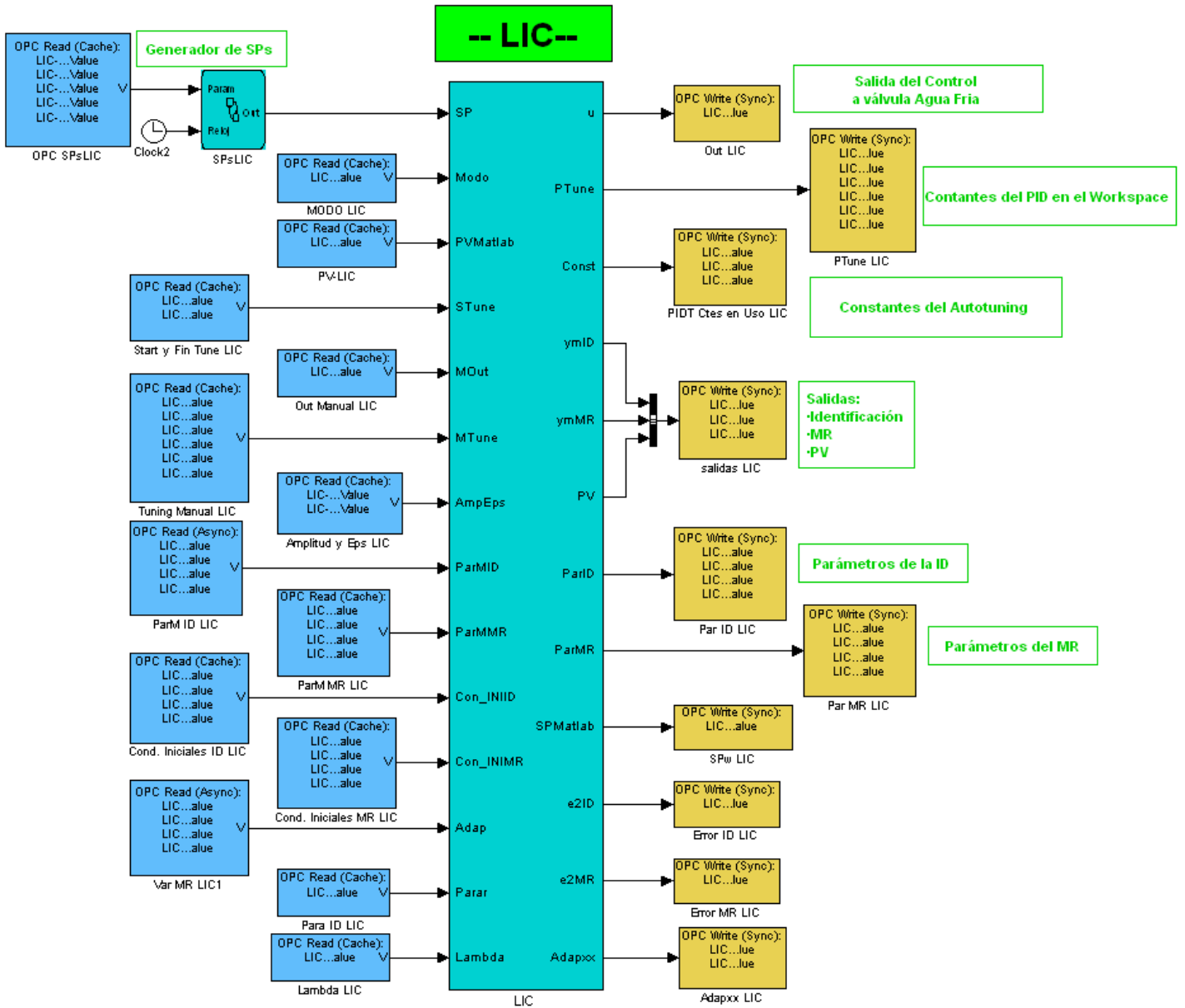


Lo encerrado en azul es la implementación del PID con $c = 0$ (PID con ponderación de referencia, ver 5.3.3), esta implementación es muy usada en los PLCs actuales. En color rojo se encierra la implementación del antiwindup (basado en la técnica de Back-Calculation, ver 5.4.2.5) y bumpless.

Con color verde se destacan los bloques encargados de configurar el PID, por ejemplo: para utilizarlo en modo automático o manual, ejecutar la autosintonía o cambiar las constantes del mismo. En la parte superior derecha se recuadra el bloque 'TuneParam', éste se encarga de actualizar los valores de los parámetros del PID. Finalmente en la parte inferior derecha de la imagen se muestran los valores actuales de los parámetros.

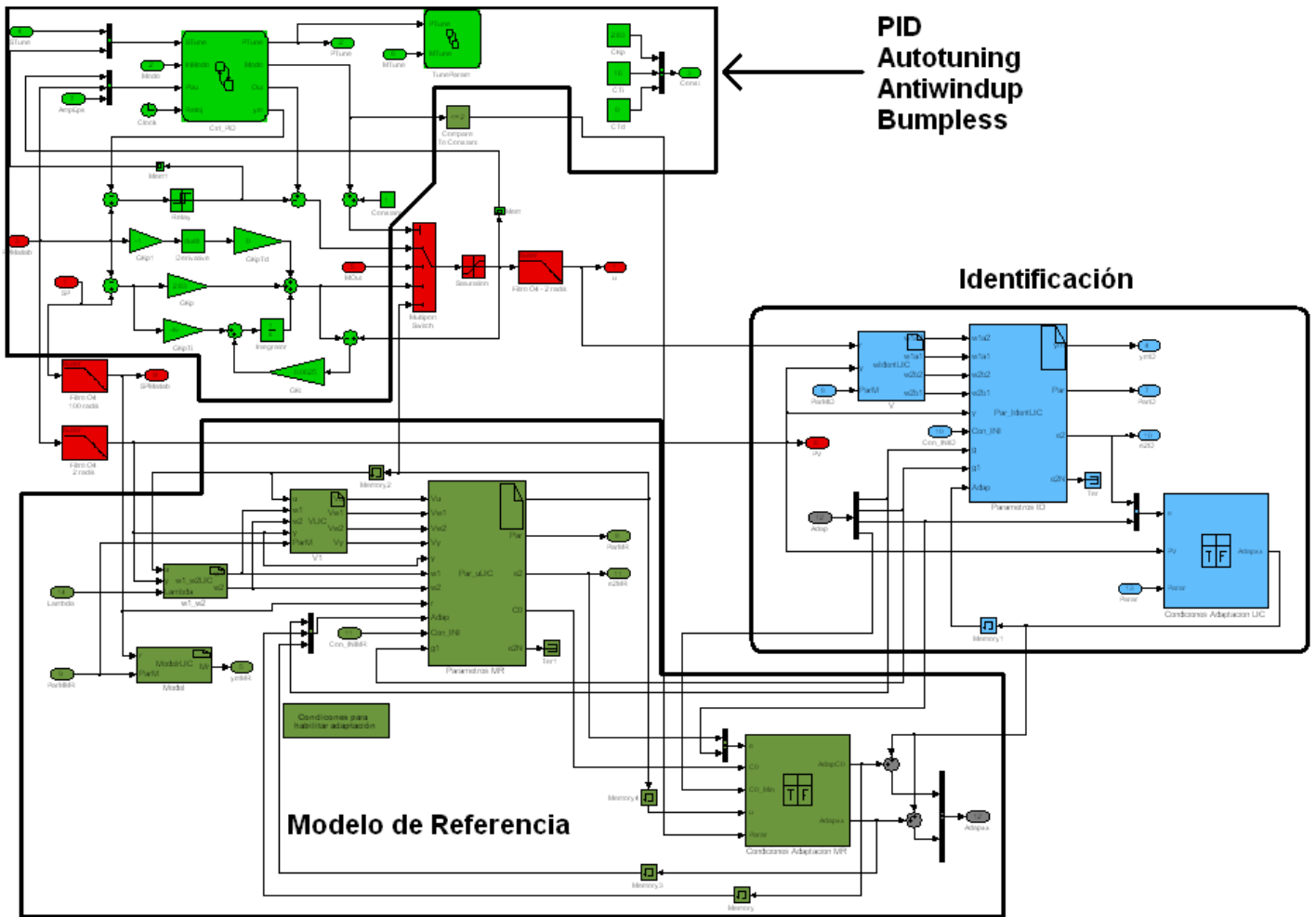
Dado que el PID es controlado desde el DCS (tanto sus constantes como el Autotuning se configuran desde una pantalla del sistema) se debieron desarrollar objetos con aspectos, faceplates y programas que son capaces de interactuar con Matlab y el PLC.

Actualmente con el agregado adicional de la identificación y el MR, temas que se explicarán más adelante, el controlador (ya no es sólo el PID anterior) desarrollado presenta la siguiente interfaz:



Interfaz actual del controlador de nivel

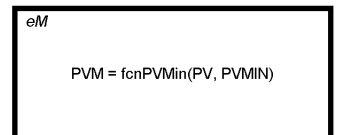
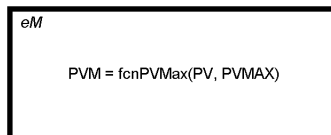
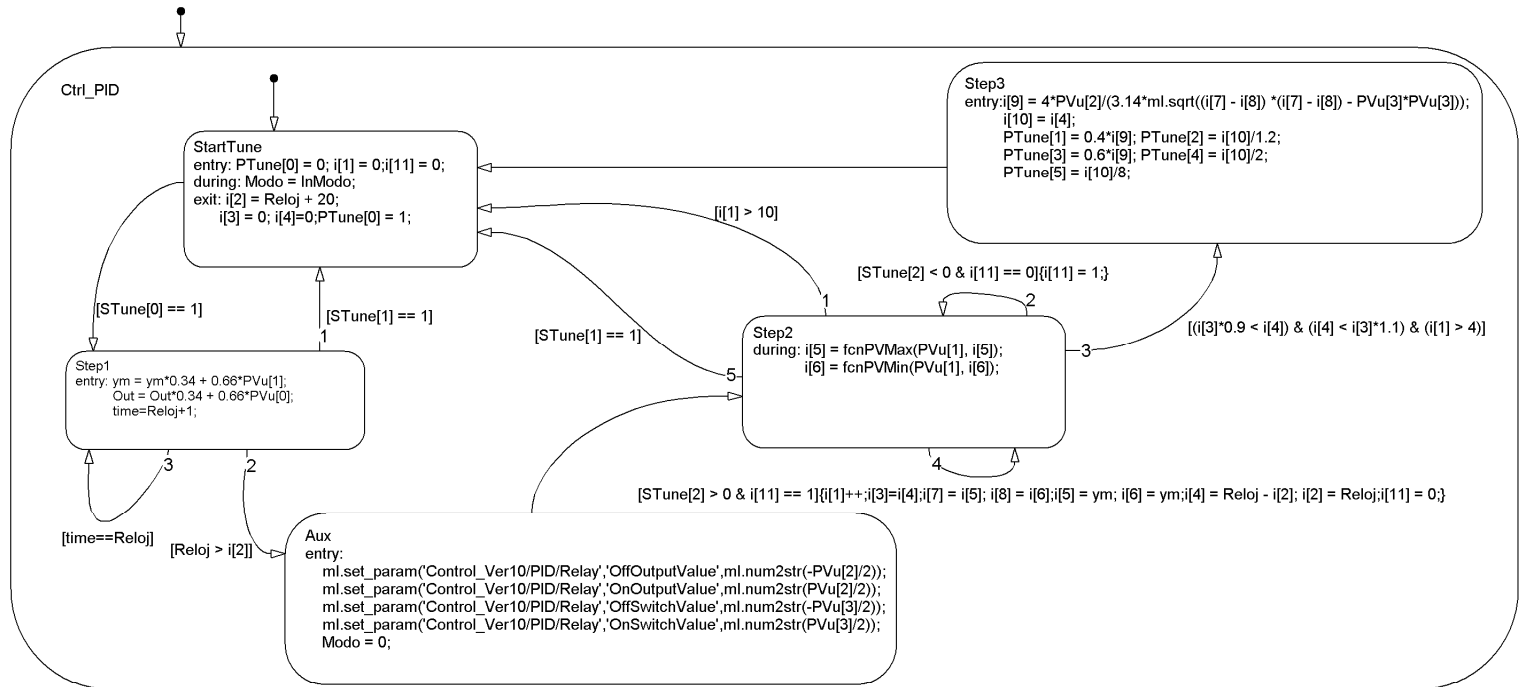
La estructura interna del controlador se puede separar en tres grandes grupos, los bloques del PID con Autotuning, los del Modelo de Referencia y los dedicados a la Identificación:



5.7 Implementación del Autotuning

5.7.1 Introducción

Una vez comprendido el método de autotuning de Relé con Histéresis se comenzó con su programación en MATLAB. Para ello y dado que se siguen una secuencia de pasos dados, se decidió la implementación del mismo en una máquina de estados como la que se muestra a continuación:



Notación:	
<ul style="list-style-type: none"> ○ i[1]: contador de pulsos del Relé ○ i[2]: tiempo de filtrado ○ i[3]: período anterior ○ i[4]: período actual ○ i[5]: PVMax actual ○ i[6]: PVMin actual ○ i[7]: PVMax Total ○ i[8]: PVMin Total ○ i[9]: Kcrítico (ganancia crítica) ○ i[10]: Tcrítico (período crítico) ○ i[11]: bandera que indica que existió transición en determinados estados 	<ul style="list-style-type: none"> ○ PTune[0]: bandera que indica si se está realizando un tuning ○ PTune[1]: K_p del PI ○ PTune[2]: T_i del PI ○ PTune[3]: K_p del PID ○ PTune[4]: T_i del PID ○ PTune[5]: T_d del PID ○ STune[0]: Empieza el Tuning ○ STune[1]: Finaliza abruptamente Tuning ○ STune[2]: salida del bloque del relé ○ PVu[0]: OUT, salida de control ○ PVu[1]: PV, process value ○ PVu[2]: amplitud pico pico de salida de relé ○ PVu[3]: amplitud de la ventana de triggeo

5.7.2 Explicación del programa

Estado “StartTune”:

En la espera por el comienzo a un nuevo tuning la máquina permanece en el estado “StartTune”. A la entrada se resetea la salida ‘Ptune[0]’, lo que indica todavía no hay un tuning en proceso; de la misma manera se resetean las variables internas ‘i[1]’ (se lleva a cero el contador de pulsos de relé) e ‘i[11]’ (bandera utilizada para indicar si existió o no transición en algunos estados). Mientras no se salga del estado, el modo de operación (salida Relé, salida manual o salida PID) se copia en la variable ‘Modo’ y a la salida del estado en la variable ‘i[2]’, que se utiliza para contar 20 segundos, se guarda el valor de $RELOJ+20$, siendo RELOJ el reloj actual. Además se resetean ‘i[3]’ e ‘i[4]’ que se utilizan para guardar los valores de los períodos anterior y actual de la señal de relé y se setea la salida ‘Ptune[0]’ en ‘1’ para indicar que hay un tuning en proceso.

Estado “Step1”:

Cuando se da comienzo a un nuevo tuning, Stune[0] pasa a valer ‘1’, se sale del estado de espera de tuning y se ingresa en el estado “Step1”. La inclusión de este estado es para filtrar las señales de PV (process value) y OUT (salida de control), y de esta manera obtener la media de cada una. El filtrado (promediado) de las señales dura unos 20 segundos y se realiza cada un segundo, mientras no transcurran los 20 segundos, o sea mientras $RELOJ < = i[2]$, se re-ingresará en este estado y se volverán a realizar las siguientes cuentas:

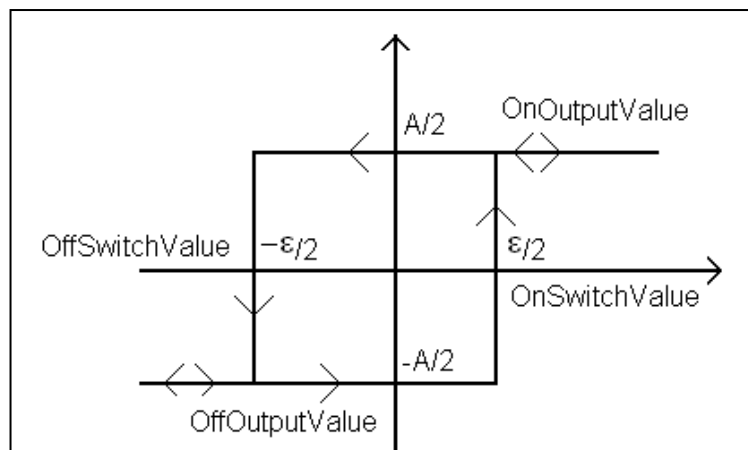
- $y_m = y_m \cdot 0.34 + 0.66 \cdot PV_u[1]$ (ym es la media actual de la señal PV)
- $Out = Out \cdot 0.34 + 0.66 \cdot PV_u[2]$ (Out es la media actual de la señal OUT)

Simplemente cada 1 segundo se acumula el valor de la media de cada señal. En el siguiente segundo la nueva media es el 34% del valor de la anterior más el 66% del valor de la señal (según corresponda) actual. Los valores de las ponderaciones fueron elegidos de forma que $34\% + 66\% = 100\%$.

Luego de transcurridos 20 segundos finaliza el filtrado y se sale al estado “Aux”.

Estado “Aux”:

Cómo su nombre lo indica, éste es un estado auxiliar que se utiliza para ajustar unas propiedades del bloque que genera los pulsos de relé. En concreto se setean los valores de salida de relé y los valores de switcheo:



Finalmente se resetea la variable “Modo” para que se seleccione, en el bloque de jerarquía superior, la salida proveniente del relé y no la manual o la propia del PID y se salta al estado “Step2”

Estado “Step2”:

Éste estado se encarga de encontrar el PV máximo y el PV mínimo mientras se generan los pulsos de relé. En la variable “i[5]” se almacena el máximo entre PVmáximo de la iteración anterior y el PVactual y en la variable “i[6]” el mínimo entre PVmínimo de la iteración anterior y el PVactual.

Este es un estado recurrente del cual se sale cuando:

- a. se finaliza abruptamente el tuning mediante un ‘1’ en la variable “STune[1]”
- b. si después de 11 períodos de relé no hay un Tcrítico estable
- c. si se terminó el tuneado y se deben cargar las contantes respectivas en los bloques.

La condición de finalización normal del tuning es que el valor del último período medido esté a menos de un 10% del valor del período anterior y además hayan pasado más de 4 pulsos.

En el transcurso normal ocurriría lo siguiente:

- 1 una vez que el sistema entra en este estado se espera a que la salida del bloque de relé sea negativa para setear la bandera de transición i[11]
- 2 una vez que la salida del bloque de relé se hace positiva y además existió una transición (pues i[11] debe ser 1)
 - 2.1 se incrementa i[1] (contador de pulsos) porque transcurrió un período de la onda de pulsos de relé
 - 2.2 se carga en i[3] (período obtenido de la iteración anterior) el valor de i[4] (período actual)
 - 2.3 en i[7] (PV máximo) se carga i[5] (PV máximo actual)
 - 2.4 en i[8] (PV mínimo) se carga i[6] (PV mínimo actual)
 - 2.5 en i[5] e i[6] se guarda la media del PV
 - 2.6 en i[4] se guarda la diferencia entre el valor actual de RELOJ y el valor de RELOJ en la transición anterior, en definitiva se guarda el valor del período actual
 - 2.7 en i[2] se almacena el valor actual de RELOJ para en la próxima transición calcular el período (tiempo transcurrido)
 - 2.8 se resetea i[11] para de ésta manera no volver a ejecutar todos estos pasos hasta que transcurra un período
- 3 luego de que el período se mantenga estable en determinado rango y hayan pasado más de 4 períodos se pasa al estado “Step3”

Estado “Step3”:

Una vez culminado el tuning este estado se encarga de dar como salidas las constantes obtenidas. Para ello se debe calcular la $K_{crítica}$ como:

$$K_{crítica} = i[9] = \frac{4 * d / 2}{\pi * \sqrt{(A/2)^2 - (\varepsilon/2)^2}} = \frac{4 * PVu[2] / 2}{\pi * \sqrt{\left(\frac{i[7] - i[8]}{2}\right)^2 - \left(\frac{PVu[3]}{2}\right)^2}} \Rightarrow$$
$$\Rightarrow i[9] = \frac{4 * PVu[2]}{\pi * \sqrt{(i[7] - i[8])^2 - (PVu[3])^2}}$$

El $T_{crítico}$ es igual a $i[4]$, que es el período de la señal del relé una vez estable.

- Si el controlador deseado es un **PI** las constantes del mismo se calculan como:

$$Kp = 0.4 * K_{crítica} \Rightarrow Ptune[1] = 0.4 * i[9]$$

$$Ti = \frac{T_{crítico}}{1.2} \Rightarrow Ptune[2] = \frac{i[10]}{1.2}$$

- Si el controlador deseado es un **PID** las constantes del mismo se calculan como:

$$Kp = 0.6 * K_{crítica} \Rightarrow Ptune[3] = 0.6 * i[9]$$

$$Ti = \frac{T_{crítico}}{2} \Rightarrow Ptune[4] = \frac{i[10]}{2}$$

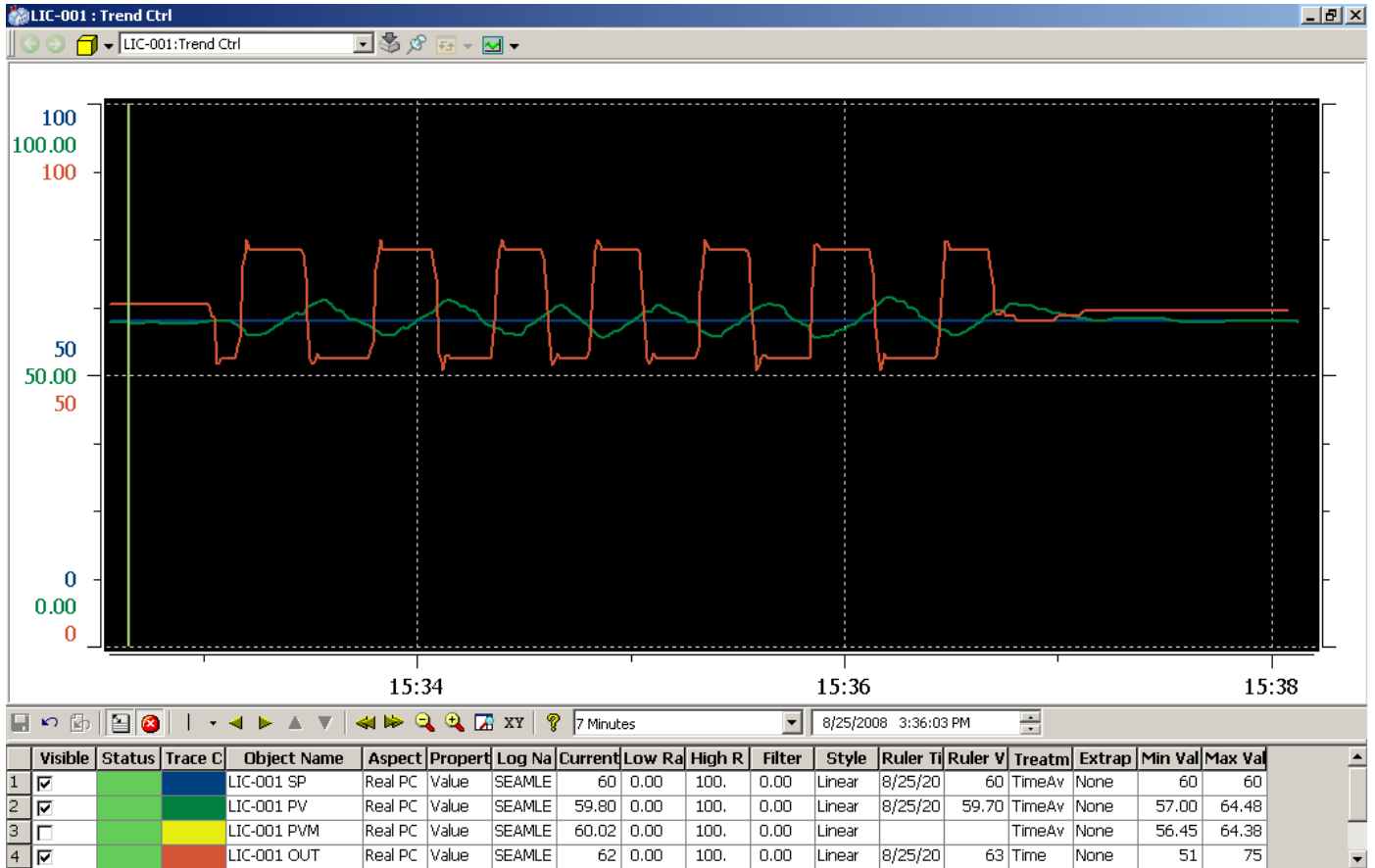
$$Td = \frac{T_{crítico}}{8} \Rightarrow Ptune[5] = \frac{i[10]}{8}$$

Finalmente, luego de calculadas las constantes, se vuelve al estado inicial a la espera de un nuevo tuning.

5.7.3 Pruebas

Para testear el procedimiento de autotuning implementado se lo probó en el PID de nivel. Se configuró el tuning con una amplitud pico a pico para los pulsos de relé del 20% y un ε pico a pico del 3%.

Con la planta controlando en un SP del 60% del nivel se inició el tuning.



Luego de 20 segundos comenzaron a generarse los pulsos de relé, y después de 3 minutos el tuning terminó obteniéndose las constantes para controladores PI y PID que se detallan en la figura siguiente:

Rele H	Param	Act Param
P Manuales		P Automaticos
KP	2.83	1.58 2.38
Ti	16.00	27.51 16.51
Td	5.00	4.13
APPLY		PID PI
		Man Auto

Para probar las constantes obtenidas, se configuró el controlador en PID y se cargaron las constantes del Autotuning. Luego se generaron SPs fijos (escalones), primero en 60%, luego en 40% y finalmente en 70%, observándose la siguiente respuesta:



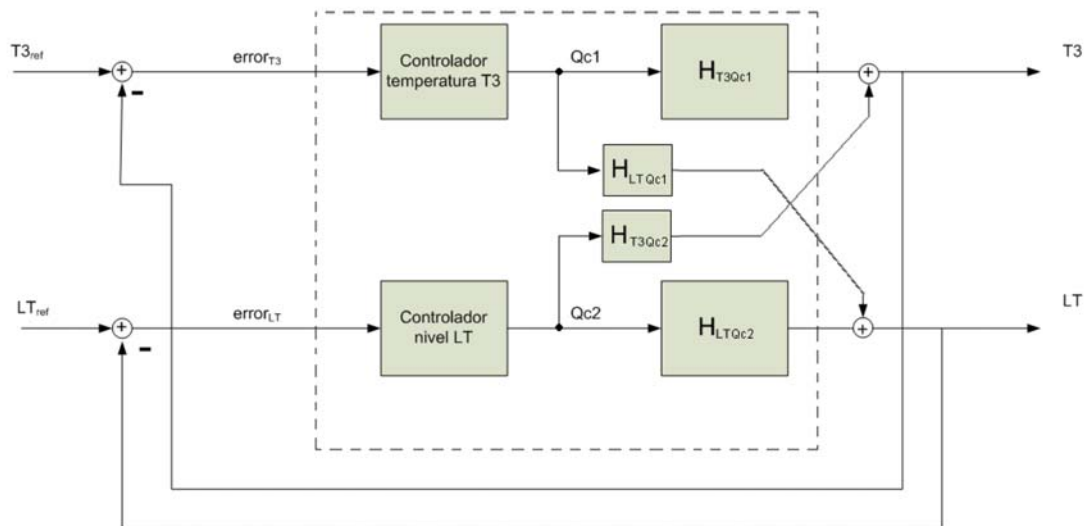
Vale la pena mencionar que las pruebas en este caso fueron hechas para el controlador de nivel, pero valen los mismos comentarios y conclusiones para el de temperatura. Lo único que se debe tener en cuenta, a la hora de realizar un tuning a dicho controlador, es elegir una temperatura media de trabajo pequeña y con amplitudes de pulsos de relé grandes, del entorno a 50%pp (50% pico a pico) de manera que se tengan grandes ΔT y no disminuya drásticamente la temperatura del agua caliente del calefón.

5.7.4 Control Paralelo de Temperatura y Nivel

Una vez testeado, en la planta física, el controlador de Nivel, se procedió a controlar nivel y temperatura a la vez. Aquí se hizo la hipótesis de que como la temperatura tiene una gran inercia, por lo tanto varía más lentamente que el nivel. En un buen diseño el nivel debe responder rápido a perturbaciones. Además para las dos respuestas es deseable un sobretiro bajo (sobrepaso máximo pequeño), para que de ésta manera se evite que la variación en una de las variables controladas influya de manera perceptible en la otra.

En la siguiente ³⁰figura se muestra un esquema del diseño ideal:

³⁰ (Berenfus, Casas, & Rezk, 2006)



Donde:

H_{T3Qc1} : Función de transferencia entre $T3$ y $Qc1$.

H_{LTQc2} : Función de transferencia entre LT y $Qc2$.

H_{T3Qc2} : Función de transferencia entre $T3$ y $Qc2$.

H_{LTQc1} : Función de transferencia entre LT y $Qc1$.

³¹ Inicialmente como muestra la figura anterior se considera que cada variable influye en el comportamiento de la otra. Las perturbaciones de la temperatura en el nivel y del nivel en la temperatura luego en un modelo simplificado (que es utilizado para el diseño del controlador) se consideran despreciables.

Para esta prueba se realizó un autotuning de nivel, obteniéndose el siguiente juego de valores:

Kp	2.83
Ti	16.00
Td	4.00

Seguidamente se configuró de manera arbitraria el TIC para que funcionase como PI con constantes $K_p=2$ y $T_i=20$, se eligieron setpoints constantes de 30% para nivel y 30°C para temperatura y se dejó los controladores marchando. Luego de un rato se dio un escalón a 60% en nivel, la respuesta (como puede verse en el gráfico a continuación) tanto en nivel como en temperatura, es bastante aceptable por lo que se procedió a dar un escalón, en temperatura, a 45°C. Esta última respuesta de temperatura es muy oscilante y provoca un sobrepico importante en nivel. Esto indica que, de la manera que se eligieron los parámetros para el TIC, se viola la hipótesis mencionada unos párrafos antes. Por este motivo se decidió cambiar los parámetros de controlador (del TIC) a: $K_p=0.75$ y $T_i=15$. Luego se fijó el SP de temperatura en 30°C obteniéndose una respuesta bastante mejor, menos oscilante, pero más lenta.

³¹ (Berenfus, Casas, & Rezk, 2006)



Control paralelo de temperatura y nivel, LIC en modo PID y TIC en modo PI

5.8 Conclusiones PID

La implementación del PID utilizada se encuentra bien documentada en varios libros de control. Se decidió optar por la implementación descrita en estos libros ya que tienen un enfoque práctico y resuelven problemas que en el PID clásico no se tienen en cuenta, como por ejemplo que la señal de control está limitada y a causa de esto aparecen problemas con la parte integral del PID (fenómeno del Windup Integral). También se sugieren cambios en la parte derivativa para que el ruido que acompaña la señal no tome gran fuerza y logre a tornar inestable la planta.

Además existen métodos prácticos de sintonía de dichos controladores (reglas de Ziegler y Nichols [1942]) que permiten que sean muy sencillos de sintonizar inclusive para personas con muy poco conocimiento en materia de control.

Para hacer una comparación teoría-práctica, se deben contrastar la implementación del PID efectuada y el PID clásico (ideal). Los resultados prácticos relevados para el PID implementado fueron los esperados, esto era previsible porque la implementación fue basada en documentos que tratan de cómo implementar un PID para la realidad y no en la teoría.

En primer lugar el PID implementado tiene los modos manual-automático, donde los pasajes de un modo al otro pueden tener saltos bruscos de la salida de control. Para solucionar este inconveniente, se implementó la solución Bumpless que ayuda a eliminar este problema descrito anteriormente.

Otro problema que aparece en la realidad es el fenómeno del Windup del integrador. El mismo se solucionó implementando una técnica de anti-windup llamada 'Back-Calculation'. Dicha técnica además de solucionar los problemas que presenta un PID teórico en la realidad, viene acompañada de reglas prácticas para elegir los parámetros que entran en juego.

En las implementaciones prácticas el término derivativo sufre cambios importantes. En lugar de derivar el error (SP-PV) solamente se deriva el PV, y además ésta derivada no es una derivada real (s) sino que es un filtro pasa altos. Con esto se gana que a baja frecuencia su comportamiento es de derivador y que a alta frecuencia el ruido se vea amplificado sólo por una constante.

De acuerdo a la bibliografía consultada, en situaciones en las cuales los PID no cumplen con las especificaciones requeridas, aparece una fuerte tendencia a empezar a implementar otros métodos de control como por ejemplo MPC, GPC, etc.

6 Aplicación de control Avanzado

6.1 Introducción

Una vez estudiadas en detalle las distintas partes del sistema y solucionados la mayoría de los inconvenientes encontrados en el camino, los esfuerzos se concentraron en comenzar a estudiar posibles implementaciones de control avanzado. Con el tutor se trazó como meta lograr que el nivel o la temperatura en el tanque de la planta se comporte de acuerdo a un modelo de segundo orden deseado. Para lograr ésta meta se trazaron varios objetivos más pequeños de manera de dividir el problema y atacarlo desde todos los puntos vista posible.

Estos objetivos son:

- 1- Autocalibración de los instrumentos.
- 2- Adquisición de señales de interés de la planta para el control avanzado.
- 3- Estudio de las distintas técnicas de control avanzado.
- 4- Diseño y simulación del control.
- 5- Implementación del control.

6.2 Resumen de los distintos objetivos

6.2.1 Autocalibración de los instrumentos

Parte: DCS 800xA

Descripción: Con este objetivo se busca encontrar una manera sencilla y rápida de calibrar todos los instrumentos que integran la planta. Para cumplir este objetivo se planteó crear y documentar un procedimiento el cual sería programado en el PLC ABB KR 51. Luego de un estudio posterior se concluyó su implementación en Matlab. Actualmente hay un programa que realiza la autocalibración, el mismo se encuentra terminado pero se decidió no incluirlo pues una buena calibración llevaría mucho tiempo y consumiría muchos recursos hardware. Para compensar esto se diseñó una pantalla (faceplate) dentro del sistema dedicada a la calibración de todos los sensores y actuadores. Además se documentó un procedimiento muy sencillo de seguir.

6.2.2 Adquisición de señales

Parte: PLC ABB KR 51, Sistema 800xA

Descripción: Este objetivo consta de almacenar la información en históricos, o sea historizar las señales que influyen en el control de distintas situaciones para después procesarlas con algún software que permita crear un estudio más profundo de las mismas (por ej.: Matlab).

6.2.3 Estudio de las distintas técnicas de control avanzado

Parte: PLC ABB KR 51, Sistema 800xA, Matlab

Descripción: En este punto se concentraron los esfuerzos en estudiar las distintas técnicas de control avanzado. Se comenzó buscando bibliografía³² referente de: controles adaptativos, redes neuronales, fuzzy y MPC. Al comenzar a leer estas bibliografías se concluyó que para poder estudiar todos estos temas se necesitaría más tiempo del que se disponía, por eso se decidió buscar un tipo de control avanzado que se adapte a la aplicación que se está desarrollando y además que su implementación y estudio permita cumplir con los plazos del proyecto. De este estudio, que no fue muy exhaustivo a causa de la gran cantidad de técnicas existentes, se decidió implementar un control por modelo de referencia con error a la salida. Posterior a esto, dados los no tan buenos resultados obtenidos se implementó además un control por modelo de referencia con error a la entrada. En la versión final del programa se incluyeron bloques de identificación de plantas de 2° y 1° orden para nivel y temperatura. Con esto se logran estimar las dos transferencias en cuestión lo que facilita la realización de controles adaptativos. Además gracias a ello se logran realizar simulaciones más realistas pues se llegan a muy buenas estimaciones para las ambas funciones de transferencia.

6.2.4 Diseño y simulación del control avanzado

Parte: PLC ABB KR 51, Sistema 800xA, Matlab

Descripción: Los controles fueron simulados en plantas hipotéticas (programadas en Matlab) de primer y segundo orden. Para plantas de segundo orden los mejores resultados de control se obtuvieron con el modelo de referencia de error a la entrada, mientras que para sistema de primer orden ambos modelos se comportan de buena forma.

6.2.5 Implementación

Parte: PLC ABB KR 51, Sistema 800xA, Matlab

Descripción: De las pruebas reales con la planta piloto para validar las técnicas de control avanzado se concluyó con la muy buena performance del modelo de referencia con error a la entrada, descartando el de error a la salida. Además se diseñaron controladores PID con identificación que dada su robustez fueron incluidos como controladores principales para ambos lazos de control. El algoritmo de MR con error a la salida queda en su versión de código y lo suficientemente documentado como para ser empleado. El esquema de MR con error a la entrada se agregó en cada controlador, en el de nivel y en el de temperatura, para de esta manera optar por control Manual, PID o MR.

³² (Sastry & Bodson, 1989)

6.3 Control Adaptivo

6.3.1 Introducción

³³ La adaptabilidad, cualidad propia de los seres vivos y en especial de los seres humanos, permite la evolución y el desarrollo al propiciar los cambios necesarios para resistir y aceptar las modificaciones en el entorno. Por eso, el control adaptivo o adaptativo se basa en la posibilidad de acomodar el funcionamiento de la planta o del sistema de control, tanto a los cambios ambientales como a los pequeños errores de diseño, además de a las fallas leves en algún elemento del sistema sin que se pierda el nivel óptimo de operación.

El control adaptivo despertó el interés de los diseñadores al tiempo en que la industria se introdujo en los procesos de mayor automatización. Debido a la complejidad de estos procesos, éste se volvió una necesidad por cuanto permite una mayor autonomía del sistema y reduce el error propiciado por las variaciones en el ambiente interno o externo de dicho sistema.

Un control adaptivo es especialmente importante en el caso de una planta (o sistema) que se vea afectada por cambios ambientales que modifiquen sus variables cada cierto tiempo.

Lo primero que se plantea en el control adaptivo es: ¿cuál es el funcionamiento que se desea?, es decir, el nivel óptimo de operación de la planta, pues con base en esto, se escoge el ámbito de los parámetros que requiere modificar la señal de control.

6.3.2 Control Adaptivo

³⁴ A mediados del siglo XX se realizaron los primeros intentos de control adaptivo, sin mucho éxito pues las aplicaciones requerían un mayor desarrollo tecnológico que el existente en la época. A partir de la implementación de las computadoras es que se logró aplicar la teoría desarrollada alrededor del concepto de control adaptivo.

Dos de los ejemplos de los primeros controles adaptivos son el control adaptivo del modelo de referencia cuyo propósito es lograr que un sistema actúe como un modelo de referencia propuesto y el segundo es el controlador autosintonizable (autotuning).

6.3.3 Definición

³⁵ “Un sistema de control adaptivo es un sistema que continua y automáticamente mide las características dinámicas (como la función de transferencia) de la planta, las compara con las características dinámicas deseadas y usa la diferencia para variar parámetros ajustables del sistema (generalmente características del controlador) o para generar una señal de accionamiento de modo que pueda mantener el funcionamiento óptimo con independencia de las variables ambientales, alternativamente este sistema puede medir continuamente su propio funcionamiento de acuerdo con un índice de comportamiento dado y modificar, en caso de ser necesario sus propios parámetros para mantener el funcionamiento óptimo con independencia de los cambios ambientales.”

³³ (Ballestero, y otros, 2001)

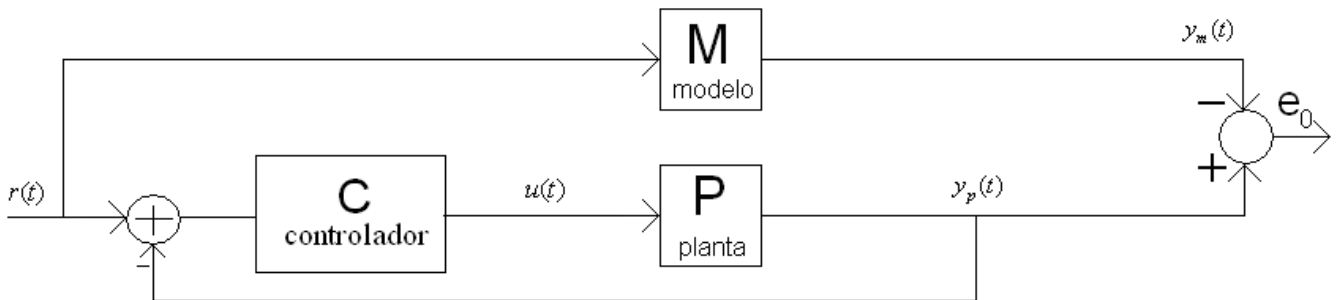
³⁴ (Ballestero, y otros, 2001)

³⁵ (Ogata, 2001)

6.4 Objetivo

De aquí en más se analizarán algoritmos de control adaptivo. La atención se centrará en el control adaptivo por modelo de referencia. El objetivo es diseñar un control adaptivo de manera tal que el comportamiento de la planta a controlar se mantenga lo más cercano posible al modelo deseado independientemente de las variaciones e incertidumbres de los parámetros de la planta.

Formalmente, dado un modelo de referencia \hat{M} , con entrada $r(t)$ y salida $y_m(t)$. La planta desconocida \hat{P} tiene entrada $u(t)$ y salida $y_p(t)$. El objetivo del control es diseñar $u(t)$ de manera tal que $y_p(t)$ siga asintóticamente a $y_m(t)$.



6.5 ³⁶ Introducción al Control adaptivo

Este punto está basado en una traducción libre del texto de S. Sastry y M. Bodson: “*Adaptive Control- Stability, Convergence, and Robustness*” (Capítulo 3), publicado en 1989.

Se considera una planta SISO y LTI con la siguiente función de transferencia:

$$\hat{P} = \frac{k_p}{s + a_p} \quad (6.5.1)$$

donde k_p y a_p son desconocidos. El modelo de referencia es un sistema estable SISO LTI del mismo orden:

$$\hat{M} = \frac{k_m}{s + a_m} \quad (6.5.2)$$

donde k_m y $a_m > 0$ son arbitrarios elegidos por el diseñador. En el dominio del tiempo, la planta se describe por:

$$\dot{y}_p(t) = -a_p y_p(t) + k_p u(t) \quad (6.5.3)$$

y el modelo de referencia por:

$$\dot{y}_m(t) = -a_m y_m(t) + k_m r(t) \quad (6.5.4)$$

La entrada del controlador será dada por:

$$u(t) = c_0(t)r(t) + d_0(t)y_p(t) \quad (6.5.5)$$

³⁶ (Sastry & Bodson, 1989)

La motivación es que existen valores nominales de los parámetros:

$$c_0^* = \frac{k_m}{k_p} \quad d_0^* = \frac{a_p - a_m}{k_p} \quad (6.5.6)$$

de manera tal que la función de transferencia en lazo cerrado copia la función de transferencia del modelo de referencia. Específicamente, de (6.5.3) y (6.5.5):

$$\begin{aligned} \dot{y}_p(t) &= -a_p y_p(t) + k_p (c_0(t)r(t) + d_0(t)y_p(t)) \\ \dot{y}_p(t) &= -(a_p - k_p d_0(t))y_p(t) + k_p c_0(t)r(t) \end{aligned} \quad (6.5.7)$$

que se convierte en: $\dot{y}_p(t) = -a_m y_p(t) + k_m r(t) \quad (6.5.8)$

Entonces se demostró que cuando $c_0(t) = c_0^*$, $d_0(t) = d_0^*$: (6.5.3) y (6.5.4) coinciden. Para el análisis, es conveniente introducir una formulación del error. Se define el error de salida: $e_0 = y_p - y_m \quad (6.5.9)$

y el parámetro del error:

$$\phi = \begin{bmatrix} \phi_r(t) \\ \phi_y(t) \end{bmatrix} = \begin{bmatrix} c_0(t) - c_0^* \\ d_0(t) - d_0^* \end{bmatrix} \quad (6.5.10)$$

Sustrayendo (6.5.4) de (6.5.7):

$$\begin{aligned} \dot{e}_0 &= -a_m (y_p - y_m) + (a_m - a_p + k_p d_0) y_p + k_p c_0 r - k_m r \\ \dot{e}_0 &= -a_m e_0 + k_p [(c_0 - c_0^*)r + (d_0 - d_0^*)y_p] \\ \dot{e}_0 &= -a_m e_0 + k_p (\phi_r r + \phi_y y_p) \end{aligned} \quad (6.5.11)$$

Se puede representar (6.5.11) en una forma compacta de la siguiente manera:

$$\begin{aligned} e_0 &= \frac{k_p}{s + a_m} (\phi_r r + \phi_y y_p) \\ e_0 &= \frac{k_p}{k_m} \hat{M} (\phi_r r + \phi_y y_p) = \frac{1}{c_0^*} \hat{M} (\phi_r r + \phi_y y_p) \end{aligned} \quad (6.5.12)$$

La ecuación (6.5.12) es de la forma de la ecuación del error real estrictamente positiva. Entonces, tentativamente se eligen las leyes de actualización:

$$\begin{aligned} \dot{c}_0 &= -g e_0 r \\ \dot{d}_0 &= -g e_0 y_p \end{aligned} \quad g > 0 \quad (6.5.13)$$

Asumiendo que $\frac{k_p}{k_m} > 0$ y que \hat{M} es real estrictamente positiva.

6.6 ³⁷Control Adaptivo por Modelo de Referencia

Este punto está basado en una traducción libre del texto de S. Sastry y M. Bodson: “*Adaptive Control- Stability, Convergence, and Robustness*” (Capítulo 3), publicado en 1989.

6.6.1 Suposiciones

(A1) Suposiciones de la planta

La planta es SISO y LTI, descrita por la función de transferencia

$$\frac{\hat{y}_p(s)}{\hat{u}(s)} = \hat{P}(s) = k_p \frac{\hat{n}_p(s)}{\hat{d}_p(s)} \quad (6.6.1.1)$$

Donde $\hat{n}_p(s)$, $\hat{d}_p(s)$ son polinomios mónicos y coprimos de grado ‘m’ y ‘n’ respectivamente. La planta es estrictamente propia ($n > m$) y de fase mínima. El signo de la ganancia de alta frecuencia k_p es conocido y sin pérdida de generalidad se asume $k_p > 0$.

(A2) Suposiciones del modelo de referencia

El modelo de referencia se describe por

$$\frac{\hat{y}_m(s)}{\hat{r}(s)} = \hat{M}(s) = k_m \frac{\hat{n}_m(s)}{\hat{d}_m(s)} \quad (6.6.1.2)$$

Donde $\hat{n}_m(s)$, $\hat{d}_m(s)$ son polinomios mónicos y coprimos de grado ‘m’ y ‘n’ respectivamente (los mismos grados correspondientes a los polinomios de la planta). El modelo de referencia es estable de fase mínima y $k_m > 0$.

(A3) Suposiciones de la entrada de referencia

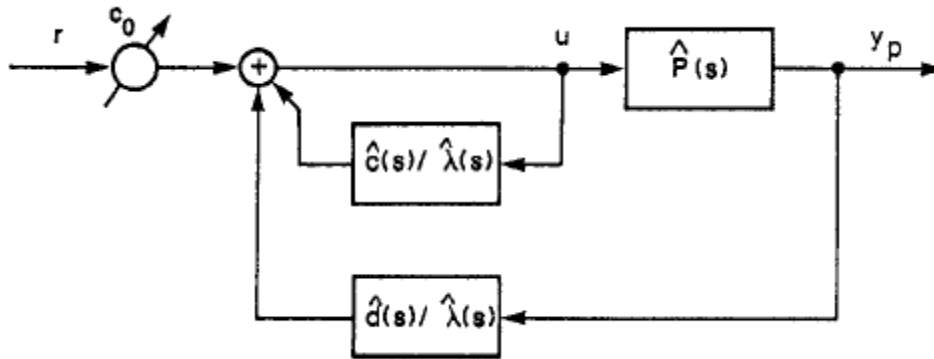
La entrada de referencia $r(\cdot)$ es lineal y continua a tramos.

Notar que $\hat{P}(s)$ se asume de fase mínima, pero NO necesariamente estable.

³⁷ (Sastry & Bodson, 1989)

6.6.2 ³⁸Estructura del Controlador

Para alcanzar el objetivo de control, se considera la estructura de control mostrada en la figura a continuación:



Estructura de control

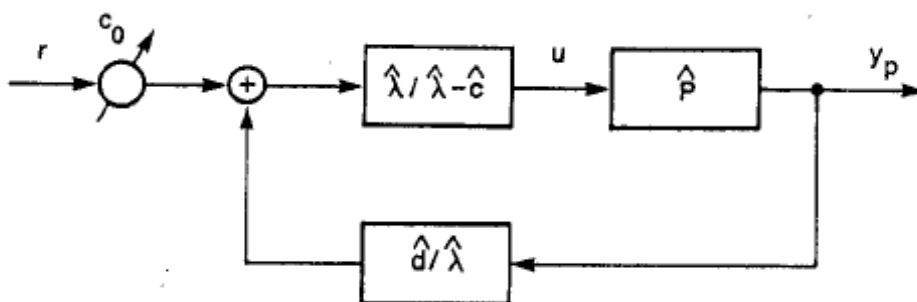
Inspeccionando la figura se ve que:

$$u = c_0 r + \frac{\hat{c}(s)}{\hat{\lambda}(s)}(u) + \frac{\hat{d}(s)}{\hat{\lambda}(s)}(y_p) \quad (6.6.2.1)$$

Donde c_0 es un escalar, $\hat{c}(s)$, $\hat{d}(s)$ y $\hat{\lambda}(s)$ son polinomios de grados 'n-2', 'n-1' y 'n-1' respectivamente.

$$\text{De (6.7.1):} \quad u = \frac{\hat{\lambda}}{\hat{\lambda} - \hat{c}} \left(c_0 r + \frac{\hat{d}}{\hat{\lambda}}(y_p) \right) \quad (6.6.2.2)$$

Esto se refleja en la siguiente figura:



Forma equivalente de la estructura de control.

(Se demuestra, ver Sastry págs.: 105-107, que esta estructura de controlador es adecuada para lograr el control deseado)

³⁸ (Sastry & Bodson, 1989)

6.6.3 ³⁹Representación en el espacio de estados

Consideramos la representación en el espacio de estados del controlador. Se elige $\Lambda \in \mathfrak{R}^{n-1, n-1}$ y $b_\lambda \in \mathfrak{R}^{n-1}$, de tal manera que (Λ, b_λ) está en la forma controlable canónica y $\det(sI - \Lambda) = \hat{\lambda}(s)$. Entonces:

$$(sI - \Lambda)^{-1} b_\lambda = \frac{1}{\hat{\lambda}(s)} \begin{bmatrix} 1 \\ s \\ \cdot \\ \cdot \\ s^{n-2} \end{bmatrix} \quad (6.6.3.1)$$

Sea $c \in \mathfrak{R}^{n-1}$ el vector de coeficientes del polinomio $\hat{c}(s)$, tal que:

$$\frac{\hat{c}(s)}{\hat{\lambda}(s)} = c^T (sI - \Lambda)^{-1} b_\lambda \quad (6.6.3.2)$$

Consecuentemente, esta función de transferencia puede realizarse por:

$$\begin{aligned} \dot{w}^{(1)} &= \Lambda w^{(1)} + b_\lambda u \\ \frac{\hat{c}}{\hat{\lambda}}(u) &= c^T w^{(1)} \end{aligned} \quad (6.6.3.3)$$

Donde el estado $w^{(1)} \in \mathfrak{R}^{n-1}$ y la condición inicial $w^{(1)}(0)$ son arbitrarios.

Similarmente, existe $d_0 \in \mathfrak{R}$ y $d \in \mathfrak{R}^{n-1}$, tal que:

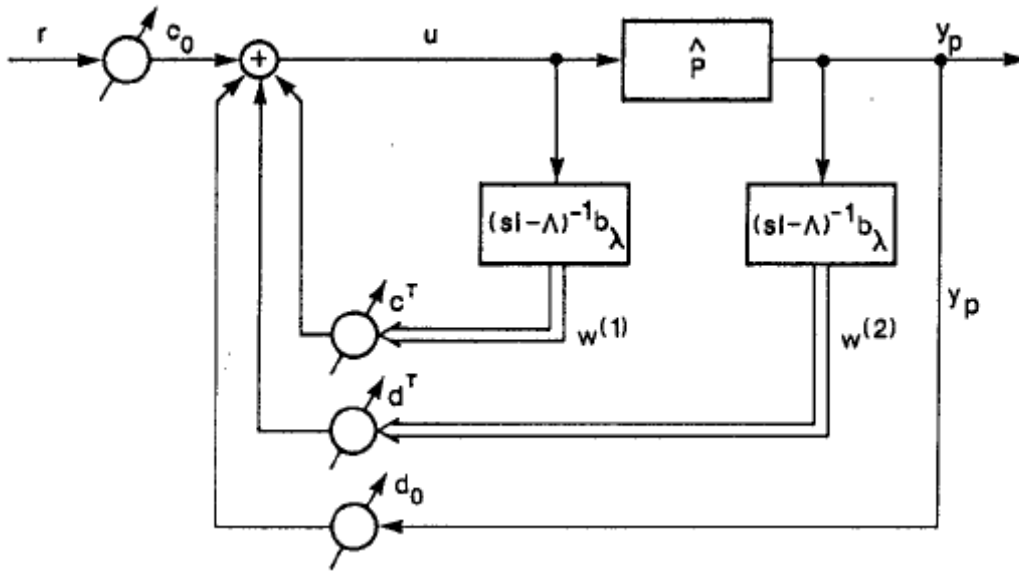
$$\frac{\hat{d}(s)}{\hat{\lambda}(s)} = d_0 + d^T (sI - \Lambda)^{-1} b_\lambda \quad (6.6.3.4)$$

y

$$\begin{aligned} \dot{w}^{(2)} &= \Lambda w^{(2)} + b_\lambda y_p \\ \frac{\hat{d}}{\hat{\lambda}}(y_p) &= d_0 y_p + d^T w^{(2)} \end{aligned} \quad (6.6.3.5)$$

Donde el estado $w^{(2)} \in \mathfrak{R}^{n-1}$ y la condición inicial $w^{(2)}(0)$ son arbitrarios. El controlador puede ser representado de la siguiente manera:

³⁹ (Sastry & Bodson, 1989)



Estructura de control, forma adaptiva.

Con:

$$u = c_0 r + c^T w^{(1)} + d_0 y_p + d^T w^{(2)} \quad (6.6.3.8)$$

$$u := \theta^T w$$

Donde:

$\theta^T := (c_0, \bar{\theta}^T) := (c_0, c^T, d_0, d^T) \in \mathfrak{R}^{2n}$ es el vector de parámetros del controlador
 $w^T := (r, \bar{w}^T) := (r, w^{(1)T}, y_p, w^{(2)T}) \in \mathfrak{R}^{2n}$ es un vector de señales que puede ser obtenido sin conocimiento de los parámetros de la planta.

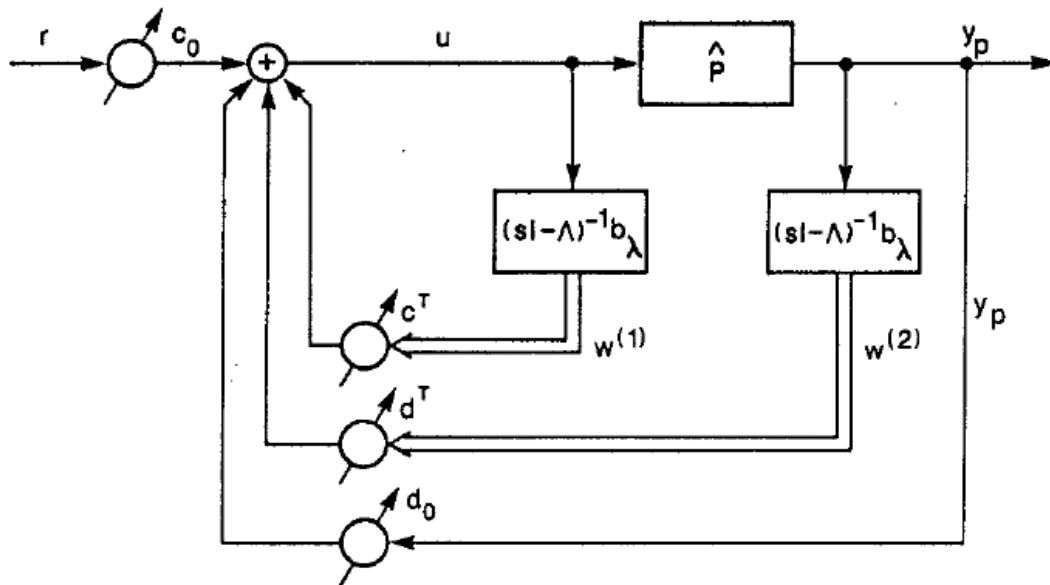
Hasta aquí se desarrollo la estructura de control a implementar, para posteriormente comenzar el desarrollo de los algoritmos de control.

6.7 Modelo de Referencia (MR) con error a la salida

6.7.1 MR con error a la salida y grado relativo 1

Este punto está basado en una traducción libre del texto de S. Sastry y M. Bodson: “*Adaptive Control- Stability, Convergence, and Robustness*” (Capítulo 3), publicado en 1989.

La estructura de controlador propuesta en el libro anteriormente mencionado es:



- **Suposiciones:** A1, A2, A3 y A6

(A1) Suposiciones de la planta

La planta es SISO y LTI, descrita por la función de transferencia

$$\frac{\hat{y}_p(s)}{\hat{u}(s)} = \hat{P}(s) = k_p \frac{\hat{n}_p(s)}{\hat{d}_p(s)} \quad (6.7.1.1)$$

Donde $\hat{n}_p(s)$, $\hat{d}_p(s)$ son polinomios mónicos y coprimos de grado ‘m’ y ‘n’ respectivamente. La planta es estrictamente propia ($n > m$) y de fase mínima. El signo de la ganancia de alta frecuencia k_p es conocido y sin pérdida de generalidad asumiremos $k_p > 0$.

(A2) Suposiciones del modelo de referencia

El modelo de referencia se describe por

$$\frac{\hat{y}_m(s)}{\hat{r}(s)} = \hat{M}(s) = k_m \frac{\hat{n}_m(s)}{\hat{d}_m(s)} \quad (6.7.1.2)$$

Donde $\hat{n}_m(s)$, $\hat{d}_m(s)$ son polinomios mónicos y coprimos de grado ‘m’ y ‘n’ respectivamente (los mismos grados correspondientes a los polinomios de la planta). El modelo de referencia es estable de fase mínima y $k_m > 0$.

(A3) Suposiciones de la entrada de referencia

La entrada de referencia $r(\cdot)$ es lineal y continua a tramos.

Notar que $\hat{P}(s)$ se asume de fase mínima, pero NO necesariamente estable.

(A6) Grado relativo = 1 $\Rightarrow n-m = 1$

- Datos: n, k_p

- Entrada

$$r(t), y_p(t) \in \mathfrak{R}$$

- Salida

$$u(t) \in \mathfrak{R}$$

- Señales internas

$$w(t) \in \mathfrak{R}^{2n} [w^{(1)}(t), w^{(2)}(t) \in \mathfrak{R}^{n-1}]$$

$$\theta(t) \in \mathfrak{R}^{2n} [c_0(t), d_0(t) \in \mathfrak{R}, d(t) \in \mathfrak{R}^{n-1}]$$

$$y_m(t), e_0(t) \in \mathfrak{R}$$

- Las condiciones iniciales son arbitrarias.

Parámetros de diseño:

$$* \hat{M}(k_m, \hat{n}_m, \hat{d}_m) \text{ satisface A2 y A6}$$

$$* \Lambda \in \mathfrak{R}^{n-1 \times n-1}, b_\lambda \in \mathfrak{R}^{n-1} \text{ es controlable de forma canónica tal que}$$

$$\det(sI - \Lambda) = \hat{n}_m(s)$$

$$* g > 0$$

Estructura de control:

$$\dot{w}^{(1)} = \Lambda w^{(1)} + b_\lambda u$$

$$\dot{w}^{(2)} = \Lambda w^{(2)} + b_\lambda y_p$$

$$\theta^T = (c_0, c^T, d_0, d^T)$$

$$w^T = (r, w^{(1)T}, y_p, w^{(2)T})$$

$$u = \theta^T w$$

Estructura del identificador:

$$y_m = \hat{M}(r)$$

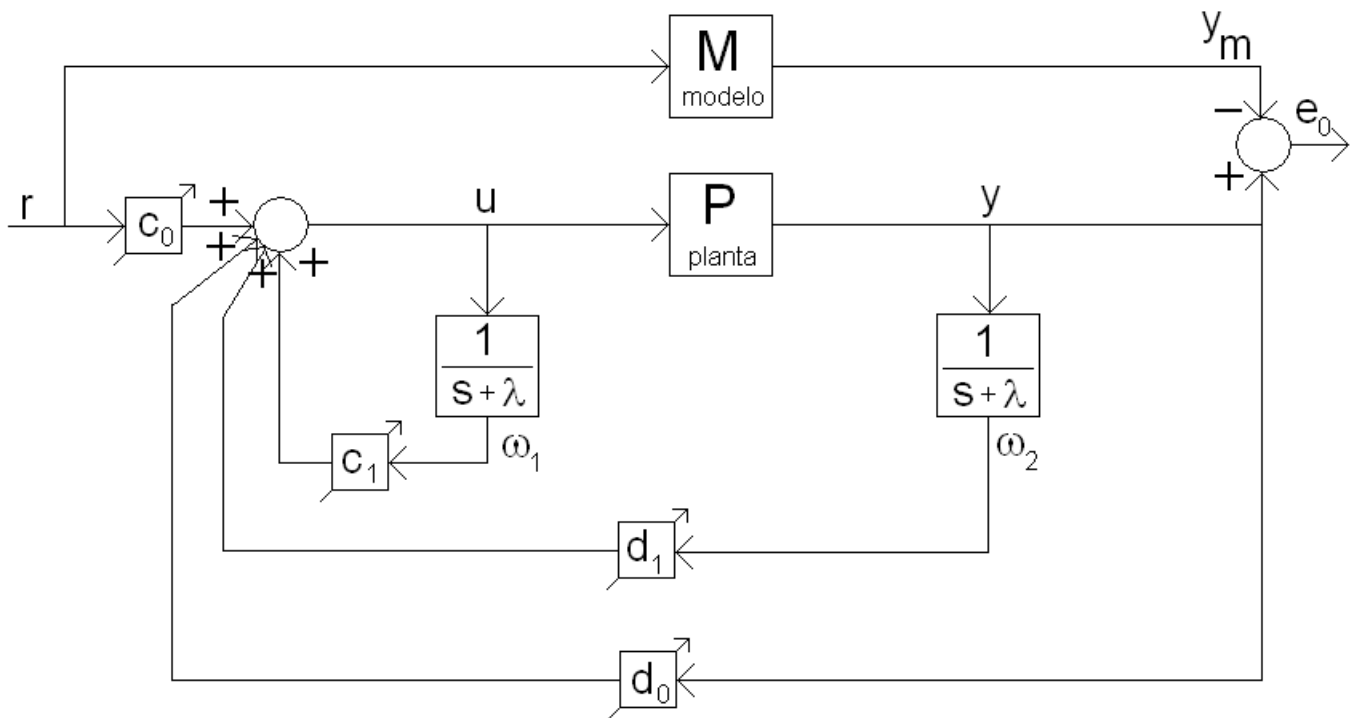
$$e_0 = y_p - y_m$$

Algoritmo gradiente

$$\boxed{\dot{\theta} = -g e_0 w}$$

6.7.2 Ecuaciones del Controlador

La estructura del modelo de referencia de error a la salida es la siguiente:



Con:

$$e_0 = y - rM$$

$$\dot{\theta} = -ge_0 w \text{ (Algoritmo del Gradiente)}$$

Donde:

$$\theta = [c_0, c_1, d_0, d_1]$$

$$w = [r, w_1, y, w_2]$$

Ahora se buscará llegar a expresar todas las señales que intervienen en el control en forma de ecuaciones diferenciales de primer orden o en ecuaciones lineales, según sea el caso.

De la figura anterior se verifica que: $u = \theta^T w = c_0 r + c_1 w_1 + d_1 w_2 + d_0 r$ (i)

El algoritmo del gradiente impone las siguientes leyes de actualización, que ya están en el formato deseado:

$$\begin{bmatrix} \dot{c}_0 \\ \dot{c}_1 \\ \dot{d}_0 \\ \dot{d}_1 \end{bmatrix} = -ge_0 \begin{bmatrix} r \\ w_1 \\ y \\ w_2 \end{bmatrix} \begin{matrix} (ii) \\ (iii) \\ (iv) \\ (iv) \end{matrix}$$

Si la transferencia de la planta modelo es:

$$\frac{y_m}{r} = \frac{as + b}{s^2 + cs + d}$$

Operando en el tiempo se obtiene la siguiente igualdad:

$$ar + br = \ddot{y}_m + c\dot{y}_m + dy_m \Rightarrow \underbrace{\ddot{y}_m - ar}_{\dot{x}_2} = -c\dot{y}_m - dy_m + br \quad (*)$$

Luego se realizan los siguientes cambios de variable:

$$1) x_2 = \dot{y}_m - ar$$

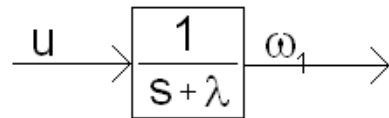
$$2) x_1 = y_m$$

Entonces de (*) se concluye: $\dot{x}_2 = -cx_2 - acr - x_1d + br$

$$\Rightarrow \dot{x}_2 = -x_2c - x_1d + (b - ac)r$$

De 1) y 2) se tiene que: $x_2 = \dot{x}_1 - ar \Rightarrow \dot{x}_1 = x_2 + ar$

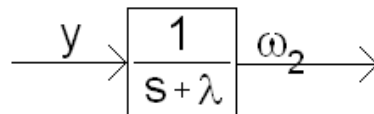
Para los bloques “identificadores”:



$$w_1(s + \lambda) = u \Rightarrow \dot{w}_1 + w_1\lambda = u$$

$$\Rightarrow \dot{w}_1 + w_1\lambda = u$$

$$\Rightarrow \dot{w}_1 = u - w_1\lambda \quad (v)$$

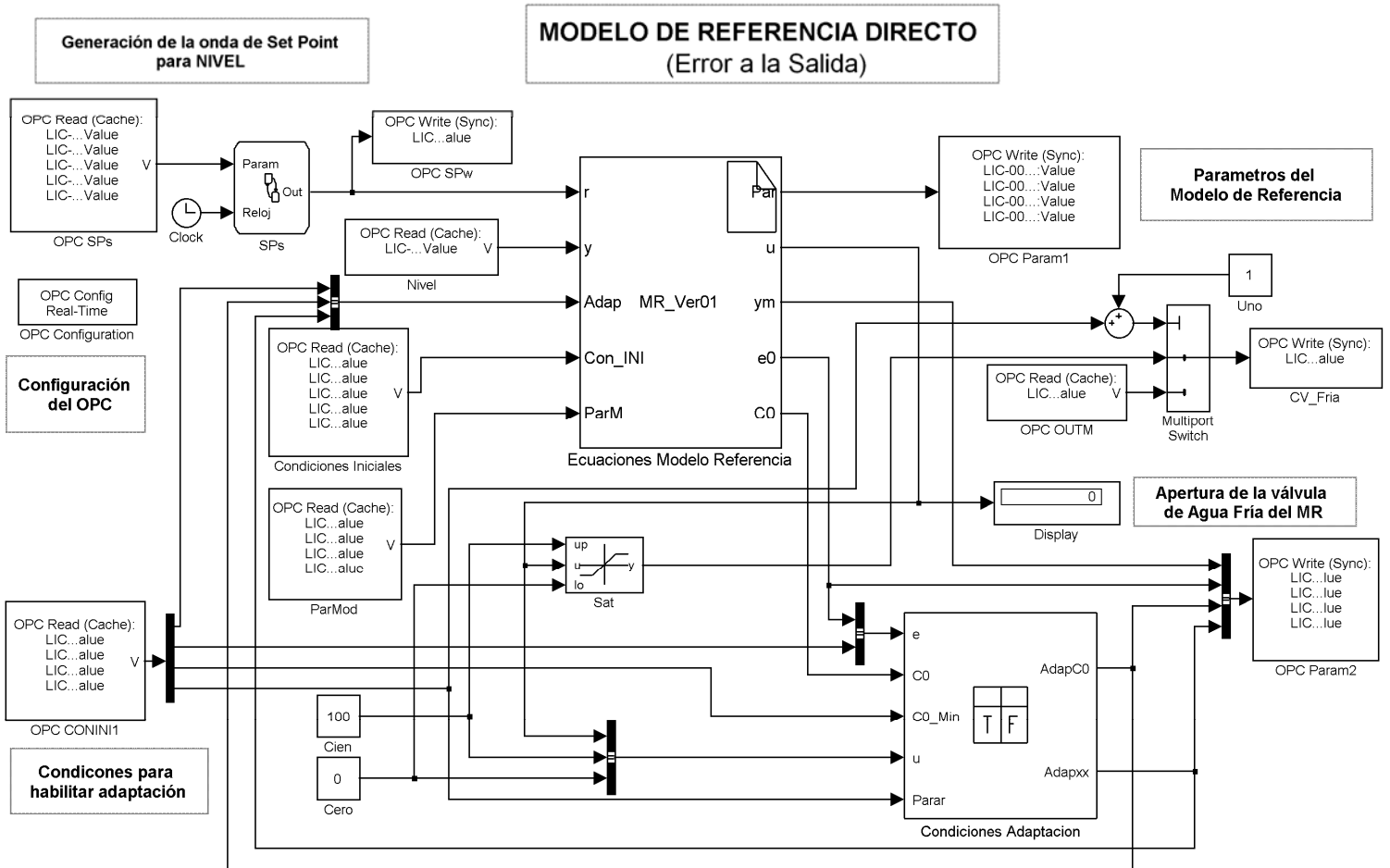


$$\text{De manera análoga: } \dot{w}_2 = y - w_2\lambda \quad (vi)$$

Las seis ecuaciones remarcadas son las que se implementaran en el bloque central del MR, que es el encargado de resolver las ecuaciones diferenciales de 1º orden y de generar las salidas deseadas.

6.7.3 Implementación Matlab

El diagrama utilizado para la simulación y posterior testeo de esta versión de modelo de referencia es el siguiente:



El corazón del controlador es una S-Function utilizada (bloque 'MR_Ver01') para resolver las ecuaciones diferenciales (i)...(vi).

Los bloques OPC se encargan de la comunicación con el DCS, los OPC Read se encargan de leer vía OPC los datos desde el DCS, mientras que los OPC Write se encargan de escribir datos al DCS.

A la salida de a la válvula de agua fría se agregó un bloque de "saturación lineal" de 0-100 para evitar dañar al servo que la comandan (bloque: 'Sat').

La tabla de verdad (bloque 'Condiciones Adaptación') se encarga de decidir si se deben, o no, actualizar los parámetros del MR según criterios que se detallarán a continuación.

Finalmente, el bloque 'Multiport Switch' permite seleccionar, desde pantalla del DCS, si el control de la válvula es realizado por el MR o de forma manual.

Implementación en S-Function: bloque 'MR-Ver01'

Salidas

```
/* C0 = Par[0]*/
   Par[0] = xC[2] + Con_INI[0];
   C0[0] = Par[0];
/* C1 = Par[1]*/
   Par[1] = xC[3] + Con_INI[1];
/* D1 = Par[2]*/
   Par[2] = xC[4] + Con_INI[2];
/* D0 = Par[3]*/
   Par[3] = xC[5] + Con_INI[3];
/* u = r*C0 + w1*C1 + w2*D1 + D0*y */
   u[0] = r[0]*Par[0] + xC[0]*Par[1] + xC[1]*Par[2] + y[0]*Par[3];
/* La planta a simular es de la siguiente forma
   Par[0]*S + Par[1]
   -----
   S2 + Par[2]*S + Par[3]*/
ym[0] = xC[6];
e0[0] = ym[0] - y[0]; (e0 se definió con signo opuesto para que las leyes de actualización
sean con signo positivo)
```

Ecuaciones Diferenciales

```
/* W1 = xC[0]*/
   dx[0] = u[0] - xC[0]*Con_INI[4];
/* W2 = xC[1]*/
   dx[1] = y[0] - xC[1]*Con_INI[4];
/* Actualización C0 = xC[2]*/
   dx[2] = r[0]*Adap[0]*e0[0]*Adap[1]*Adap[2];
/* Actualización C1 = xC[3]*/
   dx[3] = xC[0]*Adap[0]*e0[0]*Adap[2];
/* Actualización D1 = xC[4]*/
   dx[4] = xC[1]*Adap[0]*e0[0]*Adap[2];
/* Actualización D0 = xC[5]*/
   dx[5] = y[0]*Adap[0]*e0[0]*Adap[2];
/* La planta a simular es de la siguiente forma
   Par[0]*S + Par[1]
   -----
   S2 + Par[2]*S + Par[3]*/
dx[6] = xC[7] + ParM[0]*r[0]; ( $dx[6] = \dot{x}_1 = \dot{y}$ )
dx[7] = -xC[7]*ParM[2] - xC[6]*ParM[3] + (ParM[1] - ParM[0]*ParM[2])*r[0]; ( $dx[7] = \dot{x}_2$ )
```

Donde:

- Adap[0]: 'g' del método del gradiente
- Adap[1]: habilita a adaptar C0
- Adap[2]: habilita adaptación general
- Con_INI[4]: lambda del MR

Bloque: 'Condiciones Adaptación'

Este bloque es una Truth-Table (Tabla de Verdad) e implementa una función lógica. Para la misma se definen dos tablas, una de condiciones y otra de acciones. Si se cumple una condición se ejecuta entonces la acción correspondiente.

Tabla de condiciones:

#	Descripción	Condición	1	2	3	4	5
1	Abs(e0) < eMin \Rightarrow no se precisa actualizar parámetros	abs(e(1)) < e(2)	T	-	-	-	F
2	Parar actualización por saturación de 'u'	(u(1) >= 100) (u(1) <= 0)	-	T	-	-	F
3	Parar actualizaciones	Parar == 1	-	-	T	-	F
4	Parar actualización de c0	(C0 <= C0_Min) && (e(1) < 0	-	-	-	T	F
		Acciones:	1	1	1	2	3

Tabla de acciones:

#	Descripción	Acción
1	Parar la actualización de todos los parámetros	AdapC0 = 0; Adapxx = 0;
2	Parar actualización de c0	AdapC0 = 0; Adapxx = 1;
3	Actualizar parámetros	AdapC0 = 1; Adapxx = 1;

Por ejemplo, supongamos que se fuerza al MR a parar las actualizaciones de los parámetros (para esto se requiere que 'Parar' sea igual a 1). Si se cumple la condición respectiva se ejecuta la acción de la misma línea que tenga una T (True), en éste caso la 1, por lo que luego, si se observa la tabla de acciones, esto genera que las dos salidas del bloque sean 0. En el caso contrario, si Parar=0, como la condición es falsa, si no hay otra condición que sea verdadera entonces se ejecuta la acción de la F, que siguiendo con el ejemplo sería la 3, por lo que el bloque sacaría en sus dos salidas un 1.

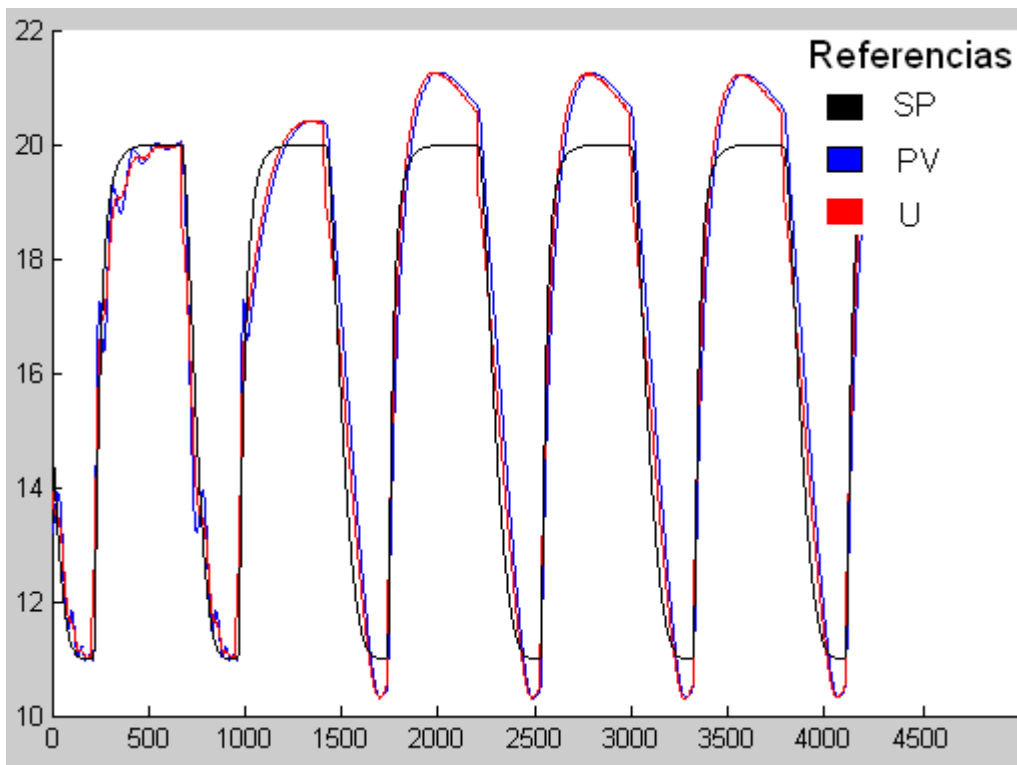
6.7.4 Simulación

Para simular el controlador se utilizó la planta modelo: $P_M(s) = \frac{s+1/4}{s^2+s+1/4}$, la planta

$P(s) = \frac{0.02s+1.36}{s^2+3.02s+1.24}$ (similar a una obtenida identificando la transferencia de nivel), un

valor para 'g' (constante del algoritmo del gradiente) de 0.005, $\lambda = 4$ y condiciones iniciales $c_0 = 1$, $c_1 = 0$, $d_1 = -1$ y $d_0 = 1$.

El SP que ingresa a la planta y las señales de control y valor del proceso son las detalladas en la siguiente figura:

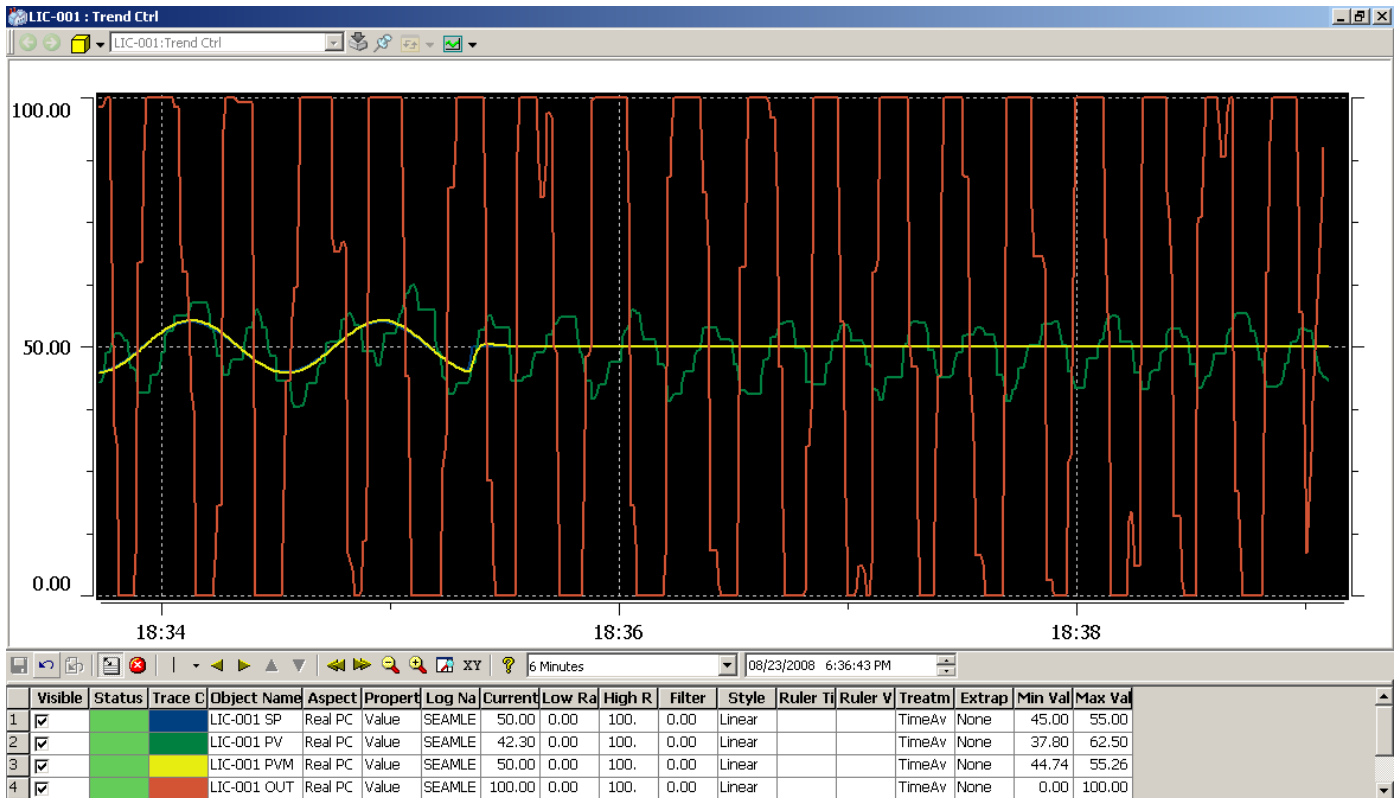


6.7.5 Prueba Física

Para testear el controlador se utilizó la planta modelo: $P_M(s) = \frac{s+1/4}{s^2+s+1/4}$, un valor para 'g' (constante del algoritmo del gradiente) de 0.005, $\lambda = 4$ y condiciones iniciales $c_0 = 1$, $c_1 = 0$, $d_1 = -1$ y $d_0 = 1$. Además se lo utilizó como controlador de nivel.

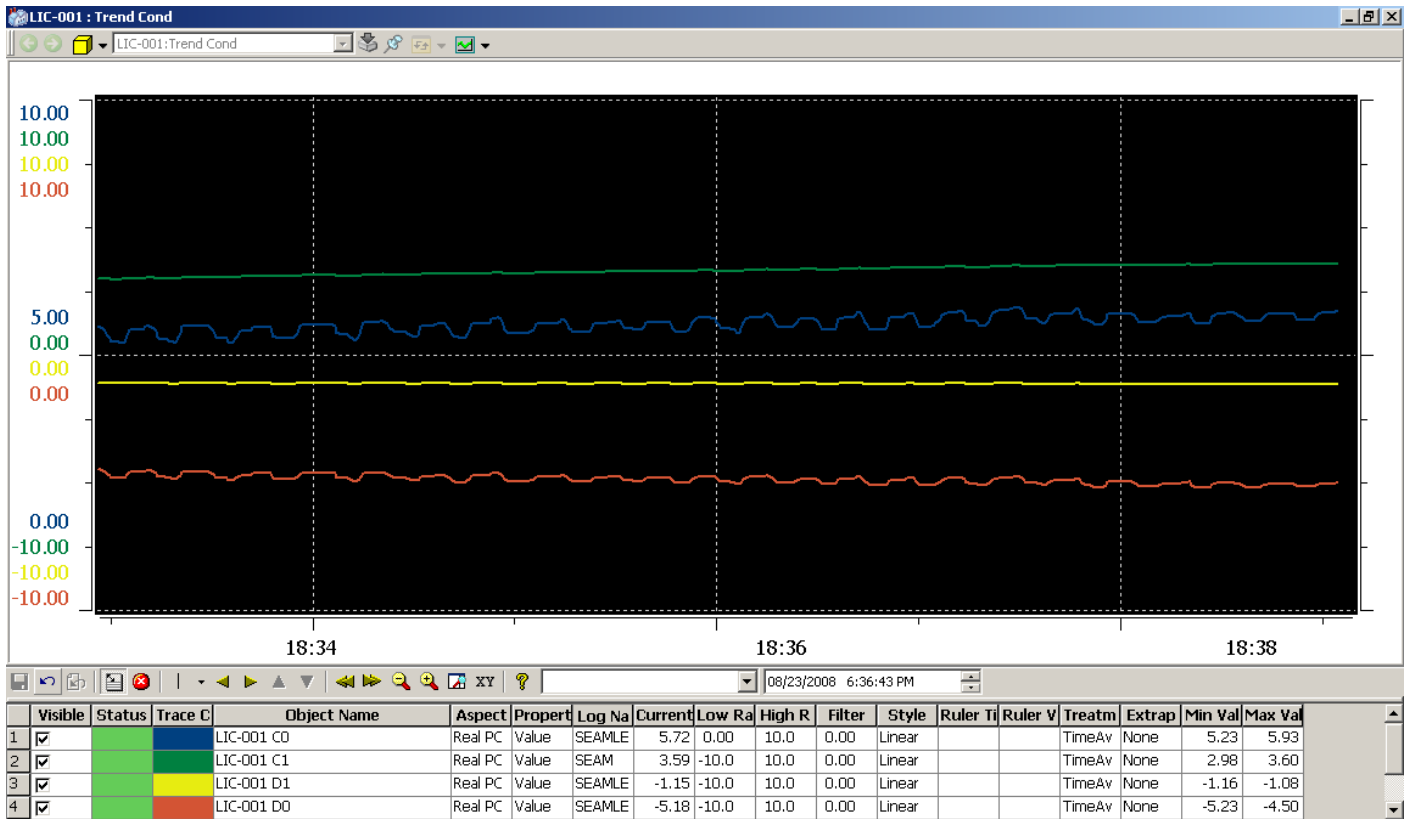
Mediante un seno de baja amplitud y alto período se dejó al MR adaptar sus parámetros, luego de un tiempo prolongado se fijó el SP en el 50% y se relevó su respuesta.

En la siguiente figura se muestra la evolución de las señales de Control:



De la gráfica se observa que la señal de control a la válvula es similar a la de un controlador ON-OFF. El controlador, de forma prácticamente instantánea, cierra al 0% y abre al 100% la válvula de agua fría. Esto genera una oscilación alrededor del SP (en ésta gráfica el SP coincide con el PVM que es el PV del modelo). Esto es de esperar si se observa cómo fue con anterioridad la respuesta del controlador cuando el SP era una senoide.

En la siguiente figura se presenta la evolución en el tiempo de los diferentes parámetros que juegan en este controlador:



Con ésta gráfica se termina de confirmar el mal desempeño de este controlador, pues como puede observarse los parámetros evolucionan, algunos de forma más estable, otros con oscilaciones alrededor de una recta, sin converger a un juego de parámetros que optimice el rendimiento del controlador.

Para otras pruebas con diferente elección de parámetros y SPs se llegan a resultados similares y peores.

Finalmente se concluye que este controlador es poco robusto, si bien se puede decir que sirve para hacer un “control” elemental, su comportamiento no es el esperado.

6.7.6 Interpretación de los parámetros del control MR

Observando las ecuaciones que rigen el modelo de referencia con error a la salida, la adaptación de los parámetros está determinada por las siguientes ecuaciones:

$$\begin{bmatrix} \dot{c}_0 \\ \dot{c}_1 \\ \dot{d}_0 \\ \dot{d}_1 \end{bmatrix} = -ge_0 \begin{bmatrix} r \\ w_1 \\ y \\ w_2 \end{bmatrix}$$

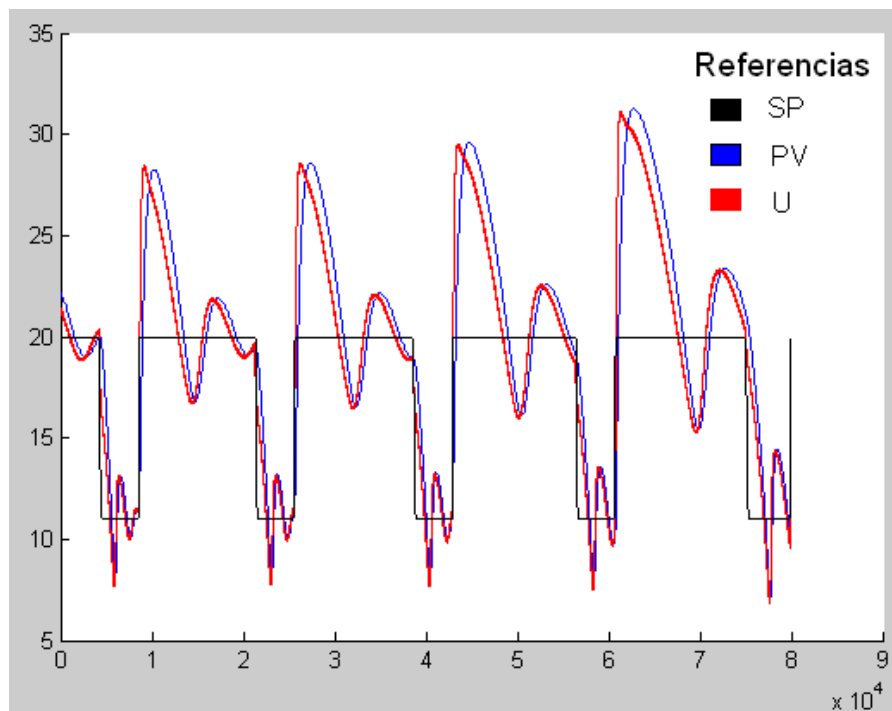
Ganancia del método del gradiente:

El término ‘g’ es una variable que permite variar los parámetros del modelo de referencia de acuerdo a las señales de error y a las internas del modelo.

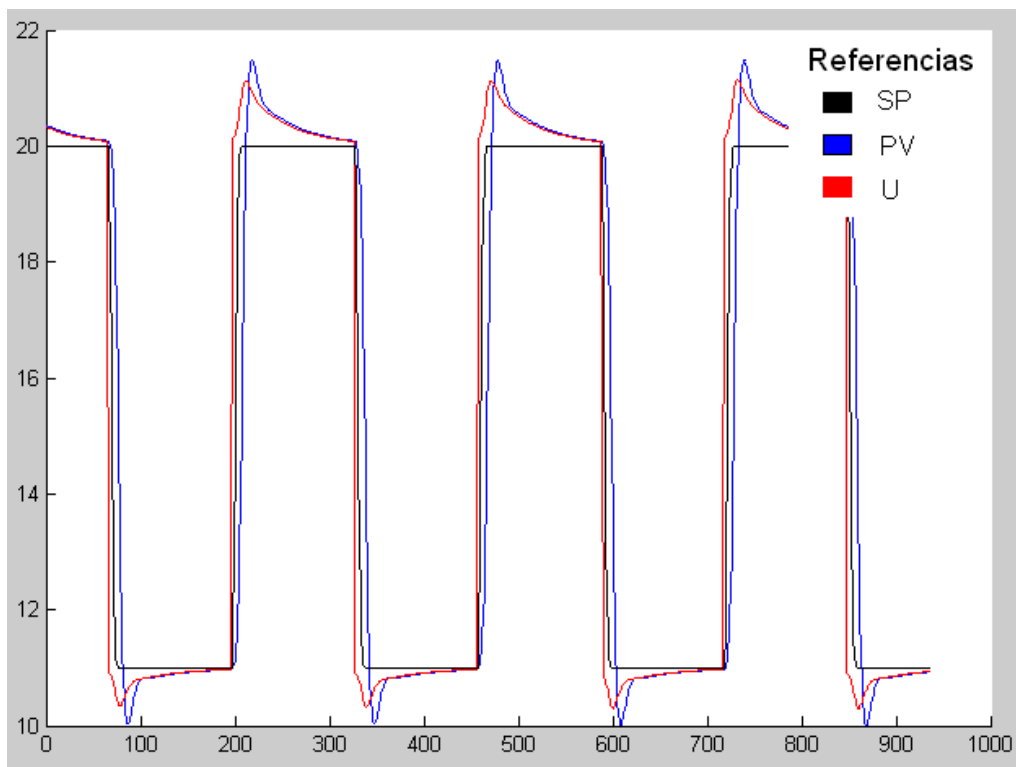
Este parámetro juega un rol muy importante, ya que si se lo elige muy grande, es muy probable que el algoritmo del modelo de referencia nunca converja, en cambio si se lo elige muy chico, es probable que la convergencia se dé lentamente y además se acentúen las posibilidades de encontrar un punto de equilibrio que no sea óptimo.

De acuerdo a las simulaciones realizadas, para poder tener una idea de cómo elegir estos parámetros, las cuales no fueron muy minuciosas ni exigentes ya que no hay bibliografía de cómo elegirlos, estas arrojaron que con un ‘g’ en el rango de 1/50 y 1/100 de la ganancia de la planta es donde mejor se comporta el modelo de referencia.

Si 'g' es grande el MR diverge, como puede apreciarse en la siguiente figura:



Si 'g' se encuentra en el rango: $1/50$ y $1/100$ de la ganancia de la planta y los demás parámetros son "buenos", se llega al siguiente control por MR:



El parámetro lambda: $\left(\frac{1}{s + \lambda}\right)$

Observando la bibliografía⁴⁰ este parámetro fija en cuánto tiempo el modelo de la planta va a converger desde que se inicia la simulación. O sea que con esto sería lógico elegir el parámetro lambda lo más grande posible, pero en las simulaciones realizadas se ve que existe un lambda óptimo.

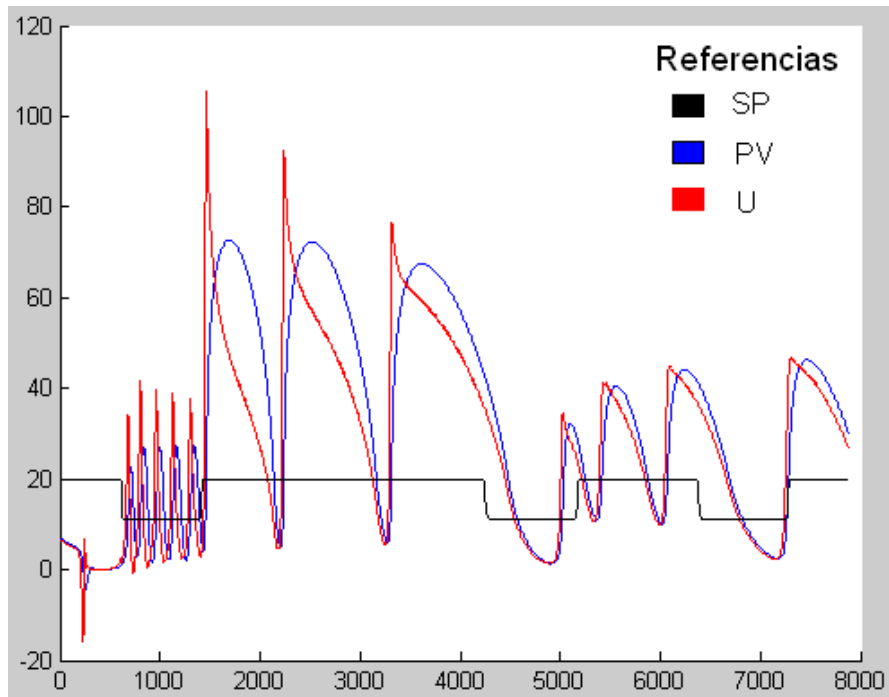
Si se elige lambda muy grande, se estarían dejando pasar mayores frecuencias (filtro pasabajos) y achicando la ganancia dentro del rango de interés, por el contrario si se disminuye el valor de lambda, se están filtrando las altas frecuencias y dándole mayor ganancia al pasabajos.

En la realidad las frecuencias altas (para procesos industriales lentos) se relacionan con el ruido, por lo que elegir valores altos para lambda haría que se tome en cuenta el ruido en mayor proporción. Por otro lado si se elige lambda bajo, se filtraría inclusive lo que se desea modelar.

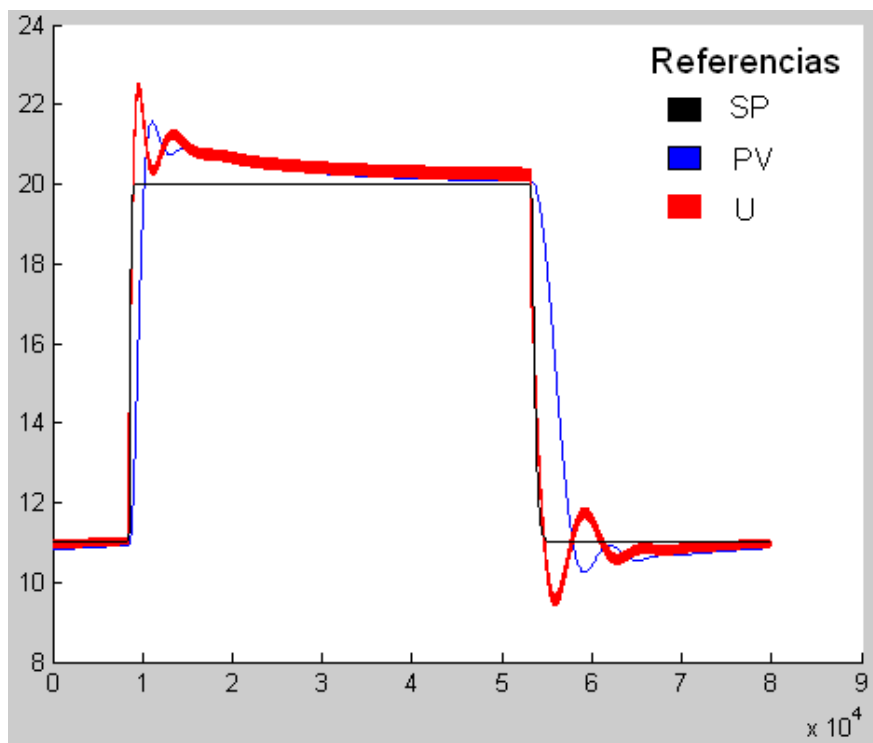
Para este parámetro también se realizaron simulaciones y se llegó a la conclusión que la elección ideal se encuentra entre 3 y 6 veces el módulo del polo más cercano a cero.

⁴⁰ (Sastry & Bodson, 1989)

Si lambda es muy grande en las simulaciones se obtiene lo siguiente:



En cambio si lambda es chico:

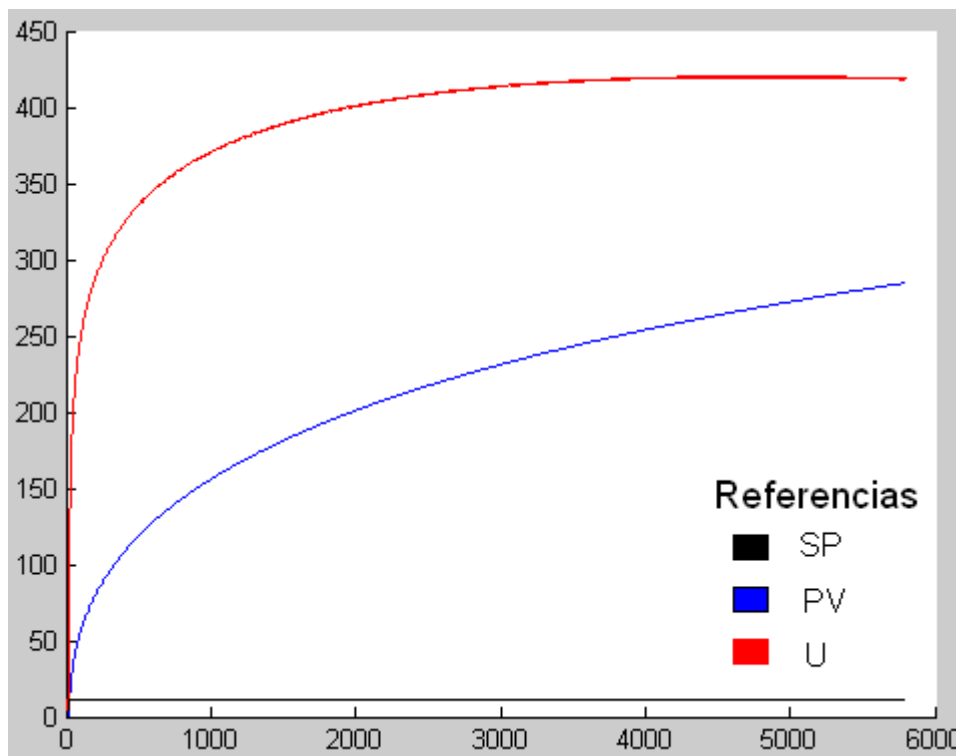


Condiciones iniciales

Como este algoritmo es recursivo, las condiciones iniciales juegan un papel muy importante, ya que si se eligen muy cercanas a un punto de equilibrio los parámetros del modelo convergen rápidamente. En cambio si se eligen lejos del punto de equilibrio óptimo, se pueden encontrar otros puntos de equilibrio, en estos el algoritmo puede funcionar bien, pero al hacer cambios en las señales de referencia las constantes cambian nuevamente buscando otro punto de equilibrio.

Para el algoritmo desarrollado se notó que d_0 siempre tiende a tomar valores negativos y c_0 valores positivos, si se dan condiciones iniciales de signo distinto al que por naturaleza d_0 y c_0 tienen, el algoritmo diverge. Esto queda claro si se piensa al parámetro d_0 del algoritmo como una realimentación, si la misma es positiva, seguramente el control diverge.

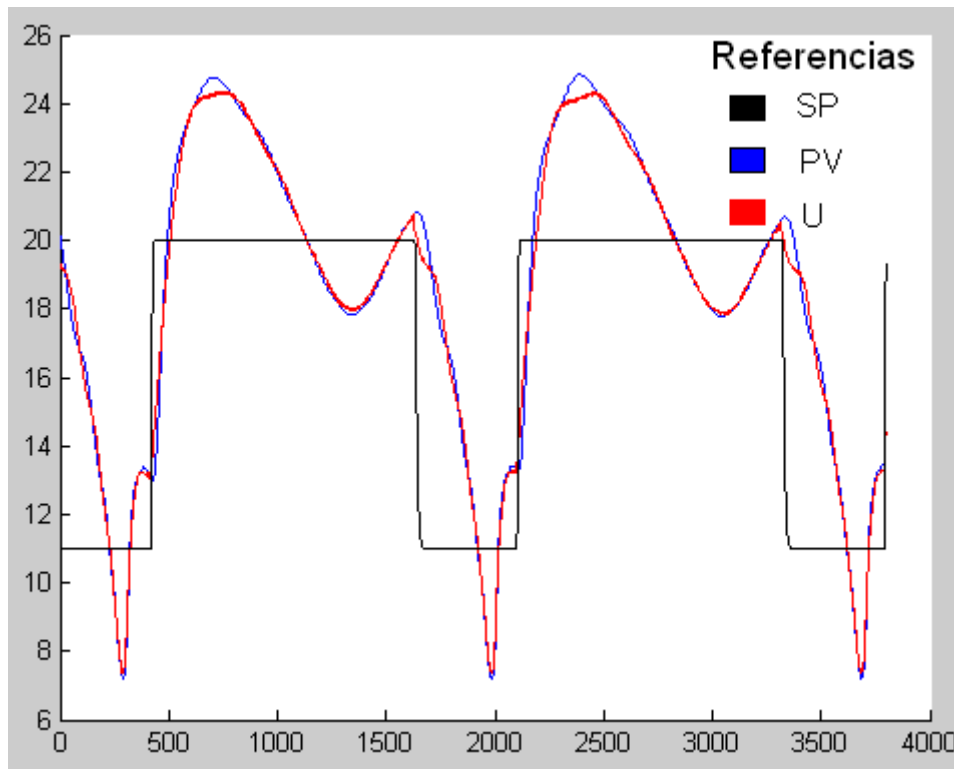
Un ejemplo de control MR con condiciones iniciales muy malas es el siguiente:



Otra observación interesante es que el algoritmo funciona muy bien cuando la onda a seguir tiene media cero. La explicación que se encontró es la siguiente: cuando se trabaja con señales de media cero los errores de bajada y subida del algoritmo suman o restan por igual, cuando se sale de media cero los errores en la subida son positivos (o negativos) y en la bajada son negativos o (positivos) haciendo que oscilen los parámetros de control. Esto es correcto si se piensa que esta teoría está desarrollada para sistemas lineales.

Otro punto que no se ha tomado en cuenta hasta ahora es que esta teoría es para plantas estables (incluyendo plantas con polos en el eje imaginario). De las simulaciones realizadas se llega a la conclusión que las plantas con polos en el eje imaginario, no deben funcionar como la bibliografía lo indica, ya que las mismas son muy sensibles a las condiciones iniciales.

Para una planta con dos polos en el eje imaginario:



Para terminar el análisis, se encontró que el MR cuando debe seguir una referencia fija y el PV tiene un poco de ruido, este comienza a divergir de a poco, en un principio se pensó que había algún error en el algoritmo implementado. Después de revisar el algoritmo y de releer cuidadosamente la bibliografía consultada no se pudo encontrar una explicación a este hecho. Esto fue así hasta que se consultaron libros más prácticos sobre el tema y se encontró que este problema está reportado y es tema de estudio, sugiriéndose no adaptar los parámetros en una franja de error mínima, lo cual fue posteriormente implementado.

6.7.7 Conclusiones

En el inicio de este proyecto, cuando se quiso controlar el nivel con el modelo de referencia, se encontraron las dificultades que se describieron anteriormente. Además la transferencia para Nivel tiene un polo cercano a cero, lo que hace que el proyecto de controlar nivel con un controlador por Modelo de Referencia de error a la salida, tuviera grandes chances de fracasar.

La mayor complicación encontrada fue la falta de reglas prácticas para estimar las condiciones iniciales y parámetros a sintonizar, esto podría ser un proyecto a futuro, ya que para el actual el tiempo fue muy corto por tener que hacer una gran cantidad de tareas extras que impidieron profundizar en este tema dicho sea de paso no es para nada trivial.

6.8 Modelo de Referencia con error a la Entrada

El modelo de referencia con error a la entrada es muy similar al modelo de referencia con error a la salida, pero tiene la siguiente diferencia: en el modelo a la entrada se compara la entrada 'u' de la planta con $M^{-1}(y)$, mientras que en el MR con error a la salida se compara $M(u)$ con y_m (salida del modelo deseado). Por lo tanto para la adaptación de los parámetros se está evaluando la diferencia entre la 'u' que debería entrar a la planta y la que efectivamente lo hace. La estructura de control es idéntica a la del modelo de referencia de error a la salida.

6.8.1 ⁴¹Implementación del algoritmo

Este punto está basado en una traducción libre del texto de S. Sastry y M. Bodson: "*Adaptive Control- Stability, Convergence, and Robustness*" (Capítulo 3), publicado en 1989.

Para implementar el controlador se tiene:

- Suposiciones: (A1), (A2), (A3) y (A4)

(A1) Suposiciones de la planta

La planta es SISO y LTI, descrita por la función de transferencia

$$\frac{\hat{y}_p(s)}{\hat{u}(s)} = \hat{P}(s) = k_p \frac{\hat{n}_p(s)}{\hat{d}_p(s)} \quad (6.8.1.1)$$

Donde $\hat{n}_p(s)$, $\hat{d}_p(s)$ son polinomios mónicos y coprimos de grado 'm' y 'n' respectivamente. La planta es estrictamente propia ($n > m$) y de fase mínima. El signo de la ganancia de alta frecuencia k_p es conocido y sin pérdida de generalidad asumiremos $k_p > 0$.

(A2) Suposiciones del modelo de referencia

El modelo de referencia se describe por

$$\frac{\hat{y}_m(s)}{\hat{r}(s)} = \hat{M}(s) = k_m \frac{\hat{n}_m(s)}{\hat{d}_m(s)} \quad (6.8.1.2)$$

Donde $\hat{n}_m(s)$, $\hat{d}_m(s)$ son polinomios mónicos y coprimos de grado 'm' y 'n' respectivamente (los mismos grados correspondientes a los polinomios de la planta). El modelo de referencia es estable de fase mínima y $k_m > 0$.

(A3) Suposiciones de la entrada de referencia

La entrada de referencia $r(\cdot)$ es lineal y continua a tramos.

Notar que $\hat{P}(s)$ se asume de fase mínima, pero NO necesariamente estable.

(A4) Suposición sobre la ganancia de alta frecuencia

Se asume una cota superior en k_p conocida, o sea $k_p \leq k_{\max}$ para algún k_{\max}

⁴¹ (Sastry & Bodson, 1989)

- Datos

$$n, m, K_{\max}$$

- Entrada

$$r(t), y_p(t) \in \mathfrak{R}$$

- Salida

$$u(t) \in \mathfrak{R}$$

- Señales internas

$$w(t) \in \mathfrak{R}^{2n} [w^{(1)}(t), w^{(2)}(t) \in \mathfrak{R}^{n-1}]$$

$$\theta(t) \in \mathfrak{R}^{2n} [c_0(t), d_0(t) \in \mathfrak{R}, c(t), d(t) \in \mathfrak{R}^{n-1}]$$

$$v(t) \in \mathfrak{R}^{2n}, e_2(t) \in \mathfrak{R}$$

- Las condiciones iniciales son arbitrarias, salvo $c_0(0) \geq c_{\min} = \frac{k_m}{k_{\max}} > 0$

Parámetros de diseño:

$$* \hat{M}(k_m, \hat{n}_m, \hat{d}_m) \text{ satisfaciendo A2.}$$

$$* \Lambda \in \mathfrak{R}^{n-1 \times n-1}, b_\lambda \in \mathfrak{R}^{n-1}, \text{ de forma canónica controlable tal que } \det(sI - \Lambda) \text{ sea Hurwitz y contenga los ceros de } \hat{n}_m(s)$$

$$* \hat{L}^{-1} \text{ estable, función de transferencia de fase mínima de grado } n - m.$$

$$* g, \gamma > 0$$

Estructura de control:

$$\dot{w}^{(1)} = \Lambda w^{(1)} + b_\lambda u$$

$$\dot{w}^{(2)} = \Lambda w^{(2)} + b_\lambda y_p$$

$$\theta^T = (c_0, c^T, d_0, d^T)$$

$$w^T = (r, w^{(1)}, y_p, w^{(2)})$$

$$u = \theta^T w$$

Estructura del identificador:

$$v^T = \left[(\hat{M}\hat{L})^{-1}(y_p), \hat{L}^{-1}(w^{(1)T}), \hat{L}^{-1}(y_p), \hat{L}^{-1}(w^{(2)T}) \right]$$

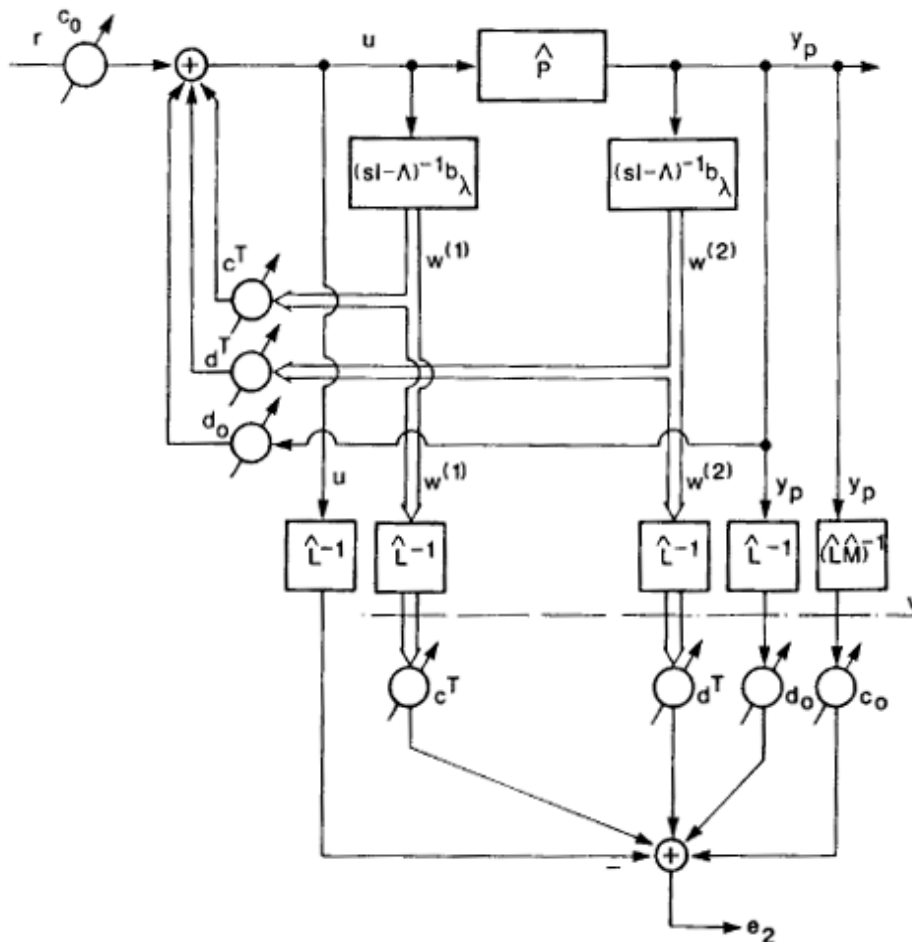
$$e_2 = \theta^T v - \hat{L}^{-1}(u)$$

Algoritmo del gradiente normalizado:

$$\dot{\theta} = -g \frac{e_2 v}{1 + \gamma w^T v}$$

Si $c_0 = c_{\min}$ y $\dot{c}_0 < 0$, entonces se setea $\dot{c}_0 = 0$.

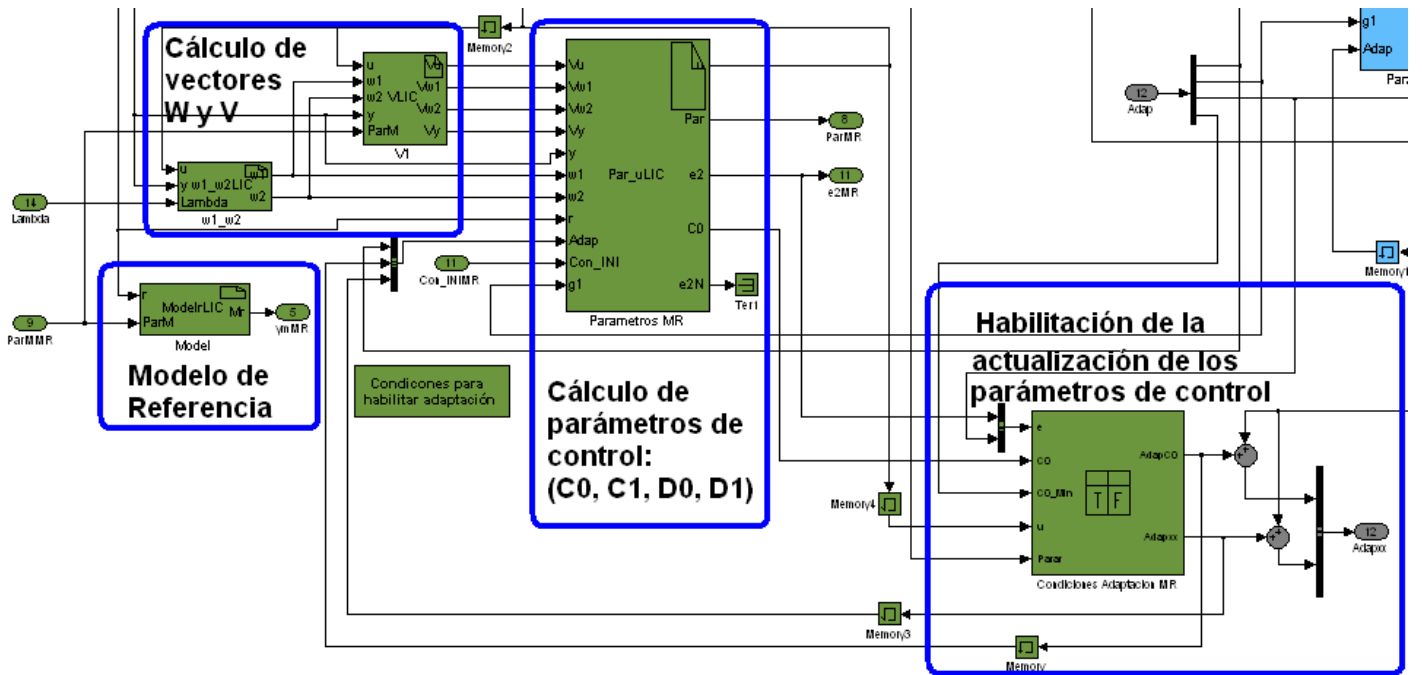
A continuación se muestra el controlador y estructura de identificación por error a la entrada:



6.8.2 Implementación Matlab

El diagrama utilizado para la simulación y posterior testeo de esta versión de modelo de referencia es muy similar al del MR con error a la salida. La única diferencia es el corazón del control, que antes era realizado por una S-Function, ahora es realizado por cuatro de éstas.

En la figura debajo se muestra la implementación práctica del modelo de referencia realizada en este proyecto y a continuación se explicaran en detalle los distintos bloques S-Function que intervienen en el algoritmo de modelo de referencia.



Programación S-Function: 'Model'

➤ Implementación ecuaciones diferenciales

Debajo se muestra la implementación de las ecuaciones diferenciales de la función de transferencia que se desea seguir:

```

/* Mr
    Par[0]*S + Par[1]
    -----
    S2 + Par[2]*S + Par[3]*/
dx[0] = xC[1] + ParM[0]*r[0];
dx[1] = -xC[1]*ParM[2] - xC[0]*ParM[3] + (ParM[1] -
ParM[0]*ParM[2])*r[0];

```

➤ Salidas S-Function

La salida es la solución de la ecuación diferencial arriba implementada:

```

/* Mr*/
Mr[0] = xC[0];

```

Programación S-Function: 'w1_w2'

➤ Implementación ecuaciones diferenciales

Se implementan las ecuaciones diferenciales para generar las señales w:

```

/* W1 = xC[0]*/
dx[0] = u[0] - xC[0]*Lambda[0];
/* W2 = xC[1]*/
dx[1] = y[0] - xC[1]*Lambda[0];

```

➤ **Salidas S-Function**

```
/* w1 */  
w1[0] = xC[0];  
/* w2 */  
w2[0] = xC[1];
```

Programación S-Function: 'V1'

➤ **Implementación ecuaciones diferenciales**

Se implementan las ecuaciones diferenciales de las señales V del algoritmo de modelo de referencia:

```
/* Vu  
          Par[0]*S + Par[1]  
          -----  
          S2 + Par[2]*S + Par[3]*/  
dx[0] = xC[1] + ParM[0]*u[0];  
dx[1] = -xC[1]*ParM[2] - xC[0]*ParM[3] + (ParM[1] -  
ParM[0]*ParM[2])*u[0];  
  
/* Vw1  
          Par[0]*S + Par[1]  
          -----  
          S2 + Par[2]*S + Par[3]*/  
dx[2] = xC[3] + ParM[0]*w1[0];  
dx[3] = -xC[3]*ParM[2] - xC[2]*ParM[3] + (ParM[1] -  
ParM[0]*ParM[2])*w1[0];  
  
/* Vw2  
          Par[0]*S + Par[1]  
          -----  
          S2 + Par[2]*S + Par[3]*/  
dx[4] = xC[5] + ParM[0]*w2[0];  
dx[5] = -xC[5]*ParM[2] - xC[4]*ParM[3] + (ParM[1] -  
ParM[0]*ParM[2])*w2[0];  
  
/* Vy  
          Par[0]*S + Par[1]  
          -----  
          S2 + Par[2]*S + Par[3]*/  
dx[6] = xC[7] + ParM[0]*y[0];  
dx[7] = -xC[7]*ParM[2] - xC[6]*ParM[3] + (ParM[1] -  
ParM[0]*ParM[2])*y[0];
```

➤ Salidas S-Function

Las salidas son las soluciones de las ecuaciones diferenciales implementadas en esta S-Function:

```
/* Vu*/  
Vu[0] = xC[0];  
/* Vw1*/  
Vw1[0] = xC[2];  
/* Vw2*/  
Vw2[0] = xC[4];  
/* Vy*/  
Vy[0] = xC[6];
```

Programación S-Function: 'Parametros MR'

➤ Implementación ecuaciones diferenciales

Se implementan las ecuaciones diferenciales del algoritmo de gradiente normalizado:

```
/* Actualización C0 = xC[0]*/  
dx[0] = y[0]*Adap[0]*e2N[0]*Adap[1]*Adap[2];  
/* Actualización C1 = xC[1]*/  
dx[1] = Vw1[0]*Adap[0]*e2N[0]*Adap[2];  
/* Actualización D0 = xC[3]*/  
dx[2] = Vw2[0]*Adap[0]*e2N[0]*Adap[2];  
/* Actualización D1 = xC[2]*/  
dx[3] = Vy[0]*Adap[0]*e2N[0]*Adap[2];
```

➤ Salidas S-Function

```
/* C0 = Par[0]*/  
Par[0] = xC[0] + Con_INI[0];  
C0[0] = Par[0];  
/* C1 = Par[1]*/  
Par[1] = xC[1] + Con_INI[1];  
/* D0 = Par[2]*/  
Par[2] = xC[2] + Con_INI[2];  
/* D1 = Par[3]*/  
Par[3] = xC[3] + Con_INI[3];  
/* u = r*C0 + w1*C1 + w2*D1 + D0*y */  
u[0] = r[0]*Par[0] + w1[0]*Par[1] + w2[0]*Par[3] +  
y[0]*Par[2];  
/* e2 = -(Vw1*C1 + Vw2*D1 + Vy*D0 + y*D1 - Vu)*/  
e2[0] = (Vw1[0]*Par[1] + Vw2[0]*Par[3] + Vy[0]*Par[2]  
+ y[0]*Par[0] - Vu[0]);  
e2N[0] = -e2[0]/(1 + g1[0]*(y[0]*y[0] + Vw1[0]*Vw1[0]  
+ Vw2[0]*Vw2[0] + Vy[0]*Vy[0]));
```

Programación de la Truth Table: 'Condiciones adaptación MR' (LIC-1)

Este bloque es el encargado de habilitar la adaptación de los parámetros del Modelo de referencia. El mismo tiene programado las condiciones normales en las cuales el algoritmo puede adaptar los parámetros del controlador (ver Truth Table del MR con error a la salida).

Block: PPiloto/LIC/Condiciones Adaptacion MR

File Edit Settings Add Help

Condition Table

	Description	Condition	D1	D2	D3	D4	D5
1	Abs(e0) < eMin	abs(e(1)) < e(2)	T	-	-	-	F
2	Parar actualizacion por saturacion de u	(u(1) >= 100) (u(1) <= 0)	-	T	-	-	F
3	Parar actualizacion	Parar == 1	-	-	T	-	F
4	Parar actualizacion de CO	(CO <= CO_Min) && (e(1) < 0)	-	-	-	T	F
	Actions: Specify a row from the Action Table		1	1	1	2	3

Action Table

#	Description	Action
1	Parar la actualizacion de todos lo parametros	AdapCO = 0; Adapxx = 0;
2	Parar solo la actualizacion de CO	AdapCO = 0; Adapxx = 1;
3	Actualizar parametros	AdapCO = 1; Adapxx = 1;

6.8.3 Simulaciones

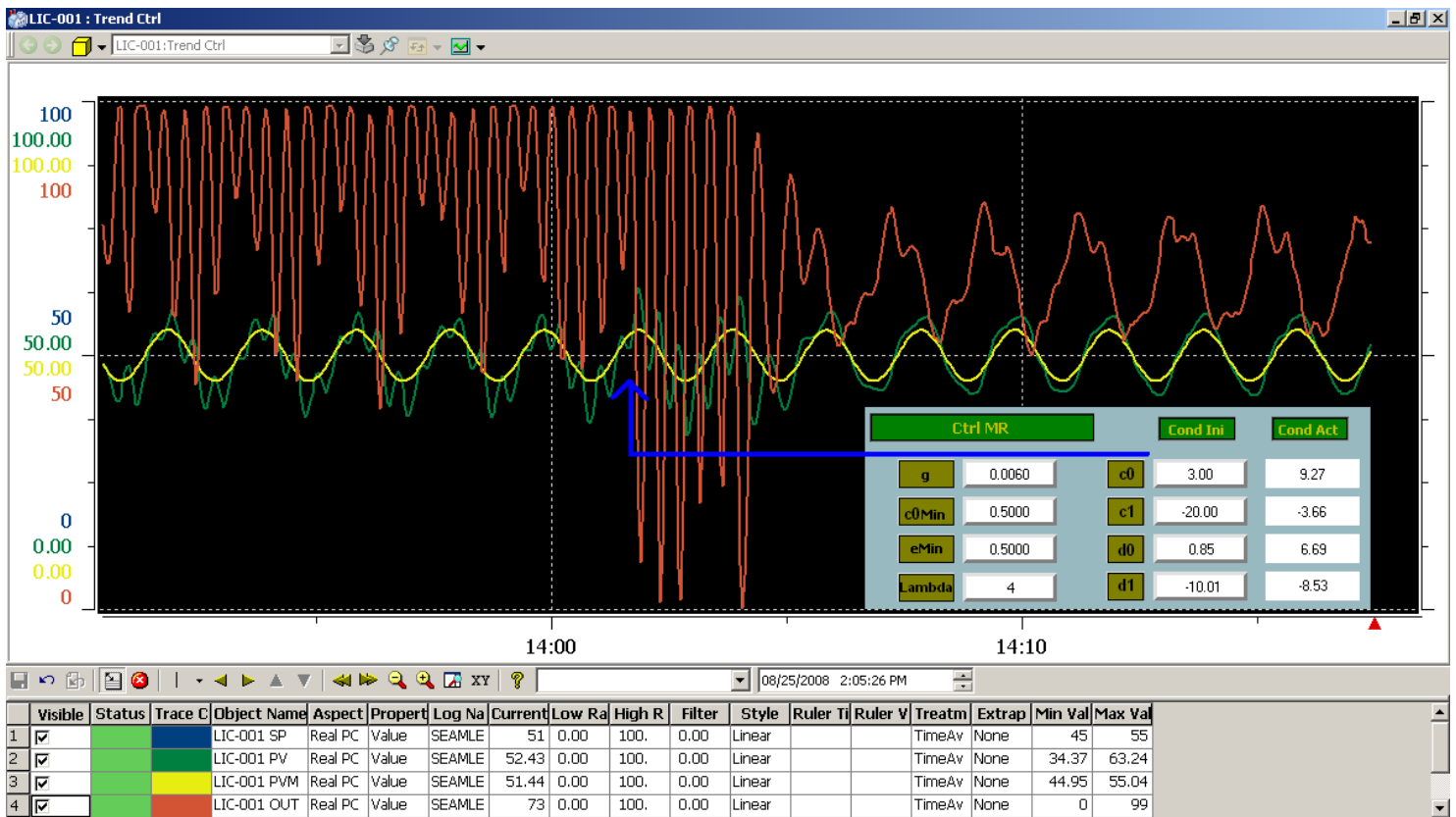
Las simulaciones llevadas a cabo muestran un comportamiento mucho mejor que el del MR con error a la salida, y las observaciones son casi las mismas que en el caso del MR anterior.

6.8.4 Prueba Física

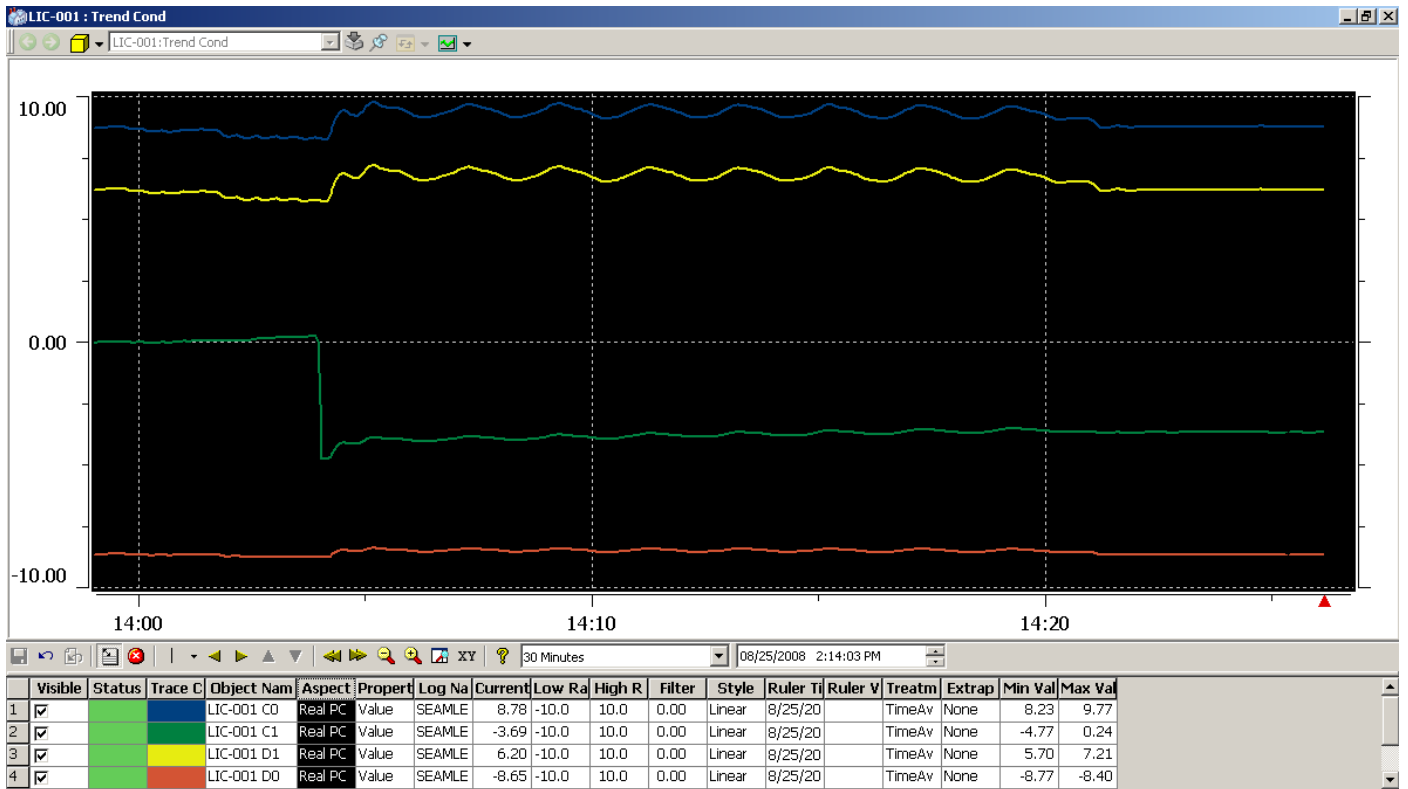
Para testear el controlador se utilizó la planta modelo: $P_M(s) = \frac{s+1/4}{s^2+s+1/4}$, un valor para 'g' (constante del algoritmo del gradiente) de 0.006, $\lambda = 4$ y condiciones iniciales $c_0 = 5.2$, $c_1 = -2.3$, $d_1 = -10$ y $d_0 = 0.85$. Además se lo utilizó como controlador de nivel.

Mediante un seno de baja amplitud y alto período se dejó al MR adaptar sus parámetros.

En la siguiente figura se observa la evolución de las señales de control, aquí se debe notar que las condiciones iniciales se cambiaron a mitad de la adaptación:

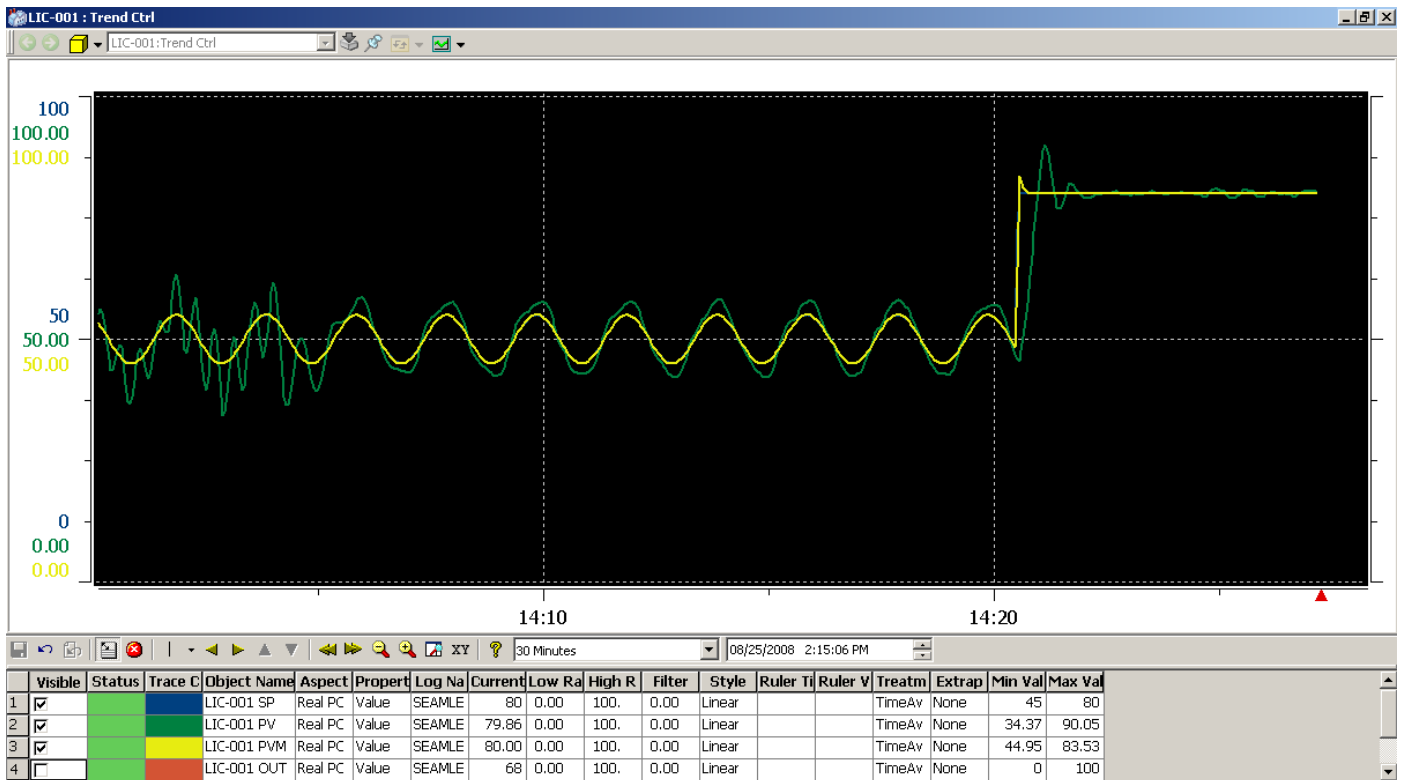


La evolución de los parámetros del controlador es la siguiente:



Lo más importante es que luego de un tiempo prolongado de adaptación se llega a que los parámetros del controlador convergen. Como puede verse a continuación esto se traduce en un control mucho más robusto que en el caso del MR con error a la salida.

Luego de dejar que “terminaran” de adaptarse los parámetros se cambió el SP, fijándolo en el 80% del nivel. En respuesta a este escalón (ver figura a continuación) se observa el muy buen desempeño del controlador:

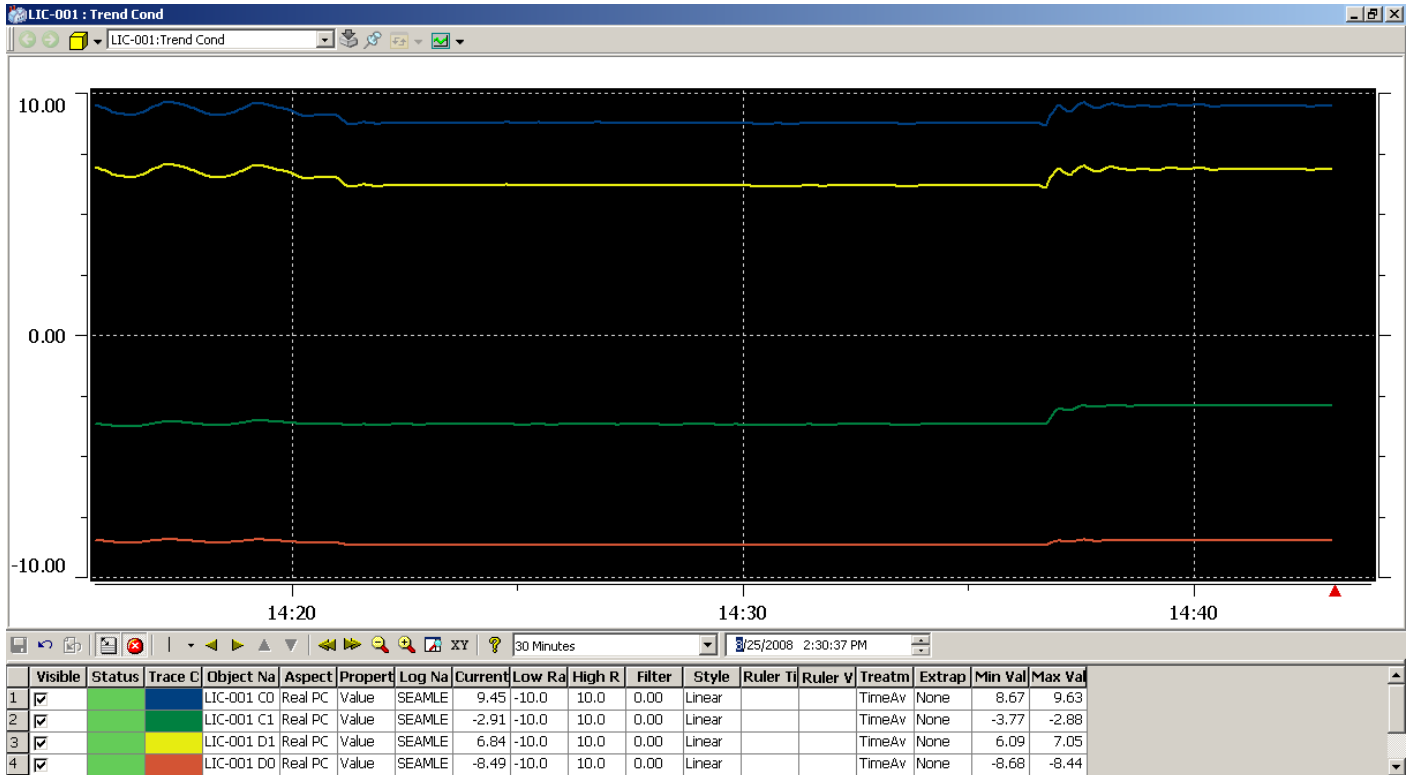


Finalmente se fija el SP en el 40% del nivel:



La señal que se incluyó en color rojo es la salida del controlador a la válvula de agua fría sin sumarle la onda cuadrada que elimina el efecto de histéresis. Esto se hizo de esta manera porque si se grafica la (salida del controlador)+(onda cuadrada) la imagen sería difícil de entender .

Por último, la evolución de los parámetros es mucho más estable que en el caso del MR con error a la salida.



6.8.5 Conclusiones

Tradicionalmente, el punto de partida en la deducción de los esquemas de control adaptivo por modelo de referencia ha sido el error a la salida: $e_0 = y_p - y_m$. El uso del error, entre la salida de la planta y la del modelo de referencia, para actualizar los parámetros del controlador es intuitivo. Sin embargo, las pruebas de estabilidad sugieren que el modelo debe satisfacer condiciones SPR (“Strictly Positive Real”, o sea debe tener polos negativos) y un error con una ponderación mayor cuando el grado relativo de la planta es mayor a uno. La deducción del esquema de error a la entrada muestra que el control adaptivo por modelo de referencia puede, en efecto, ser logrado formalmente sin involucrar el error a la salida y sin imponer condiciones de SPR al modelo.

Existen diferencias importantes entre los esquemas de error a la entrada y error a la salida. La primera es que la deducción de la ecuación de error, para el caso de error a la entrada, implica que la señal de entrada ‘u’ a la planta es igual (todo el tiempo) a la señal computada. Si dicha señal (‘u’) satura, las actualizaciones del identificador son erróneas. Cuando es utilizado un esquema de error a la entrada, este problema puede ser evitado, con tal de que el la verdadera señal de entrada a la planta sea accesible y utilizada por el identificador. Si es necesitado, los parámetros utilizados para la identificación y el control pueden también ser separados, y de ésta manera el identificador puede ser utilizado “off-line”.

Una segunda diferencia aparece entre los esquemas de error a la entrada y error a la salida cuando la ganancia de alta frecuencia K_p es desconocida y el grado relativo de la planta es mayor a uno. Aunque una ecuación de error SPR puede ser todavía obtenida en el caso de que la ganancia de alta frecuencia sea desconocida, una solución propuesta por Morse[1980] (y también Narendra, Lin y Valavani[1980]) requiere una sobreparametrización del identificador la cual excluye la posibilidad de estabilidad asintótica incluso cuando son satisfechas las condiciones de persistencia de excitación (Boyd y Sastry[1986], Abderson, Dasgupta y Tsoi[1985]). En vista de los de los ejemplos reciente debidos a Rohrs, y las conexiones entre convergencia exponencial y robustez (ver Sastry: “Stability, Convergence, and Robustness” capítulo 5), este parece ser el mayor inconveniente del algoritmo.

Otra ventaja del esquema de error a la entrada es que conduce a una ecuación de error lineal para la cual otros algoritmos de identificación, como por ejemplo mínimos-cuadrados, son aplicables. Estos algoritmos son una alternativa ventajosa al algoritmo del gradiente. Además, fue mostrado que hay ventajas del esquema de error a la entrada en términos de robustez hacia dinámicas sin modelar (Bodson[1988]). En algunos casos, el esquema de error a la entrada requiere de más cálculos. Esto se debe a que los observadores $\omega^{(1)}$, $\omega^{(2)}$ son de orden ‘n’, en lugar de ‘n-1’ como lo es en el caso del esquema de error a la salida. También el filtro \hat{L}^{-1} es de un orden mayor. Cuando el orden relativo de la planta es 1, simplificaciones significantes surgen en el esquema del error a la salida (ese es el caso del MR con error a la salida implementado en este proyecto).

A modo de resumen, se concluye que este controlador es lo suficientemente robusto como para dar cumplimiento a la implementación de un control MR y además se lo deja como controlador opcional para los lazos de control de nivel y temperatura.

7 Identificación

Como inicialmente las pruebas realizadas a los controladores por modelo de referencia no daban buenos resultados se decidió implementar la Identificación de forma separada al controlador y estudiar su funcionamiento.

Para esto se detalla la teoría de Identificación para una planta (con función de transferencia propia) de orden 'n' y se estudian en particular las identificaciones para plantas de segundo y primer orden.

Finalmente se explica la implementación de la Identificación en Matlab y se concluye con la convergencia de sus parámetros a posibles modelos de la planta.

7.1 Introducción

Partiendo de una planta desconocida (con función de transferencia propia) la identificación consiste en lograr identificarla, o sea relevar su función de transferencia.

Dado un sistema SISO y LTI descrito por la función de transferencia:

$$\frac{y_p(s)}{r(s)} = P(s) = K_p \cdot \frac{n_p(s)}{d_p(s)} \quad (7.1)$$

donde $r(s)$ y $y_p(s)$ son las transformadas de Laplace correspondientes a la entrada y salida de la planta, respectivamente, y $n_p(s)$ y $d_p(s)$ son polinomios mónicos y coprimos de grados 'm' y 'n' respectivamente. El coeficiente 'm' es desconocido pero la planta es estrictamente propia ($m \leq n - 1$). Además se asume que $r(s)$ es continua a tramos y que la planta es estable.

7.2 Estructura del Identificador

Este punto está basado en una traducción libre del texto de S. Sastry y M. Bodson: "*Adaptive Control- Stability, Convergence, and Robustness*" (Capítulo 2), publicado en 1989.

La estructura del identificador que se presenta a continuación es generalmente conocida como identificador de "Error de Ecuación" (Ljung & Soderstrom [1983]). La función de transferencia $P(s)$ puede ser escrita explícitamente como:

$$\frac{y_p(s)}{r(s)} = P(s) = \frac{\alpha_n s^{n-1} + \dots + \alpha_1}{s^n + \beta_n s^{n-1} + \dots + \beta_1} \quad (7.2)$$

En la misma son desconocidos $2 \cdot n$ coeficientes: $\alpha_n \dots \alpha_1$ y $\beta_n \dots \beta_1$. Esta expresión es una parametrización de una planta (desconocida), que es un modelo en el cual sólo un finito número de parámetros serán determinados. Con motivo de la identificación, es conveniente encontrar una expresión que dependa linealmente de los parámetros desconocidos. Por ejemplo la siguiente expresión es lineal en los parámetros α_i y β_i :

$$s^n y_p(s) = (\alpha_n s^{n-1} + \dots + \alpha_1) r(s) - (\beta_n s^{n-1} + \dots + \beta_1) y_p(s) \quad (7.3)$$

Sin embargo, requeriría derivadores para ser implementada. Para evitar este problema se introduce el polinomio mónico de n-ésimo orden: $\lambda(s) = s^n + \lambda_n s^{n-1} + \dots + \lambda_1$. Este polinomio se asume Hurwitz pero es arbitrario. Luego haciendo uso de la (7.1):

$$\frac{y_p(s)}{r(s)} = \frac{y_p(s) \cdot (1 + \lambda(s) - \lambda(s))}{r(s)} = K_p \cdot \frac{n_p(s)}{d_p(s)} \Rightarrow$$

$$\lambda(s) \cdot y_p(s) = K_p \cdot n_p(s) \cdot r(s) + (\lambda(s) - d_p(s)) \cdot y_p(s) \quad (7.4)$$

$$\text{De (7.2): } y_p(s) = \frac{\alpha_n s^{n-1} + \dots + \alpha_1}{\lambda(s)} \cdot r(s) + \frac{(\lambda_n - \beta_n) s^{n-1} + \dots + (\lambda_1 - \beta_1)}{\lambda(s)} \cdot y_p(s) \quad (7.5)$$

Esta expresión es una nueva parametrización de la planta. Para simplificar la notación, llamemos:

$$\begin{aligned} a(s) &= \alpha_n s^{n-1} + \dots + \alpha_1 = K_p \cdot n_p(s) \\ b(s) &= (\lambda_n - \beta_n) s^{n-1} + \dots + (\lambda_1 - \beta_1) = \lambda(s) - d_p(s) \end{aligned} \quad (7.6)$$

Entonces la nueva representación de la planta puede ser escrita como:

$$y_p(s) = \frac{a(s)}{\lambda(s)} \cdot r(s) + \frac{b(s)}{\lambda(s)} \cdot y_p(s) \quad (7.7)$$

De ésta manera la función de transferencia queda: $\frac{y_p(s)}{r(s)} = \frac{a(s)}{\lambda(s) - b(s)}$ y verifica que es la función de transferencia de la planta P(s).

7.3 Representación en espacio de estados

Este punto está basado en una traducción libre del texto de S. Sastry y M. Bodson: “*Adaptive Control- Stability, Convergence, and Robustness*” (Capítulo 2), publicado en 1989.

Una representación en espacio de estados de la expresión anterior puede ser encontrada eligiendo $\Delta \in \mathfrak{R}^n, b_\lambda \in \mathfrak{R}^n$ de forma canónica controlable, como la que se muestra a continuación:

$$\Delta = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & 0 \\ 0 & 0 & \cdot & \dots & 1 \\ -\lambda_1 & \cdot & \cdot & \dots & -\lambda_n \end{bmatrix} \quad b_\lambda = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} \quad \Rightarrow (sI - \Delta)^{-1} \cdot b_\lambda = \frac{1}{\lambda(s)} \begin{bmatrix} 1 \\ s \\ \cdot \\ \cdot \\ \cdot \\ s^{n-1} \end{bmatrix} \quad (7.8)$$

En analogía con la (7.6), se definen:

$$\begin{aligned} a^{*T} &:= (\alpha_1, \dots, \alpha_n) \\ b^{*T} &:= (\lambda_1 - \beta_1, \dots, \lambda_n - \beta_n) \end{aligned} \quad (7.9)$$

y los vectores $\omega_p^{(1)}(t), \omega_p^{(2)}(t) \in \mathfrak{R}^n$:

$$\begin{aligned} \dot{\omega}_p^{(1)} &= \Delta \cdot \omega_p^{(1)} + b_\lambda \cdot r \\ \dot{\omega}_p^{(2)} &= \Delta \cdot \omega_p^{(2)} + b_\lambda \cdot y_p \end{aligned} \quad (7.10)$$

con condiciones iniciales $\omega_p^{(1)}(0)$, $\omega_p^{(2)}(0)$. En transformadas de Laplace:

$$\begin{aligned} \omega_p^{(1)} &= (s \cdot I - \Delta)^{-1} \cdot b_\lambda \cdot r(s) + (s \cdot I - \Delta)^{-1} \cdot \omega_p^{(1)}(0) \\ \omega_p^{(2)} &= (s \cdot I - \Delta)^{-1} \cdot b_\lambda \cdot y_p(s) + (s \cdot I - \Delta)^{-1} \cdot \omega_p^{(2)}(0) \end{aligned} \quad (7.11)$$

Con esta notación la descripción de la planta (7.7) se transforma en:

$$y_p(s) = a^{*T} \cdot \omega_p^{(1)}(s) + b^{*T} \cdot \omega_p^{(2)}(s) \quad (7.12)$$

Como los parámetros 'a*' y 'b*' son constantes, la misma expresión es válida en el dominio del tiempo:

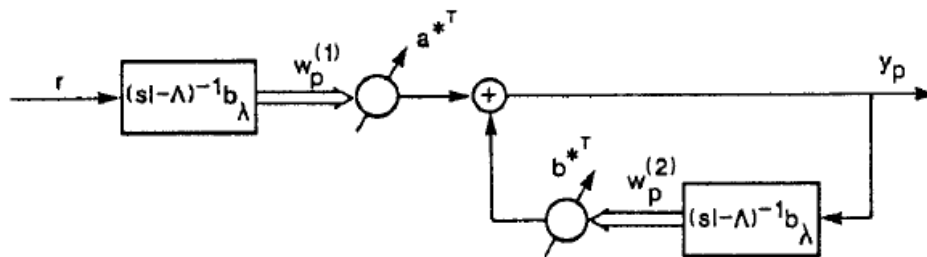
$$y_p(t) = a^{*T} \cdot \omega_p^{(1)}(t) + b^{*T} \cdot \omega_p^{(2)}(t) := \theta^{*T} \cdot \omega_p(t) \quad (7.13)$$

donde se definió:

$$\begin{aligned} \theta^{*T} &:= (a^{*T}, b^{*T}) \in \mathfrak{R}^{2n} \\ \omega_p^T(t) &:= (\omega_p^{(1)T}(t), \omega_p^{(2)T}(t)) \in \mathfrak{R}^{2n} \end{aligned} \quad (7.14)$$

Las ecuaciones 9-13 definen una realización de la nueva parametrización. El vector ω_p es el estado generalizado de la planta y tiene dimensión $2n$. Por eso, la realización de $P(s)$ no es mínima.

El vector θ^* está formado por parámetros desconocidos relacionados linealmente con los parámetros originales de la planta α_i , β_i por las ecuaciones 9-14. El conocimiento de un juego de parámetros es equivalente al conocimiento del otro y cada uno corresponde a una de las (equivalentes) parametrizaciones. Sin embargo, en la última parametrización, la salida de la planta depende linealmente de los parámetros desconocidos, entonces los algoritmos estándar de identificación pueden ser usados. Ésta parametrización de la planta es representada en la siguiente figura:



Parametrización de la Planta

7.4 Estructura del Identificador

Este punto está basado en una traducción libre del texto de S. Sastry y M. Bodson: “*Adaptive Control- Stability, Convergence, and Robustness*” (Capítulo 2), publicado en 1989.

El propósito del identificador es producir una estimación $\theta(t)$ recursiva de los parámetros nominales θ^* . Como ‘r’ y ‘y_p’ son accesibles se define el observador:

$$\begin{aligned} \dot{\omega}^{(1)} &= \Delta \cdot \omega^{(1)} + b_\lambda \cdot r \\ \dot{\omega}^{(2)} &= \Delta \cdot \omega^{(2)} + b_\lambda \cdot y_p \end{aligned} \quad (7.15)$$

para reconstruir los estados de la planta. Las condiciones iniciales en la (7.15) son arbitrarias. También se definen las señales del identificador:

$$\begin{aligned} \theta^T &:= (a^T, b^T) \in \mathfrak{R}^{2n} \\ \omega^T(t) &:= (\omega^{(1)T}(t), \omega^{(2)T}(t)) \in \mathfrak{R}^{2n} \end{aligned} \quad (7.16)$$

A partir de las (7.10) y (7.15) el error del observador $\omega(t) - \omega_p(t)$ decae exponencialmente a cero, incluso cuando la planta es inestable. Se debe notar que el estado generalizado de la planta $\omega_p(t)$ es tal que puede ser reconstruido a partir de las señales accesibles, sin conocimiento de los parámetros de la planta.

La salida de la planta puede ser escrita como: $y_p(t) = \theta^{*T} \cdot \omega(t) + \varepsilon(t)$ (7.17), donde la señal $\varepsilon(t)$ está para recordar la presencia de un término aditivo que decae exponencialmente dado por:

$$\varepsilon(t) = \theta^{*T} \cdot (\omega_p(t) - \omega(t)) \quad (7.18)$$

Este término es debido a las condiciones iniciales del observador. En este razonamiento se desestimará la presencia del mismo pero se puede demostrar que no afecta las propiedades del identificador (Ver Sastry, Teorema 2.4.3).

En analogía con la expresión de la salida de la planta, la salida del identificador está definida por:

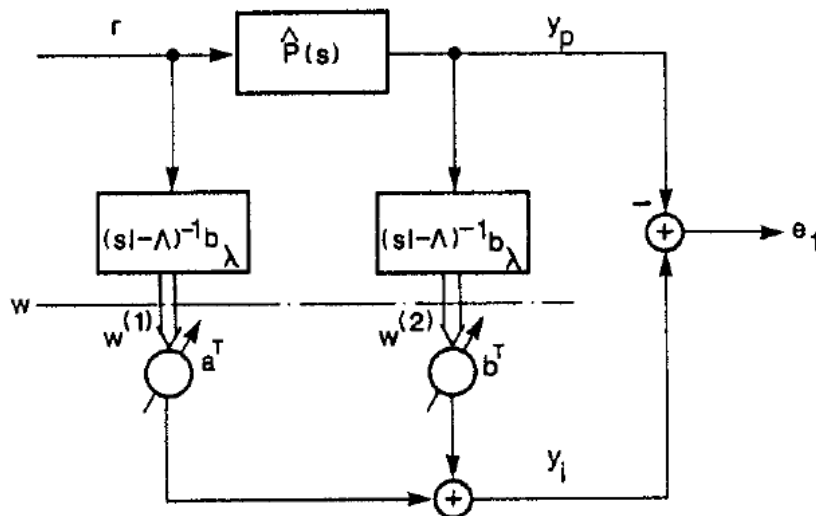
$$y_i(t) = \theta^T \cdot \omega(t) \in \mathfrak{R} \quad (7.19)$$

También se define el parámetro de error:

$$\phi(t) = \theta^T(t) - \theta^{*T} \in \mathfrak{R}^{2n} \quad (7.20)$$

y el error de identificación: $e_1(t) = y_i(t) - y_p(t) = \phi^T(t) \cdot \omega(t) + \varepsilon(t)$ (7.21)

Estas señales son usadas por el algoritmo de identificación, y son representadas en la siguiente figura:



Estructura del Identificador

7.5 Ecuación de error lineal y algoritmo de Identificación

Este punto está basado en una traducción libre del texto de S. Sastry y M. Bodson: “*Adaptive Control- Stability, Convergence, and Robustness*”(Capítulo 2), publicado en 1989.

Muchos algoritmos de Identificación (Eykhoff [1974], Ljung & Soderstrom [1983]) se confían en una expresión lineal de la forma obtenida anteriormente, ésta es:

$$y_p(t) = \theta^{*T} \cdot \omega(t) \quad (7.22)$$

donde $y_p(t)$ y $\omega(t)$ son señales conocidas y θ^* es desconocida. El vector $\omega(t)$ es usualmente denominado vector regresor. Con la expresión de $y_p(t)$ está asociada la ecuación del error lineal estándar:

$$e_1(t) = \phi^T(t) \cdot \omega(t) \quad (7.23)$$

Se separó arbitrariamente el identificador en una estructura de identificador y un algoritmo de identificación. La estructura del identificador construye el regresor $\omega(t)$ y otras señales, relacionadas por la ecuación de error del identificador. El algoritmo de identificación está definido por una ecuación diferencial, llamada ley de actualización, de la forma:

$$\dot{\theta} = \dot{\phi} = F(y_p, e_1, \theta, \omega) \quad (7.24)$$

donde ‘F’ es un operador casual explícitamente independiente de θ^* , el cual define la evolución del parámetro θ .

Algoritmo del Gradiente

La siguiente ley de actualización define el algoritmo estándar del gradiente:

$$\dot{\theta} = -g \cdot e_1 \cdot \omega \quad \text{con } g > 0 \quad (7.25)$$

El lado derecho de la ecuación es proporcional al gradiente del error de salida al cuadrado pues:

$$\frac{\partial}{\partial \theta} (e_1^2(\theta)) = 2 \cdot e_1 \cdot \omega \quad (7.26)$$

Esta ley de actualización puede así ser vista como un método “*steepest descent*”. El parámetro ‘g’ es una ganancia fija y estrictamente positiva que se denomina ganancia de adaptación. La misma permite variar la tasa de adaptación de los parámetros. Las condiciones iniciales $\theta(0)$ son arbitrarias, pero pueden ser elegidas para tomar en cuenta cualquier conocimiento, a priori, de los parámetros de la planta.

7.6 Identificación de planta de 2º orden

Sea $P(s) = K_P \cdot \frac{d_P(s)}{n_P(s)} = \frac{\alpha_2^* \cdot s + \alpha_1^*}{s^2 + \beta_2^* \cdot s + \beta_1^*} = \frac{y_P(s)}{r(s)}$ un posible modelo para la planta

desconocida a identificar.

Se introduce el polinomio arbitrario $\lambda(s) = s^2 + \lambda_2 \cdot s + \lambda_1$.

Luego se suma y resta el polinomio $\lambda(s)$ de la siguiente manera:

$$\frac{y_P(s)}{r(s)} = \frac{y_P(s) \cdot (1 + \lambda(s) - \lambda(s))}{r(s)} = K_P \cdot \frac{n_P(s)}{d_P(s)}$$

Operando se llega a que: $\lambda(s) \cdot y_P(s) = K_P \cdot n_P(s) \cdot r(s) + (\lambda(s) - d_P(s)) \cdot y_P(s)$, que reescrito de otra manera es igual a:

$$y_P(s) = \frac{\alpha_2^* s + \alpha_1^*}{s^2 + \lambda_2 \cdot s + \lambda_1} \cdot r(s) + \frac{(\lambda_2 - \beta_2^*)s + (\lambda_1 - \beta_1^*)}{s^2 + \lambda_2 \cdot s + \lambda_1} \cdot y_P(s)$$

Para simplificar la notación, llamemos:

$$\begin{aligned} a_i^* &= \alpha_i^* \\ b_i^* &= \lambda_i - \beta_i^* \end{aligned} \quad \forall i \in [1,2]$$

$$\text{Entonces: } y_P(s) = \frac{a_2^* s + a_1^*}{s^2 + \lambda_2 \cdot s + \lambda_1} \cdot r(s) + \frac{b_2^* s + b_1^*}{s^2 + \lambda_2 \cdot s + \lambda_1} \cdot y_P(s)$$

Introduciendo los siguientes observadores:

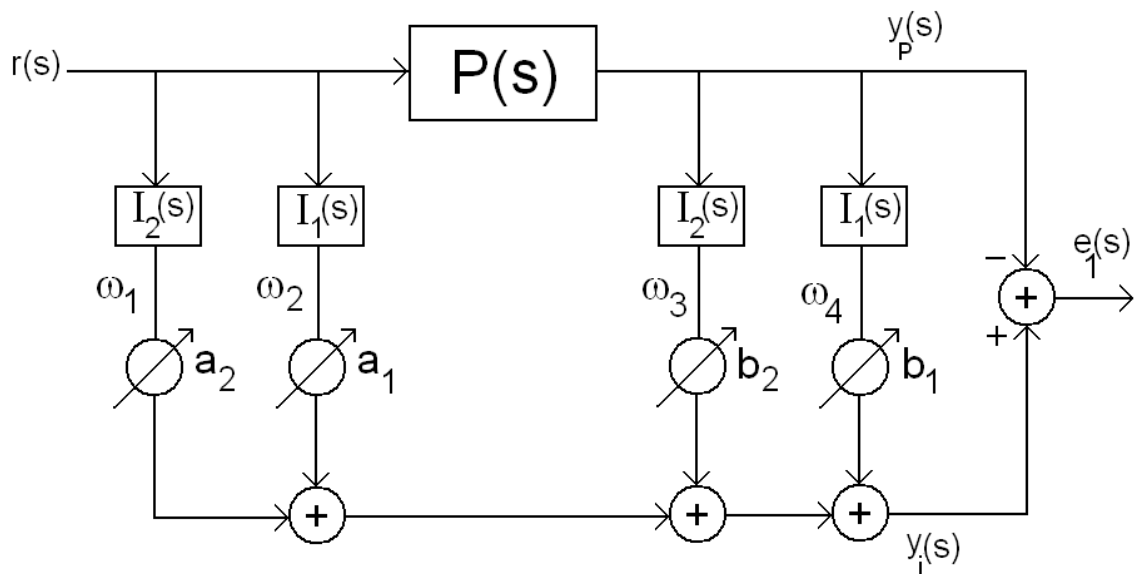
$$I_2(s) = \frac{s}{s^2 + \lambda_2 \cdot s + \lambda_1}$$

$$I_1(s) = \frac{1}{s^2 + \lambda_2 \cdot s + \lambda_1}$$

la salida de la planta identificada es:

$$y_i(s) = (a_2 \cdot I_2(s) + a_1 \cdot I_1(s)) \cdot r(s) + (b_2 \cdot I_2(s) + b_1 \cdot I_1(s)) \cdot y_P(s)$$

la misma se puede observar en el siguiente esquema:



El algoritmo de identificación propuesto es:

$$\dot{\theta} = \begin{bmatrix} \dot{a}_2 \\ a_2 \\ \dot{a}_1 \\ a_1 \\ \dot{b}_2 \\ b_2 \\ \dot{b}_1 \\ b_1 \end{bmatrix} = -g \cdot e_1 \cdot \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = -g \cdot e_1 \cdot \begin{bmatrix} I_2 \cdot r \\ I_1 \cdot r \\ I_2 \cdot y_P \\ I_1 \cdot y_P \end{bmatrix}$$

con 'g' > 0 y $\lambda(s) = s^2 + \lambda_2 \cdot s + \lambda_1$ arbitrario, pero Hurwitz.

Una vez obtenidos las estimaciones de los a_i^* y b_i^* , para obtener la transferencia estimada de la planta se debe deshacer el cambio de variable anteriormente planteado, entonces:

$$\begin{aligned} \alpha_i &= a_i \\ \beta_i &= \lambda_i - b_i \end{aligned} \quad \forall i \in [1,2]$$

Por lo tanto la estimación de la planta es:

$$P_i(s) = \frac{\alpha_2 s + \alpha_1}{s^2 + \beta_2 \cdot s + \beta_1}$$

que se demuestra converge a la $P(s)$ original.

7.7 Identificación de planta de 1º orden

Para este caso el razonamiento es análogo al anterior con la salvedad de que ahora la planta es de la forma: $P(s) = \frac{\alpha_1^*}{s + \beta_1^*}$, por lo que en este caso no intervienen ni a_2 , ni b_2 , ni λ_2 , simplificándose las ecuaciones a:

$$\dot{\theta} = \begin{bmatrix} \dot{a}_1 \\ a_1 \\ \dot{b}_1 \\ b_1 \end{bmatrix} = -g \cdot e_1 \cdot \begin{bmatrix} \omega_2 \\ \omega_4 \end{bmatrix} = -g \cdot e_1 \cdot \begin{bmatrix} I_1 \cdot r \\ I_1 \cdot y_P \end{bmatrix}$$

con $I_1(s) = \frac{1}{s + \lambda_1}$.

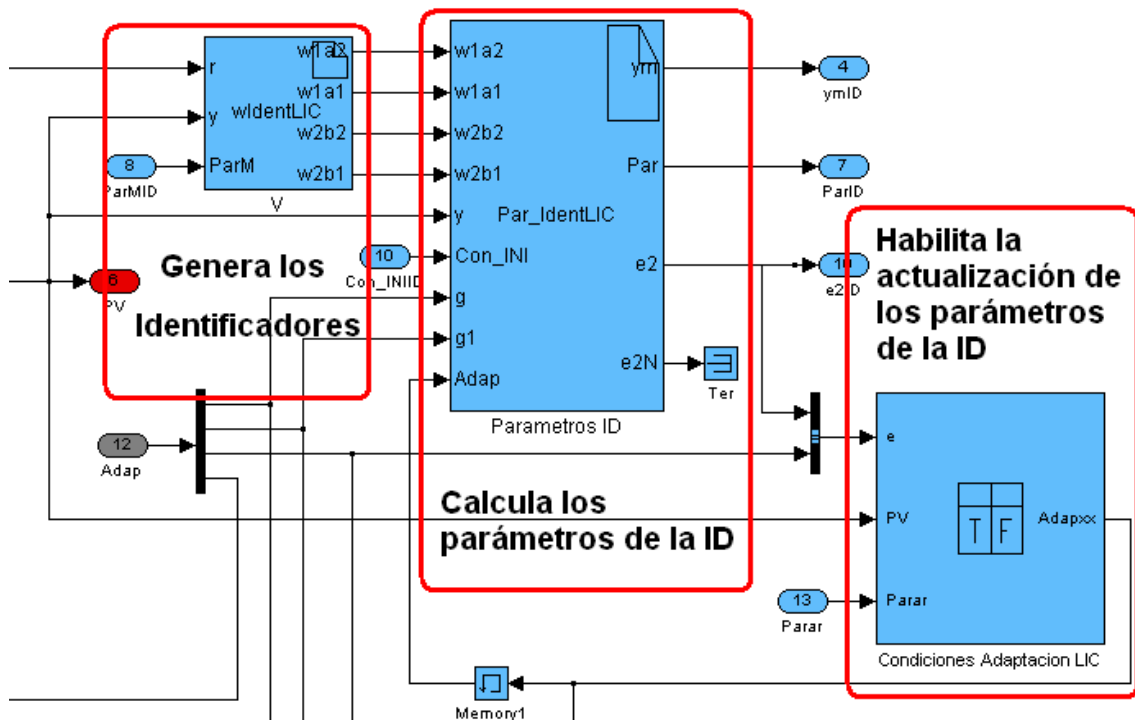
7.8 Implementación de la Identificación en Matlab

Para llevar a cabo la identificación se utilizaron bloques S-Function, esto se puede apreciar en la siguientes figura. Con el bloque de la izquierda ('wIdentLIC' en éste caso) se crean las transferencias de los identificadores para así luego obtener: $\omega_1, \omega_2, \omega_3$ y ω_4 , mientras que con el de la derecha se lleva a cabo el método del gradiente obteniéndose los parámetros buscados.

En el esquema de la identificación lo único que puede resultar extraño es una tabla de verdad a la derecha de la figura, su función es decidir si se actualizan o no los parámetros en cuestión. La inclusión de ésta lógica se debe fundamentalmente a que se optó por no actualizar los parámetros de la identificación si

- el error entre la salida Identificada y el PV real es menor que un error mínimo.
- el PV está ceca de la saturación.
- se desea parar la identificación.

A continuación se explicara en detalle los distintos bloques S-Function que intervienen en el algoritmo de identificación.



Programación S-Function: ‘V’

Este bloque es el encargado de construir las señales ‘w’ (salidas de los identificadores) necesarias para la identificación. La programación y la explicación de su programación se detallan a continuación.

➤ Implementación de ecuaciones diferenciales

Una de las ventajas de los bloques S-Function es que permiten escribir ecuaciones diferenciales de igual manera de cómo se haría en la teoría. En esta parte se escriben las ecuaciones diferenciales para generar las transferencias necesarias para identificación. En el caso de ‘wa2’ y ‘wb2’ el parámetro Par[0] es igual a 1 mientras que Par[1] es igual a 0. En las señales ‘wa1’ y ‘wb1’ el parámetro Par[0] es igual a 0 mientras que Par[1] es igual a 1. En todas las señales ‘w’ el parámetro Par[2] es igual a λ_1 y Par[3] es igual a λ_2 .

```
/* w1a2
    Par[0]*S + Par[1 ]
    -----
    S2 + Par[2]*S + Par[3]*/
dx[0] = xC[1] + r[0];
dx[1] = -xC[1]*ParM[2] - xC[0]*ParM[3] - ParM[2]*r[0];

/* w1a1
    Par[0]*S + Par[1]
    -----
    S2 + Par[2]*S + Par[3]*/
dx[2] = xC[3];
dx[3] = -xC[3]*ParM[2] - xC[2]*ParM[3] + r[0];

/* w2b2
    Par[0]*S + Par[1]
    -----
    S2 + Par[2]*S + Par[3]*/
dx[4] = xC[5] + y[0];
dx[5] = -xC[5]*ParM[2] - xC[4]*ParM[3] - ParM[2]*y[0];

/* w2b1
    Par[0]*S + Par[1]
    -----
    S2 + Par[2]*S + Par[3]*/
dx[6] = xC[7];
dx[7] = -xC[7]*ParM[2] - xC[6]*ParM[3] + y[0];
```

➤ Salidas

En la parte de salida de esta S-Function se instancian las salidas de las señales w, que en definitiva son la solución a las ecuaciones diferenciales antes descritas

```
/* Vu*/  
w1a2[0] = xC[0];  
/* Vw1*/  
w1a1[0] = xC[2];  
/* Vw2*/  
w2b2[0] = xC[4];  
/* Vy*/  
w2b1[0] = xC[6];
```

Programación S-Function: 'Parámetros ID'

➤ Implementación de ecuaciones diferenciales

Se implementan las ecuaciones diferenciales necesarias para resolver el algoritmo adaptativo de gradiente normalizado.

```
/* Actualización a2 = xC[0]*/  
dx[0] = g[0]*w1a2[0]*e2N[0]*Adap[0];  
/* Actualización a1 = xC[1]*/  
dx[1] = g[0]*w1a1[0]*e2N[0]*Adap[0];  
/* Actualización b2 = xC[3]*/  
dx[2] = g[0]*w2b2[0]*e2N[0]*Adap[0];  
/* Actualización b1 = xC[2]*/  
dx[3] = g[0]*w2b1[0]*e2N[0]*Adap[0];
```

➤ Salidas

Como salidas se tiene la evolución de los parámetros de la identificación los cuales son la solución a las ecuaciones diferenciales antes descritas. También se genera la salida identificada la cual es comparada con la salida real de la planta para generar el error que luego es utilizado en las ecuaciones diferenciales.

```
/* a2 = Par[0]*/  
Par[0] = xC[0] + Con_INI[0];  
/* a1 = Par[1]*/  
Par[1] = xC[1] + Con_INI[1];  
/* b2 = Par[2]*/  
Par[2] = xC[2] + Con_INI[2];  
/* b1 = Par[3]*/  
Par[3] = xC[3] + Con_INI[3];  
/* u = r*C0 + w1*C1 + w2*D1 + D0*y */  
ym[0] = w1a2[0]*Par[0] + w1a1[0]*Par[1] + w2b2[0]*Par[2] +  
w2b1[0]*Par[3];  
/* e2 = -(Vw1*C1 + Vw2*D1 + Vy*D0 + y*D1 - Vu)*/  
e2[0] = -(ym[0]-y[0]);  
e2N[0] = e2[0]/(1 + g1[0]*(w1a2[0]*w1a2[0] +  
w1a1[0]*w1a1[0] + w2b2[0]*w2b2[0] + w2b1[0]*w2b1[0]));
```

Programación Truth Table: ‘Condiciones adaptación LIC’

Este bloque es el encargo de habilitar la identificación, el cual tiene programado las condiciones en las cuales el algoritmo puede identificar en condiciones normales.

The screenshot shows a software window titled "Block: PPiloto/PIDLIC/Condiciones AdaptacionLIC". The window contains two tables:

Condition Table

	Description	Condition	D1	D2	D3	D4
1	Abs(e0) < eMin	abs(e(1)) < e(2) abs(e(1)) > 40	T	-	-	F
2	Parar actualizacion por PV altos	(PV <= 5) (PV >= 95)	-	T	-	F
3	Parar == 1 Para actualizacion	Parar == 1	-	-	T	F
Actions: Specify a row from the Action Table			1	1	1	2

Action Table

#	Description	Action
1	Parar la actualizacion de todos lo parametros	Adapxx = 0;
2	Actualizar parametros	Adapxx = 1;

Para el caso de la función de transferencia de nivel, la identificación implementada es de orden dos, mientras que para la función de transferencia de temperatura, es de primer orden. Ésta elección se basó en las respuestas relevadas para cada variable.

7.9 Pruebas y resultados

El controlador por Modelo de Referencia implícitamente realiza una identificación de la planta. Al concluir el testeo de la última versión del mismo se observó la convergencia en el tiempo de sus parámetros. Por este motivo resulta lógico suponer que la identificación por sí sola tiene que funcionar. En realidad luego de fracasar las pruebas del MR como controlador de nivel, se optó por implementar una Identificación para que de ésta forma se compensara el vacío resultante. Pero al obtenerse buenos resultados con la misma se siguió insistiendo con el MR, hasta que finalmente el mismo se comportó de la forma esperada.

En la práctica se comprobó el correcto funcionamiento del algoritmo, pues los parámetros de la Identificación, luego de un tiempo, permanecen bastantes estables, dando como resultado una casi coincidencia entre el PV (real) y el PV (calculado) de la planta Identificada.

8 Manual de Usuario

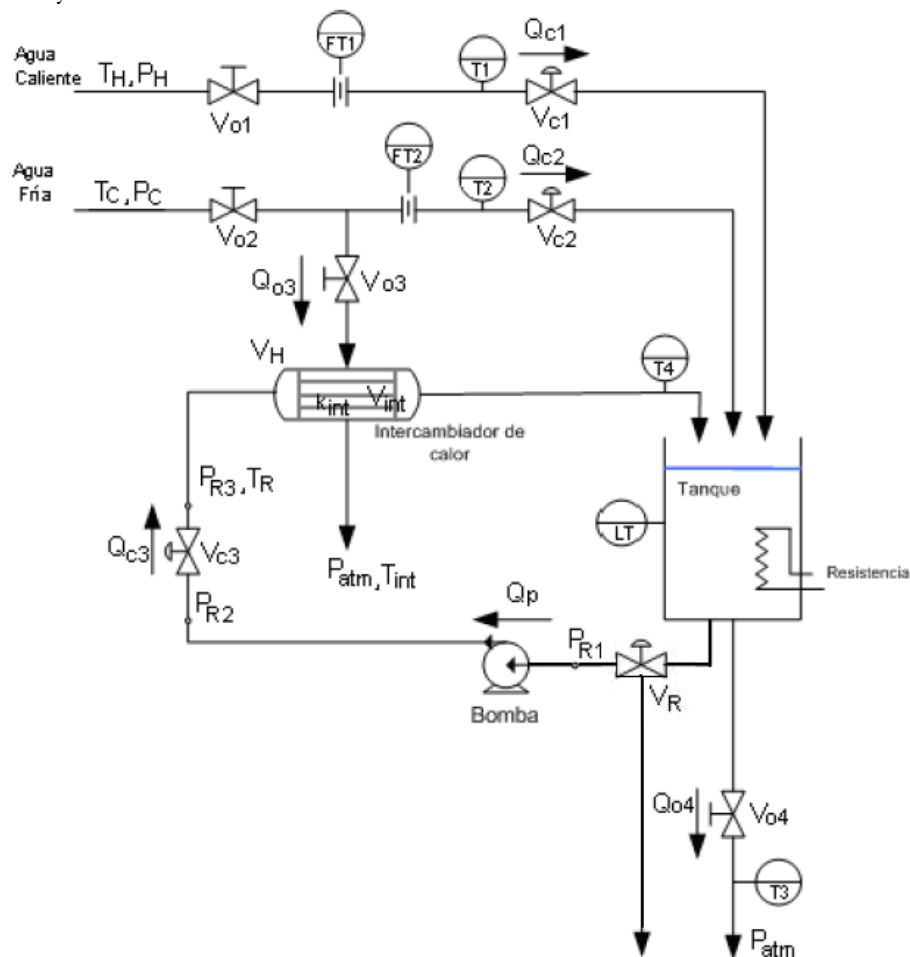
A continuación se expone una guía de ayudar a las personas que quieran familiarizarse con la planta y su manejo.

Se detalla el conexionado de los componentes (planta física) y la utilización del software necesario (Matlab y DCS ABB 800xA).

8.1 Planta Actual

8.1.1 Descripción

La planta consta de un tanque, un intercambiador de calor, una bomba, tres válvulas de control, una válvula ON-OFF de 3 vías, dos caudalímetros, cuatro sensores de temperatura y un sensor de nivel. Dichos componentes están interconectados por medio de cañerías y accesorios.



Esquema de la planta actual

El tanque es un recipiente abierto, con una resistencia interna de 1200W. En dicho tanque ingresan agua fría, agua caliente y agua tibia (proveniente de la recirculación).

El agua fría proviene directamente de OSE, por lo tanto la presión a la que es suministrada está determinada por dicha institución en aproximadamente 3bar. Por otro lado el agua caliente es agua proveniente de OSE calentada en un termotanque a 70°C

aproximadamente. El agua tibia es agua que sale del tanque y es enfriada mediante el intercambiador de calor por el cual circula agua fría con este propósito, también proveniente de la red de OSE. Dicha recirculación se realiza mediante la bomba.

Los tres caudales que ingresan al tanque son controlados por las válvulas de control correspondientes a cada uno de ellos. El caudal de agua caliente se controla con la servo-válvula Vc1, el frío con Vc2 y la recirculación con Vc3.

Además de las válvulas de control existe en la planta una válvula ON-OFF (VR) de 3 vías encargada de habilitar la recirculación (si la bomba está encendida) o permitir el desagotar del tanque.

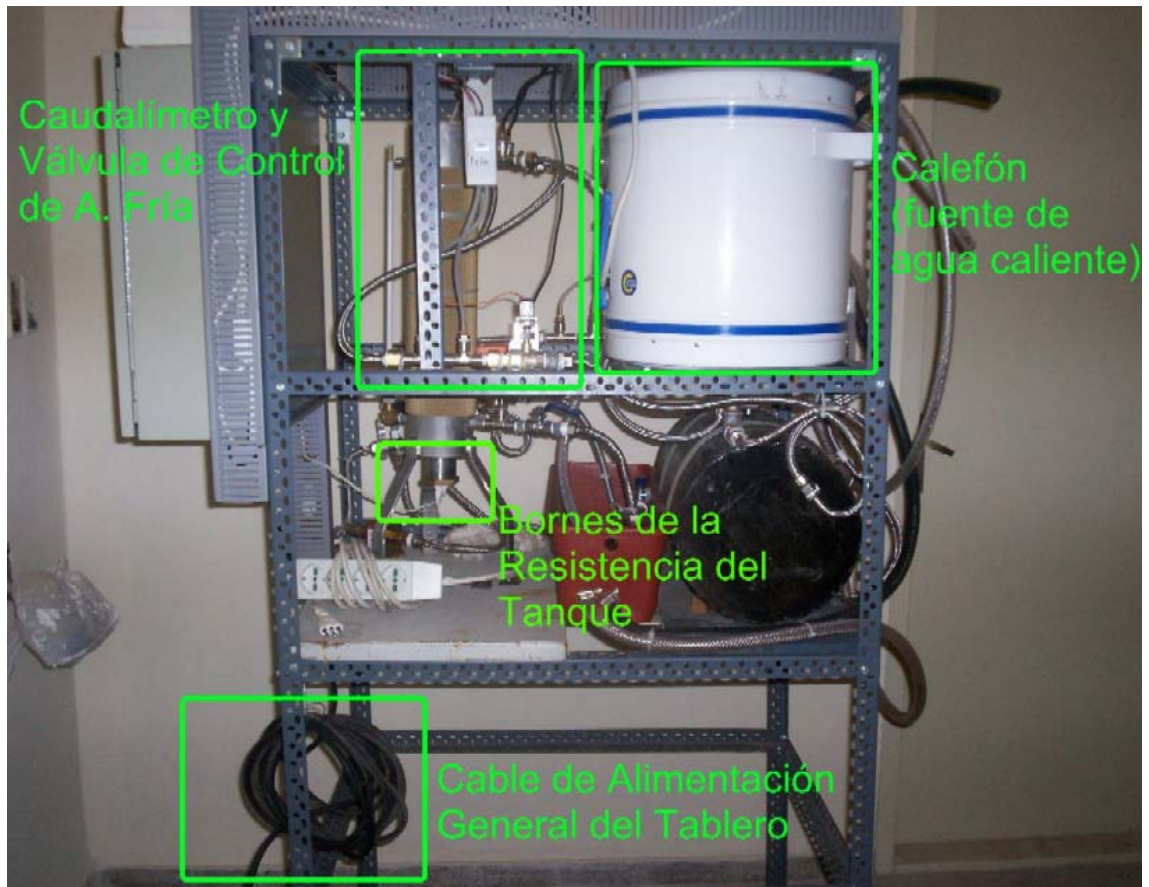
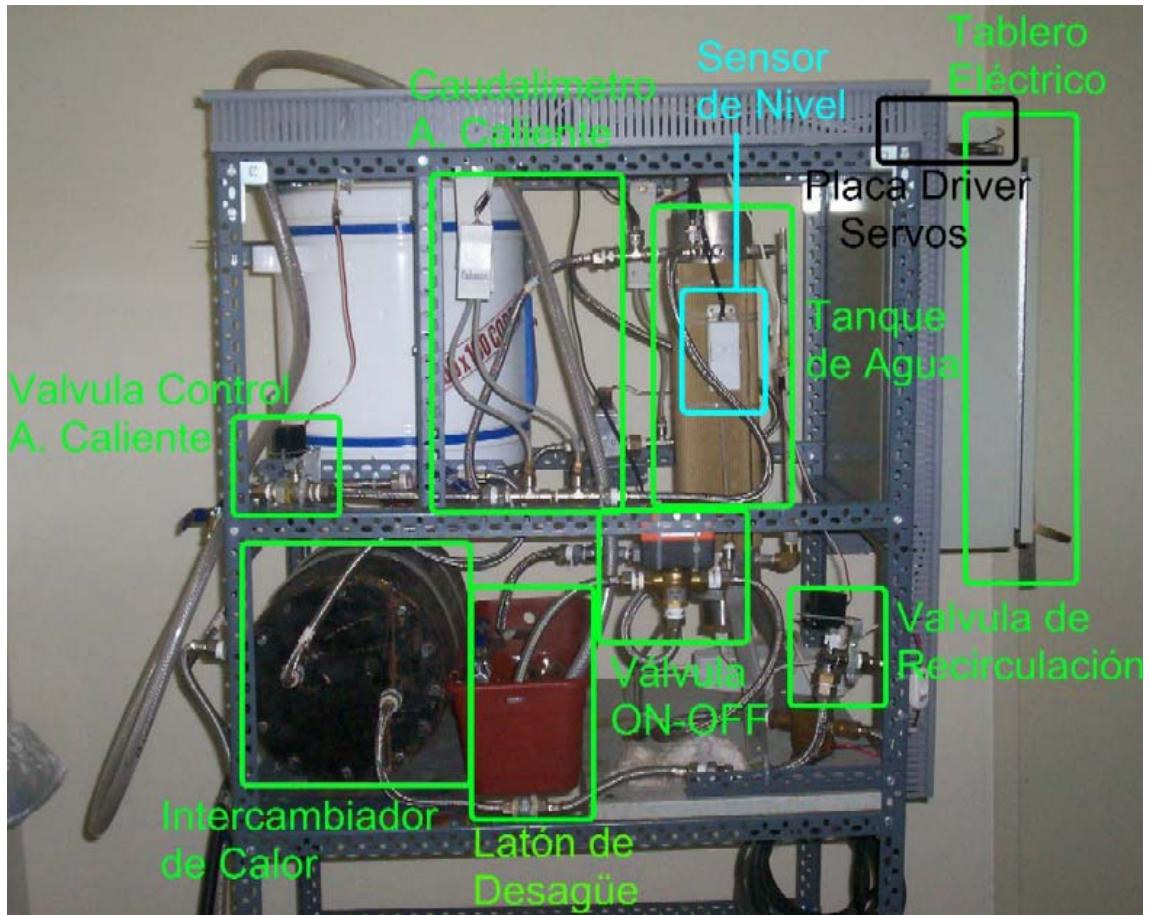
Las válvulas manuales Vo1 y Vo2 se utilizan para limitar los caudales máximos de agua caliente y agua fría respectivamente (éstas no son válvulas de control). El pasaje de caudal a través de la carcasa del intercambiador de calor se habilita mediante la apertura de la válvula Vo3 y finalmente la salida del tanque es manejada por una válvula (Vo4). Existe también una válvula manual general de entrada a la planta, la cual sirve para limitar el caudal máximo de entrada.

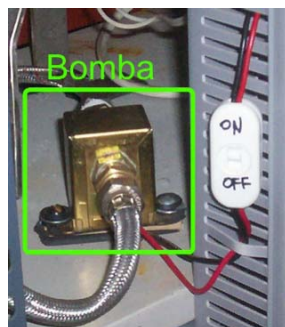
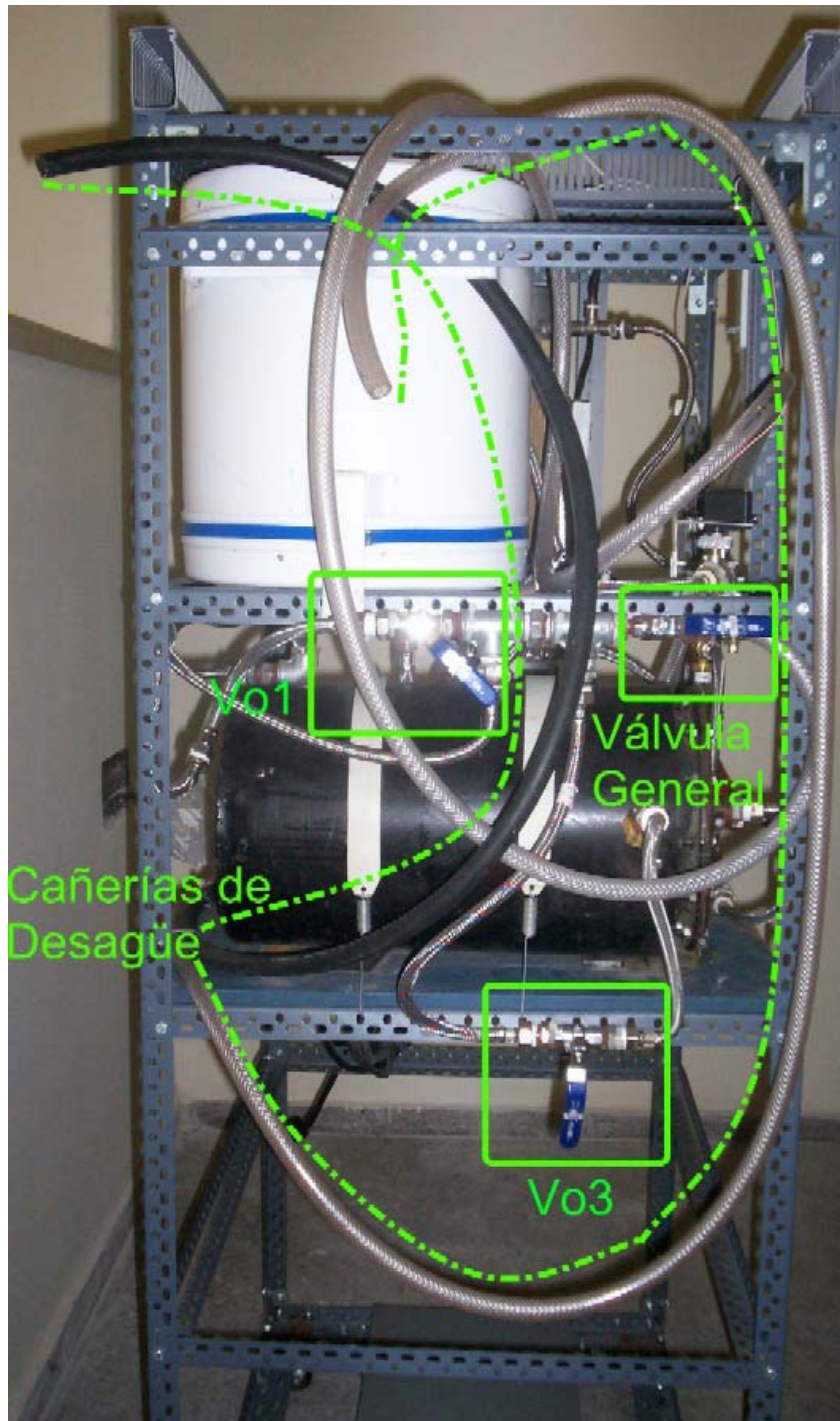
En lo que se refiere a los sensores dispuestos a lo largo del circuito de circulación de agua existen cuatro sensores de temperatura para medir las temperaturas de agua caliente (T1), agua fría (T2), agua enfriada por el intercambiador de calor (T4) y salida del tanque (T3). Además dos caudalímetros miden el caudal de agua caliente (FT1) y fría (FT2) que ingresan al tanque. Para medir el nivel de agua dentro del tanque hay un sensor de nivel (LT).

Los actuadores (servos) así como los sensores (RTD, caudalímetros, sensor de nivel) están conectados a un PLC, el cual se encarga de leer las señales, transferirlas al sistema de control y escribir en los actuadores el resultado del procesamiento.

El procesamiento de estas señales se realiza en Matlab y se utiliza el DCS 800xA de ABB como interfaz de comunicación entre PLC-Matlab, además de ser el encargado de proveer una interfaz amigable y almacenar los históricos de los experimentos realizados.

A continuación se presentan fotos con detalles de la mayoría de las componentes de la planta.





Bomba con habilitación manual



Detalle del extremo de la entrada general de agua

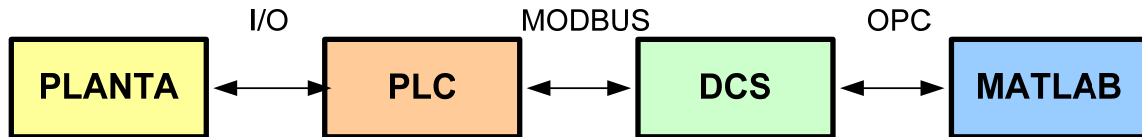
8.1.2 Pasos a seguir para iniciar la planta

- 1) Conectar la manguera de alimentación (la que tiene rosca) de agua a la red de OSE y colocar las dos mangueras (una negra y la otra transparente) de desagüe de manera tal que el agua desagüe.
- 2) Abrir completamente la válvula Vo4.
- 3) Abra el tablero de la planta.
- 4) Conectar el alargue negro al tomacorrientes, cerciorarse de que estén todas las llaves de protección apagadas.
- 5) Realizar la conexión del PC con el PLC (cable de comunicación serial negro)
- 6) Conecte el PC a la corriente, mediante el alargue conectándolo al tomacorriente Reserva
- 7) Levantar la llave diferencial
- 8) Levantar la llave PLC
- 9) Levantar la llave Calefón (si se va a usar agua caliente)
- 10) Levantar la llave Reserva
- 11) No levante la llave “Válvulas y Sensores” hasta que se le indique (en este momento no se sabe que datos tienen cargados el PLC, por lo que se podrían dañar los actuadores)
- 12) Inicie el PC y loggése con nombre de usuario: ITC y password: itc
- 13) Levante el workplace (ver detalles (más adelante) en el punto 8.3.1)
- 14) Espere que Matlab se inicialice.
- 15) Compruebe que los datos de calibración estén cargados y en caso contrario deberá realizarlo.
- 16) Abrir la canilla de alimentación.
- 17) Abra la válvula de entrada general de la planta.
- 18) Levante la llave “Válvulas y Sensores”
- 19) Realice la calibración de la planta de ser necesario.
- 20) Ahora se está en condiciones de empezar a trabajar con el sistema de control.

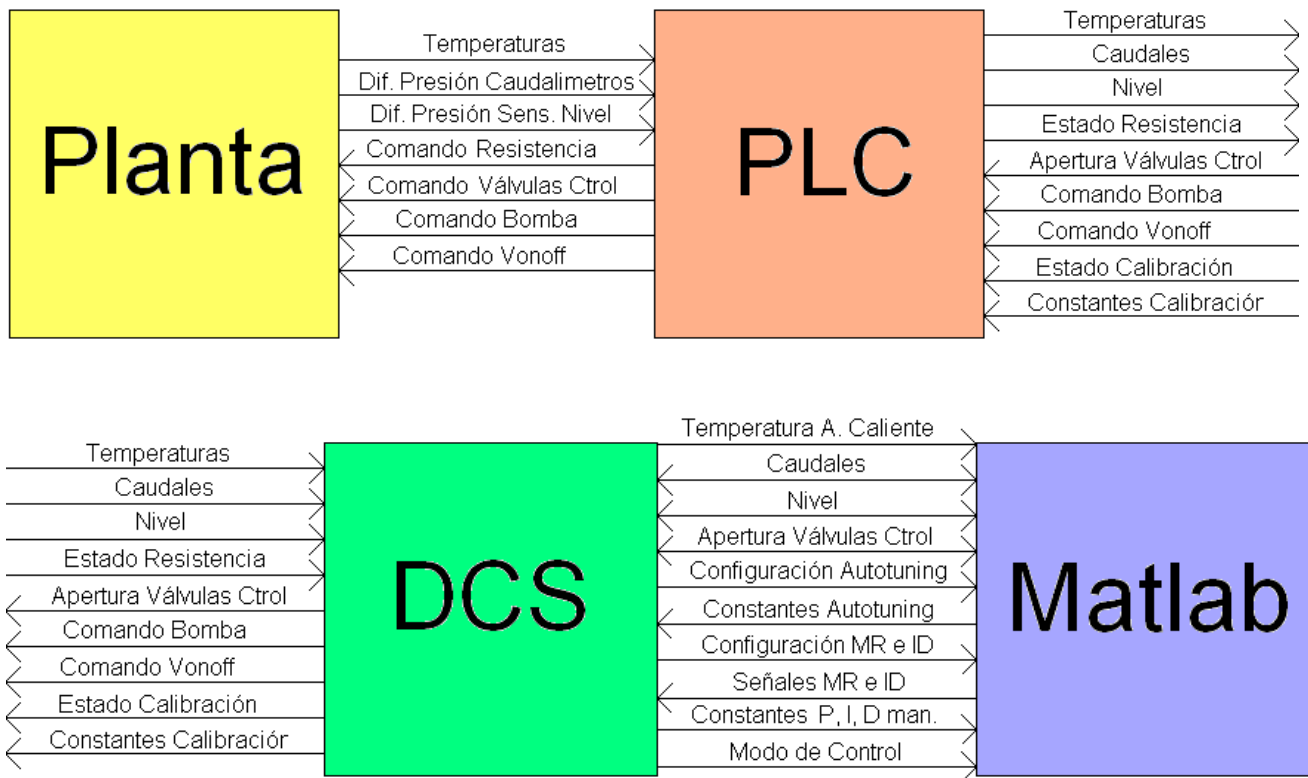
8.2 Comunicación de datos Planta-Matlab

En ésta sección se explican los pasos necesarios para pasar la información de los sensores de la planta a Matlab y cómo hacer para que las órdenes de Matlab lleguen al PLC (por detalles ver el capítulo 3 de éste informe).

El sistema consta de diferentes partes:



Entrando en mayor detalle, en cada interfaz se intercambia la información indicada en el siguiente esquema:



8.2.1 Comunicación Planta → Matlab

Planta:

En la planta se encuentran los sensores (Temperaturas, Nivel, Caudales) que envían la información al PLC mediante señales de voltaje o corriente.

I/O:

Los detalles de la implementación del pasaje de la información desde los sensores al PLC se encuentran en la documentación y en los manuales de los sensores y PLC.

PLC:

La información analógica proveniente de los sensores se obtiene en la forma de voltaje o corriente. Estos voltajes o corrientes son recogidos por las entradas analógicas del PLC. El PLC digitaliza esta información convirtiéndola en valores numéricos, en particular el PLC utilizado digitaliza estos valores analógicos en enteros de 16 bits con signo. En este proyecto se usaron entradas de voltaje de 0 a 10V y entradas de PT100, del manual del PLC se desprende las siguientes relaciones:

Para entradas de voltaje 0 a 10V (lecturas de sensores diferenciales de presión):
0V → 0 cuentas
10V → 32767 cuentas

Para entradas de PT100:
0 °C → 0 cuentas
200 °C → 2000 cuentas

La relación entre caudal y voltaje del sensor diferencial de presión no es lineal sino que es cuadrática, además la medida puede estar afectada por un offset y el máximo de la señal de entrada puede y va a ser menor a 10V. Esto es el motivo del por qué no se pasa directamente la señal proveniente de los sensores de caudal al sistema DCS y se realizan algoritmos de adaptación de señal en el PLC de manera de obtener una relación lineal.

En este proyecto se decidió pasar la información de los sensores al DCS con la siguiente relación (lineal):

Nivel:
0% → 0 cuentas
100% → 1000 cuentas

Caudal:
0 l/min → 0 cuentas
Max l/min → 100 cuentas

Temperatura: (por defecto)
0 °C → 0 cuentas
200 °C → 2000 cuentas

Esta información, una vez obtenida de los sensores, queda almacenada en variables del PLC y enseguida es transmitida por Modbus al DCS.

MODBUS:

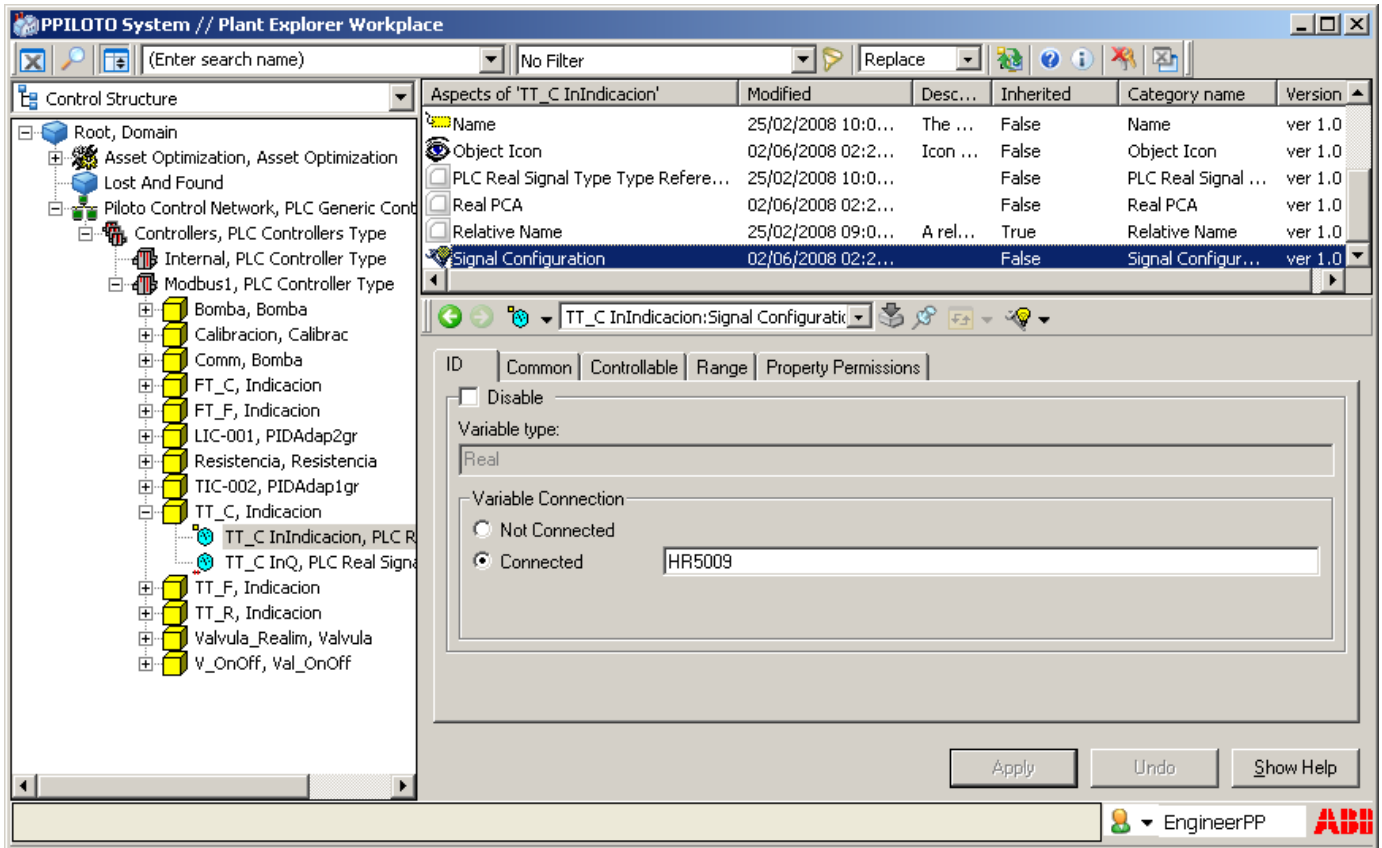
Es un protocolo de comunicación muy difundido en la industria y es el utilizado en este proyecto para implementar la comunicación en el PLC y el DCS ABB.

MODBUS es un protocolo maestro-esclavo. Se decidió que el PLC sea el esclavo ya que el DCS de ABB solo tiene la posibilidad de ser MODBUS maestro. Por defecto el PLC es un esclavo MODBUS, pero cabe la posibilidad de que sea maestro si se lo configura de tal forma.

El detalle del direccionamiento de las variables del PLC desde el DCS se encuentra en la sección 3.4.

DCS ABB:

La principal tarea del DCS es ser el nexo entre el PLC y Matlab. Como se describió antes, el DCS es maestro Modbus (su configuración se encuentra en la sección 3.4.2 de este informe). En el DCS se deben escribir las direcciones MODBUS de las variables del PLC para ser leídas por el DCS, debajo se muestra una figura con la configuración de ejemplo.



También se debe hacer el escalado de las variables en el sistema DCS (señales provenientes de los sensores después de ser linealizadas):

Nivel:

0% → 0 cuentas

100% → 1000 cuentas

Caudales:

0 l/min → 0 cuentas

Max l/min → 100 cuentas

Temperaturas:

0 °C → 0 cuentas

200 °C → 2000 cuentas

OPC:

El DCS se comunica con Matlab mediante OPC. El DCS ABB tiene un OPC Server embebido, lo que genera que toda información que ingresa a éste, de algún modo, queda disponible mediante OPC. Por lo tanto la información obtenida mediante Modbus de las señales analógicas del PLC se encuentra disponible por medio de OPC.

Matlab:

Configurando Matlab como OPC cliente del OPC Server del DCS y direccionando las variables del DCS que contienen la información proveniente del PLC de las señales analógicas linealizadas, se tiene la información disponible de los sensores en Matlab. De

esta forma se puede realizar cualquier algoritmo de control que se desee. El detalle de las configuraciones se encuentra en la sección 3.5 de este documento y en la ayuda de Matlab.

8.2.2 Comunicación Matlab → Planta

Matlab:

Una vez que el algoritmo de control en Matlab genera la salida se debe comunicar este valor al actuador de correspondiente (ej. Válvula agua fría). El primer paso para esto es pasar este valor por OPC a una variable del DCS designada para la comunicación con el PLC mediante Modbus.

En este proyecto se decidió que la salida de los algoritmos de control (apertura de las válvulas) estuvieran en el rango de de 0 a 100 %.

OPC:

El valor de la salida del algoritmo de control se pasa mediante OPC al DCS ABB.

DCS ABB:

El DCS ABB obtiene el valor de la salida del algoritmo implementado en Matlab mediante comunicación OPC, este valor debe ser escalado en el DCS para luego ser pasado al PLC en cuentas y no en unidades de ingeniería. Para que la comunicación OPC sea capaz de escribir el valor en la variable del DCS se debe tener seleccionado habilitado el atributo de que indica que la variable es controlable ('Is controllable').

En este proyecto el escalado para las salidas de los algoritmos de control se realizó de la siguiente manera:

$$\begin{aligned} 0\% &\rightarrow 0 \text{ cuentas} \\ 100\% &\rightarrow 100 \text{ cuentas} \end{aligned}$$

MODBUS:

La información del DCS es enviada, mediante MODBUS, a la posición de memoria correspondiente en el PLC. Cuando la variable del DCS es controlable entonces éste sabe que el contenido de la variable son datos a escribirse al PLC.

PLC:

Para el caso de las señales de salidas de los algoritmos de control implementados el valor de proveniente del DCS es un entero de en el rango de 0 a 100 cuentas. Este valor es pasado por un bloque FKG para generar el valor en voltaje para comandar la válvula. Los detalles de implementación se encuentran en la sección 3.3.2.4 .

I/O:

La orden de apertura de las válvulas es re-enviada desde el PLC, en voltaje, a la placa que maneja los servos. La implementación se encuentra en detalle en la secciones 2.7.2 y 2.7.3.

Planta:

La orden de apertura llega a los servos, que comandan las válvulas, desde la placa driver de servos.

Conclusiones:

Se describieron a grandes rasgos los pasos para poder realizar el pasaje de datos desde la planta a Matlab y viceversa.

8.3 DCS

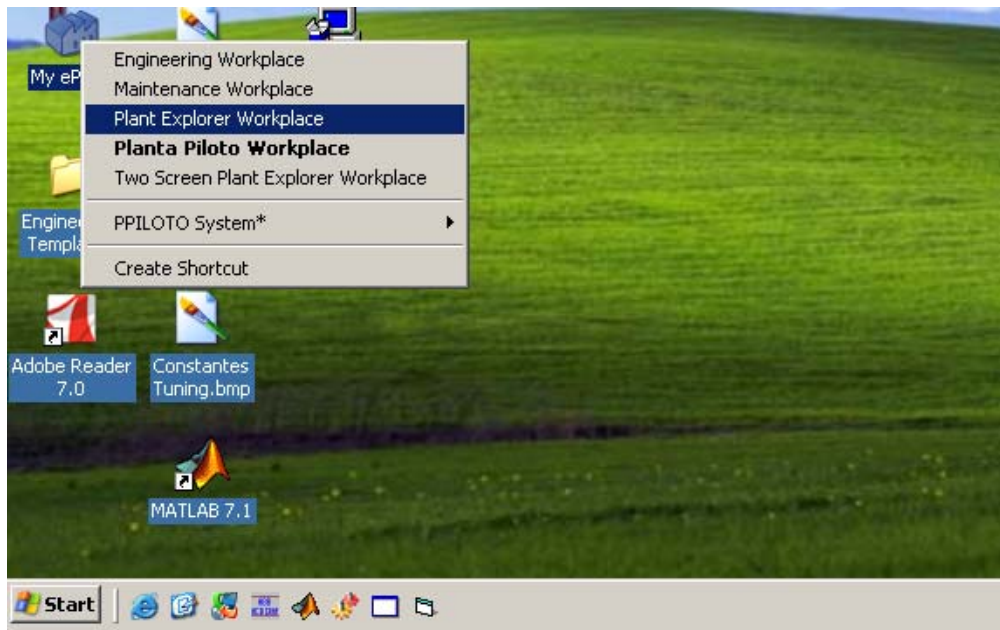
La PC tiene instalado el sistema operativo XP SP2, con el cual todos los usuarios deberían estar familiarizados ya que es uno de los sistemas operativos más usados por el común de las personas. Al prender la PC aparece la opción de loggearse con tres usuarios:

- 1.-engineer
 - a.-Contraseña: engineer
 - b.-Descripción: Este usuario fue creado para correr los servicios del sistema DCS ABB y se recomienda fuertemente no hacer ningún tipo de cambios en el mismo.
- 2.-EngineerPP
 - a.-Contraseña: engineerpp
 - b.-Descripción: Con este usuario se debe hacer la ingeniería del sistema, armar pantallas, hacer cambios en los objetos, configura Matlab, etc.
- 3.-ITC
 - a.-Contraseña: itc
 - b.-Con este usuario los estudiantes se loggearán para trabajar con el sistema de control, estando limitado en sus funciones de sólo operación.

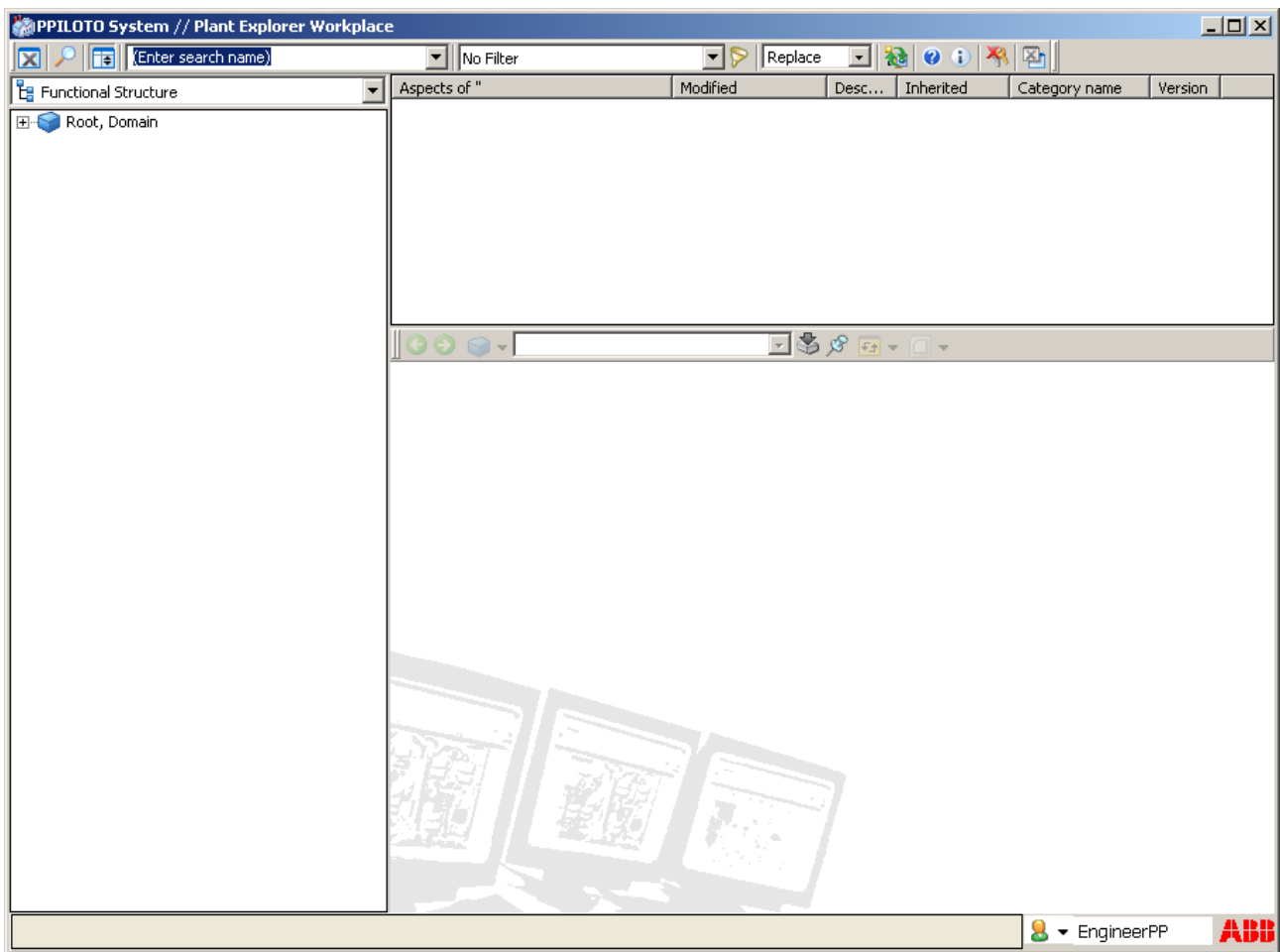
De aquí en más se va describir cómo usar el sistema para el usuario de ingeniería, ya que este requiere de más pasos que el usuario ITC. Para el usuario ITC el procedimiento es muy parecido e intuitivo.

8.3.1 Como ingresar al sistema de control

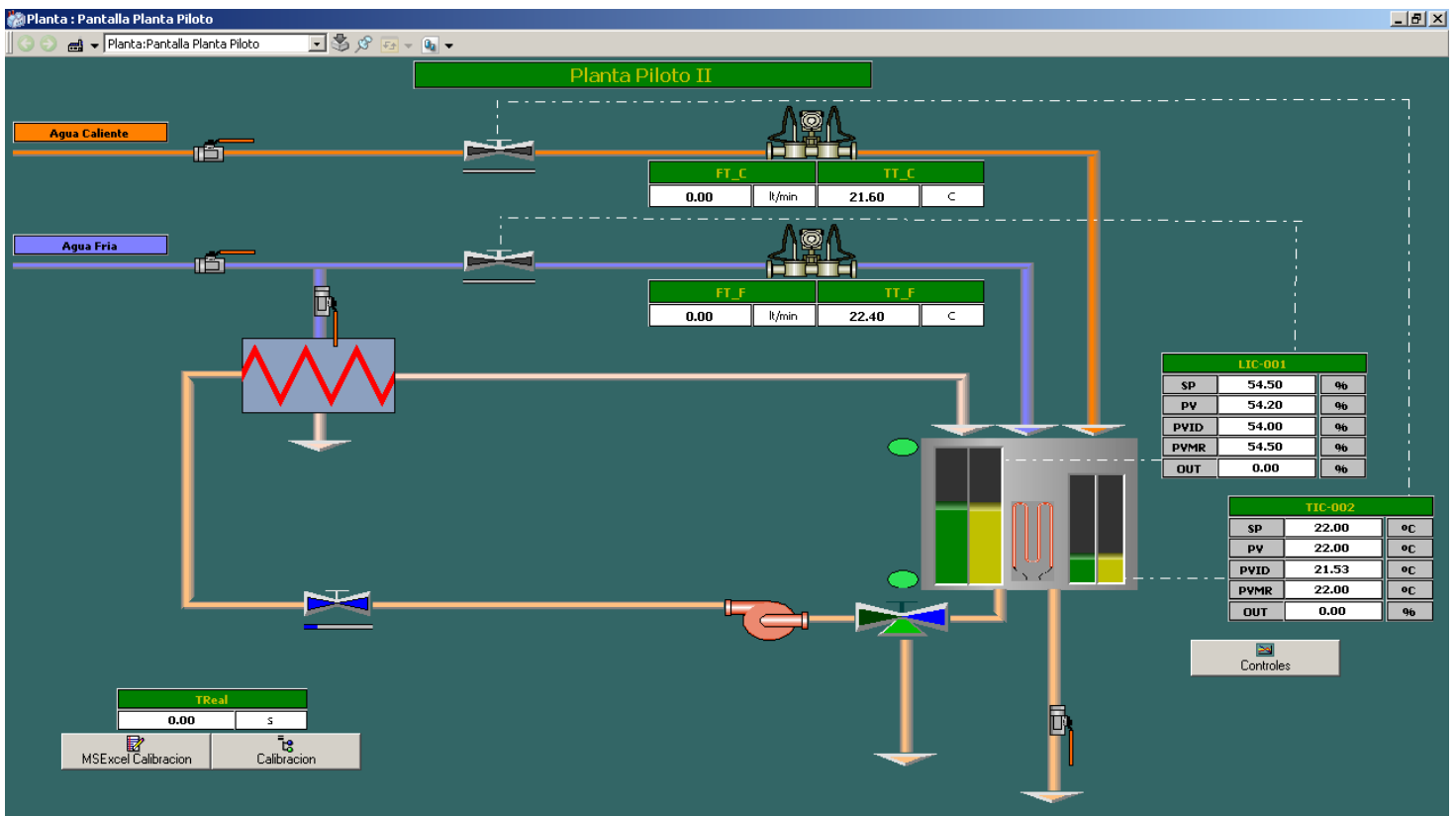
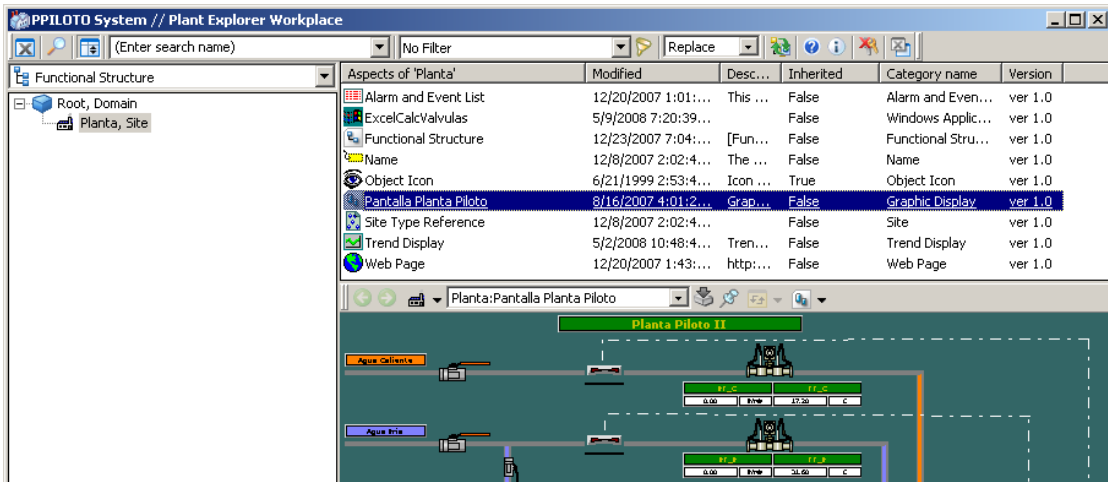
- ✓ Una vez prendida la PC el sistema comienza, solo, a cargar los servicios que necesita para funcionar, esto es así porque se seleccionó la opción de que comenzara a correr durante el inicio de la PC.
- ✓ El usuario se puede loggear como usuario ITC o EngineerPP. En el caso que se logee como ITC, la pantalla principal se abrirá ocupando todo el escritorio y posibilitando el uso del sistema de control en modo operación, también el programa de Matlab comenzará a correr en forma automática. Si se selecciona el usuario EngineerPP, se debe presionar botón derecho sobre el icono con forma de fábrica y seleccionar la opción de Plant Explorer.



Luego de haber hecho lo anterior se abrirá la siguiente ventana:



Una vez dentro del Plant Explorer se debe abrir la pantalla principal. Esto se hace dando doble click sobre el aspecto Root→Planta→Pantalla Planta Piloto (ver a la izquierda en la siguiente figura)



La figura anterior es la pantalla principal, en la misma hay íconos animados, por ejemplo hay niveles en el tanque que cambian su altura de acuerdo a valores de proceso. Los íconos además de ser animados también son un acceso directo a los Faceplates de las distintas unidades de control, por ejemplo: válvula ON-OFF de tres vías, Bomba, controles de nivel y temperatura, válvula de realimentación.

En la parte inferior izquierda hay dos botones. Uno abre una planilla de Excel que sirve para cargar automáticamente las constantes obtenidas de las calibraciones. El otro botón abre un faceplate especialmente creado para hacer dichas calibraciones.

8.3.2 Uso de los Controladores

A continuación se explica el funcionamiento de los controladores y sus faceplates, además se aprovechará la ocasión para explicar algún detalle de utilización del DCS.

8.3.2.1 Objeto Controlador (LIC o TIC)

Como el controlador de temperatura es prácticamente igual al de nivel (sólo se diferencian en que la ID y el MR del TIC son de primer orden mientras que en el LIC son de segundo orden) sólo se discutirá éste último.

Los “objetos hijos”: ‘LIC-001 PVM’, ‘LIC-001 PV’ y ‘LIC-001 u’ son los tres únicos objetos del controlador que interactúan con el PLC mediante Modbus y con Matlab mediante comunicación OPC. El resto de los “objetos hijos” del “objeto padre” LIC interactúan solamente con Matlab.

8.3.2.2 Faceplates de los Controladores

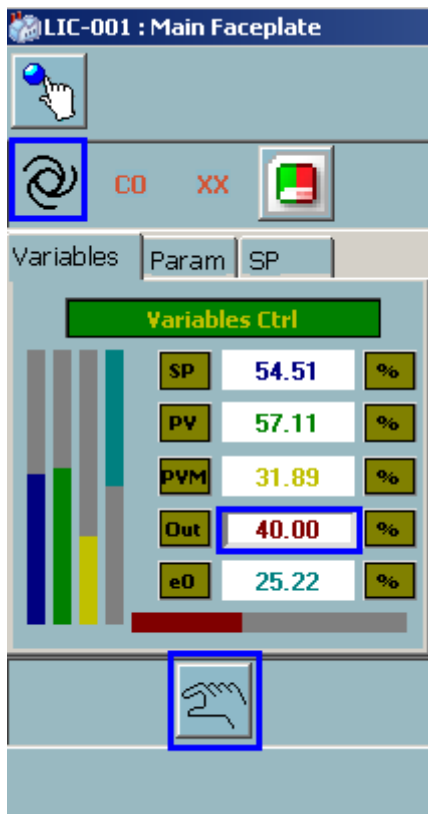
Para que la interfaz del usuario con el controlador diseñado sea amigable se creó el siguiente Faceplate:



Los controladores de nivel (LIC-001) y temperatura (TIC-002) tienen tres modos de funcionamiento: Manual, PID Automático y Modelo de Referencia, además de la posibilidad de habilitar la Identificación de la planta. Estos modos de funcionamiento son explicados, en detalle en el punto a continuación.

8.3.2.3 Modos de funcionamiento del Controlador

8.3.2.3.1 Modo Manual



En el modo manual se maneja directamente la salida del control, o sea que el usuario elije la apertura de la válvula a su juicio.

En parte superior izquierda de la figura de al lado se recuadró, en azul, un remolino. Este recuadro indica en qué modo se encuentra el control, en este caso es en modo automático.

En la parte inferior de la figura se recuadró una mano gris. Este botón habilita la opción de modo manual, el mismo aparece de ese color debido a que la opción de manual no está elegida y por lo tanto se puede elegir. Si el control estuviera en modo manual este botón tendría la mano dibujada en gris indicando que se está en modo manual y por lo tanto no se puede volver a seleccionar.

A la derecha hay cinco valores numéricos:

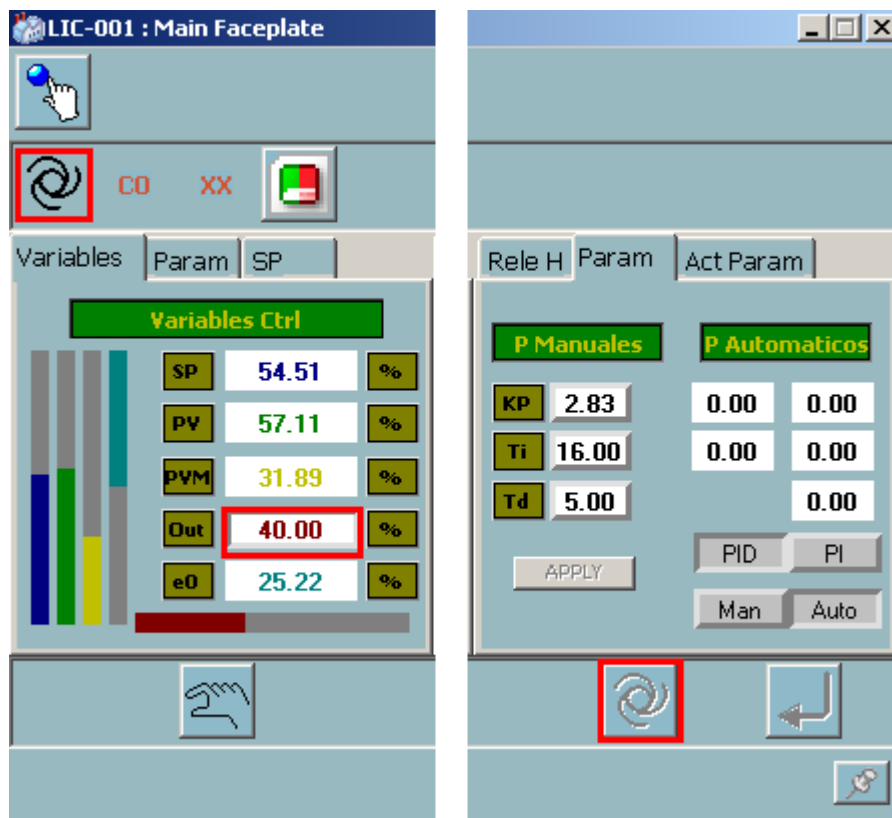
- 1- SP – Setpoint, este el valor de referencia que se desea como valor de proceso
- 2- PV – Valor de proceso
- 3- Out – Es el valor de salida del PID en forma automática, se debe notar que en modo manual la salida Out sigue a MOut (Bumpless)
- 4- MOut – Salida en modo manual, se debe notar que en modo automático no es posible cambiar manualmente este campo
- 5- E0 – Es el error a la entrada utilizado por el MR.

8.3.2.3.2 Modo PID Automático

El modo automático maneja la salida del control a través del algoritmo de PID programado, o sea que el usuario maneja el SP de proceso que desea y no se utiliza la salida de control.

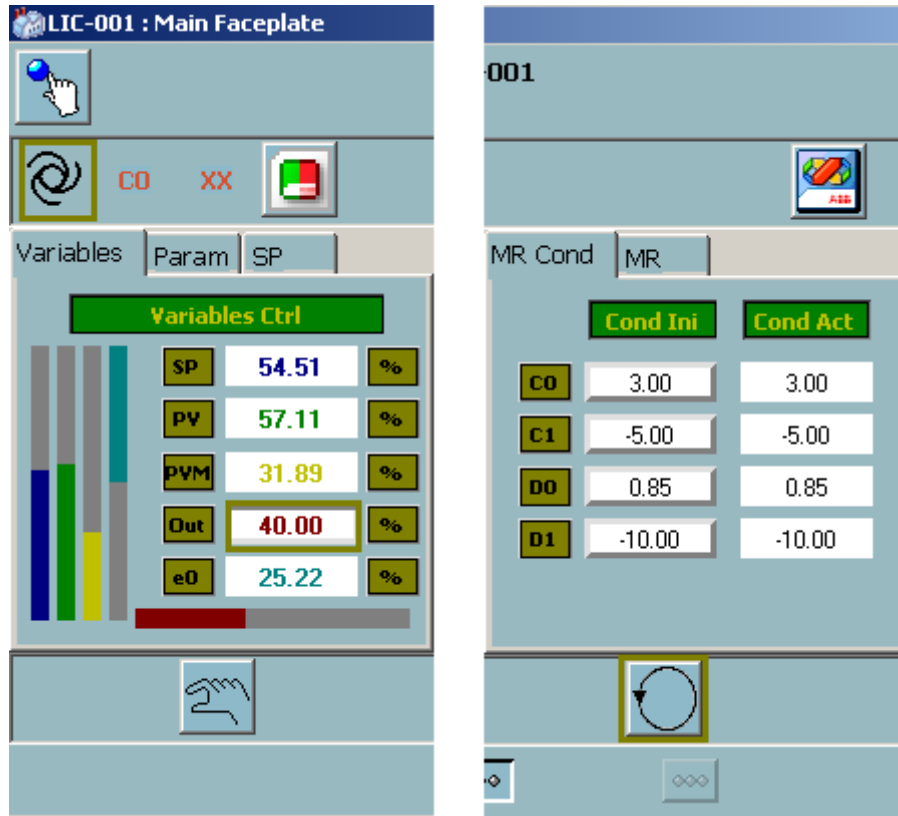
En la parte superior izquierda de la siguiente figura está encerrado, en un recuadro rojo, un dibujo de un remolino. El mismo indica que el control está en modo automático. En la parte de inferior izquierda de la figura se encuentra un remolino sin color, este botón habilita la opción de modo automático. El remolino aparece en ese color debido a que la opción de automático está elegida y por lo tanto no se puede volver a elegir. Si el control estuviera en otro modo, como por ejemplo manual, este botón tendría el remolino dibujado en color negro indicando que el control no está en modo automático y que se puede seleccionar.

En el centro de la figura aparece un número recuadrado en rojo, este número muestra el valor de la salida de control en cada instante del algoritmo.



8.3.2.3.3 Modo Control por MR

El modo MR maneja directamente la salida del control, o sea que el Modelo de Referencia maneja por su cuenta la apertura de la válvula.



En la parte superior izquierda de la figura de arriba, aparece un remolino recuadrado. Este recuadro indica en qué modo se encuentra el controlador, en este caso es en automático. En la parte de inferior izquierda de la figura se encuentra un círculo en negro, este es el botón que habilita la opción de modo Modelo de Referencia, la misma aparece en ese color ya que la opción de Modelo de referencia no se encuentra elegida y por lo tanto se puede elegir.

Cuando funciona el MR y se están adaptando los parámetros las indicaciones (en la parte superior del faceplate) 'CO' y 'XX' aparecen en verde, cuando 'CO' no se adapta aparece en rojo y cuando ningún parámetro se actualiza 'XX' también lo hace. En el centro de la figura se recuadró el valor de la salida del control.

Variables	Param	SP
Ctrl MR-ID		
g		0.0060
eMin		0.5000
g1		0.5000
Ctrl MR		
CO_Min		0.5000
Lambda		4.0000

En la figura de la izquierda se muestra donde se configuran los parámetros 'g', 'error mínimo' y 'g1'. Estos parámetros son compartidos por los algoritmos de Modelo de Referencia e Identificación.

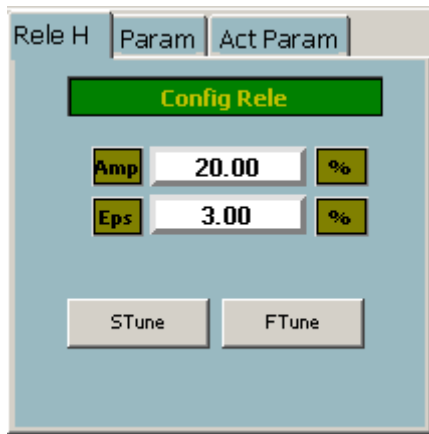
Debajo aparecen los parámetros 'CO mínimo' y 'lambda', los cuales son de uso exclusivo del Modelo de Referencia. La explicación de estos parámetros se encuentra en el Capítulo 6 de Control Avanzado.

En las figuras siguientes se muestran las solapadas en donde se deben configurar las condiciones iniciales de los parámetros de control adaptativo, y el modelo de referencia a seguir.

MR Cond	MR
Modelo Referencia	
$aS + b$ <hr style="width: 50%; margin: auto;"/> $s^2 + cS + d$	
a	1.00
b	0.25
c	1.00
d	0.25

MR Cond	MR
Cond Ini	
CO	3.00
C1	-5.00
DO	0.85
D1	-10.00

8.3.2.4 Autotuning PID

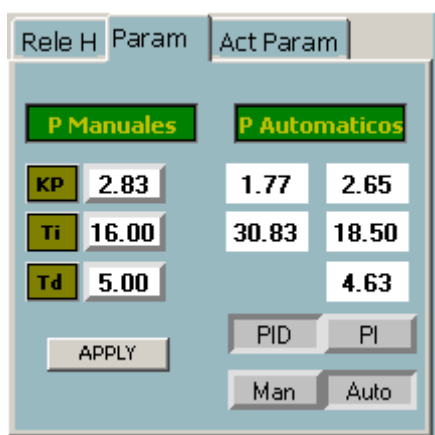


La sintonía del PID, mediante el algoritmo de relé implementado en este proyecto, se habilita y configura en la parte del faceplate del controlador mostrada en la figura a la izquierda.

La sintonía se habilita sin importar el modo de operación en el cual se encuentra el control. Una vez terminada la misma, el control vuelve al modo en el cual el controlador se encontraba antes de comenzar la sintonía.

Los parámetros que juegan en la sintonía son:

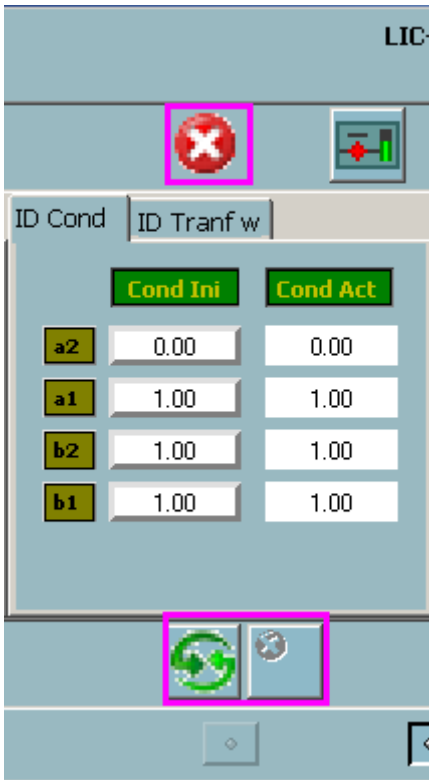
- 'Amp': amplitud pico a pico de la onda cuadrada del algoritmo.
- 'Eps': Épsilon pico a pico del algoritmo de relé
- 'STune': Se debe presionar este botón por más de un segundo para que la sintonía comience. Cuando la misma comienza aparece una indicación con forma de diagrama de Nyquist en el faceplate del Tuning.
- 'FTune': Se debe presionar este botón por más de un segundo para que la sintonía termine.



En la figura a la izquierda se muestra la parte del faceplate en donde se configuran las constantes del PID. Los botones 'PID' y 'PI' seleccionan el tipo de control a usar (solo se puede tener seleccionado uno pues el otro se deselecciona automáticamente). Los botones 'Man' y 'Auto' seleccionan de donde se toman las constantes para el algoritmo. Si se elige 'Auto' las constantes se toman del resultado del algoritmo de autosintonía de relé.

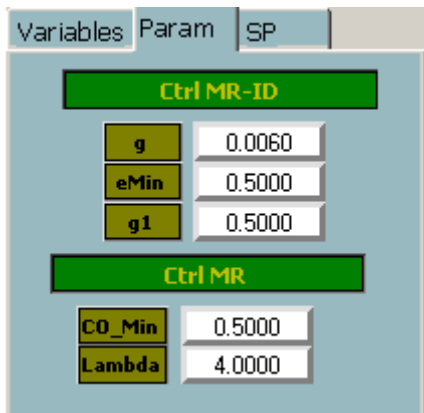
Finalmente para que la selección elegida se aplique se debe presionar el botón de 'Apply' por más de un segundo y se debe corroborar que las constantes elegidas aparezcan en la solapa 'Act Param'. El botón de 'Apply' se habilita solo si la opción elegida es correcta, si por ejemplo se selecciona control automático PID, sin antes hacer al menos una sintonía completa, el mismo permanece deshabilitado.

8.3.2.5 Identificación



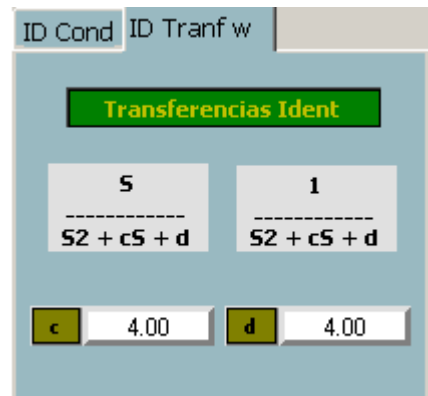
En la parte inferior de la figura de la izquierda se muestran los botones que habilitan o deshabilitan la identificación de la planta, el círculo rojo con una 'x' blanca en el interior deshabilita la identificación de la planta, mientras que las flechas verdes la habilitan. En el caso de que no se habilite la identificación, los parámetros de la misma no se actualizan.

Las condiciones iniciales del algoritmo de Identificación se encuentran en el centro de la figura a la izquierda, y a la derecha las condiciones actuales de los parámetros de la identificación.



Los parámetros 'g', 'error mínimo' y 'g1' se encuentran debajo del cartel Ctrl MR-ID. Estos son compartidos en el control por MR y en la identificación.

Las transferencias de la identificación (los identificadores, en este caso en particular la ID es de segundo orden) están en la figura de la derecha.



8.3.2.6 Botones Auxiliares

En la parte superior de los faceplates de los controladores (en las indicaciones) aparecen tres botones. El botón de la izquierda despliega las tendencias de las variables de control, el botón del centro despliega las tendencias relacionadas con el algoritmo de identificación y el botón de la derecha despliega las tendencias relacionadas del modelo de referencia.

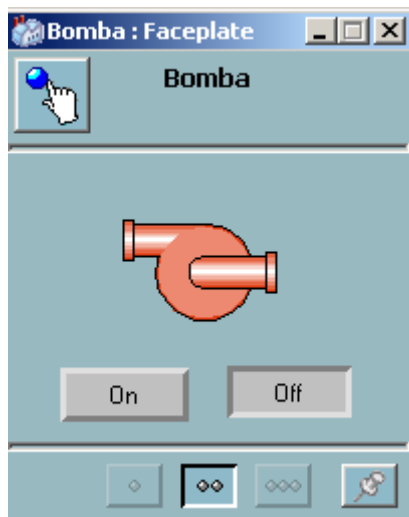


8.3.3 Otros objetos implementados

A continuación se explica el funcionamiento de los objetos Bomba, Resistencia y Válvula de Realimentación y sus respectivos Faceplates.

8.3.3.1 Bomba

El comando de la bomba es ON-OFF, o sea que la misma puede estar prendida o apagada. Para la misma se implementó un bloqueo de seguridad para que no prenda cuando por el lazo de realimentación no circula suficiente caudal, con esto se pretende que la misma no se sobre-exija .



La bomba se puede prender y apagar mediante los botones 'On' y 'Off' respectivamente. En la figura de la izquierda se muestra el faceplate de éste objeto. Cuando el botón 'Off' se encuentra presionado el dibujo de la bomba es de color rojo, esto indica la orden de parada de la misma y cuando el botón 'On' está presionado, el dibujo de la bomba es de color verde indicando que se dio la orden de marcha.

Cuando se produce algún bloqueo, automáticamente se "presiona" el botón 'Off' y se deshabilita el botón 'On' (la palabra On queda sin color). Además en la pantalla principal aparece un cartel que avisa que la bomba está bloqueada.

8.3.3.2 Resistencia

El comando de la resistencia es ON-OFF. A diferencia de la bomba ésta es controlada por el PLC y el usuario sólo tiene el control de configurar el 'duty cycle' del encendido y el período de trabajo.

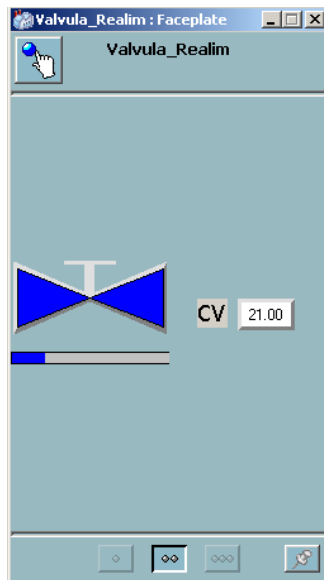
Como medida de seguridad el control de la resistencia, configurado en el PLC, comienza a funcionar cuando el nivel cubre totalmente la resistencia (nivel LT_0).



En la figura de la izquierda se muestra el faceplate de éste objeto. Cuando el botón 'On' se encuentra presionado, la resistencia se dibuja de color rojo indicando que está prendida, cuando el botón 'Off' está presionado, el dibujo de la misma es de color verde indicando que está apagada.

Los botones de este faceplate están solamente para mostrar el estado de la resistencia y no pueden ser operados.

8.3.3.3 Válvula de realimentación




Esta válvula es operada directamente por el usuario y se controla mediante el valor en el campo 'CV' que aparece en el faceplate de la izquierda. Este campo acepta valores entre 0 y 100, donde 0 significa válvula totalmente cerrada y 100 significa válvula totalmente abierta.

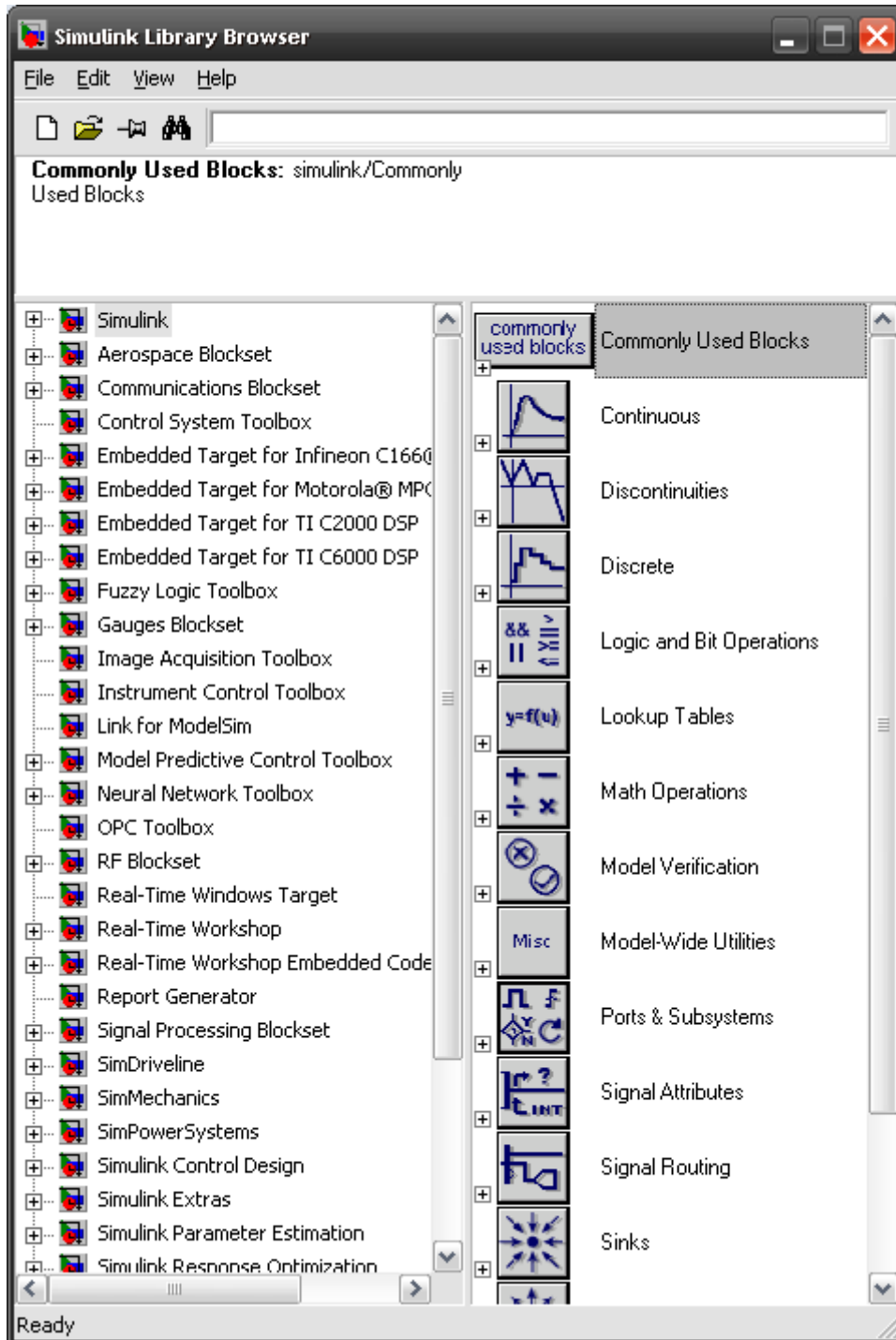
8.3.3.4 Indicaciones

Para presentar datos en pantalla de forma amigable se diseñaron objetos de indicación que presentan el valor de variables tales como caudales, temperaturas, niveles, setpoints, etc. Los mismos son muy fácilmente entendibles y no tiene sentido una explicación más detallada.

8.4 Implementación de un controlador en Simulink

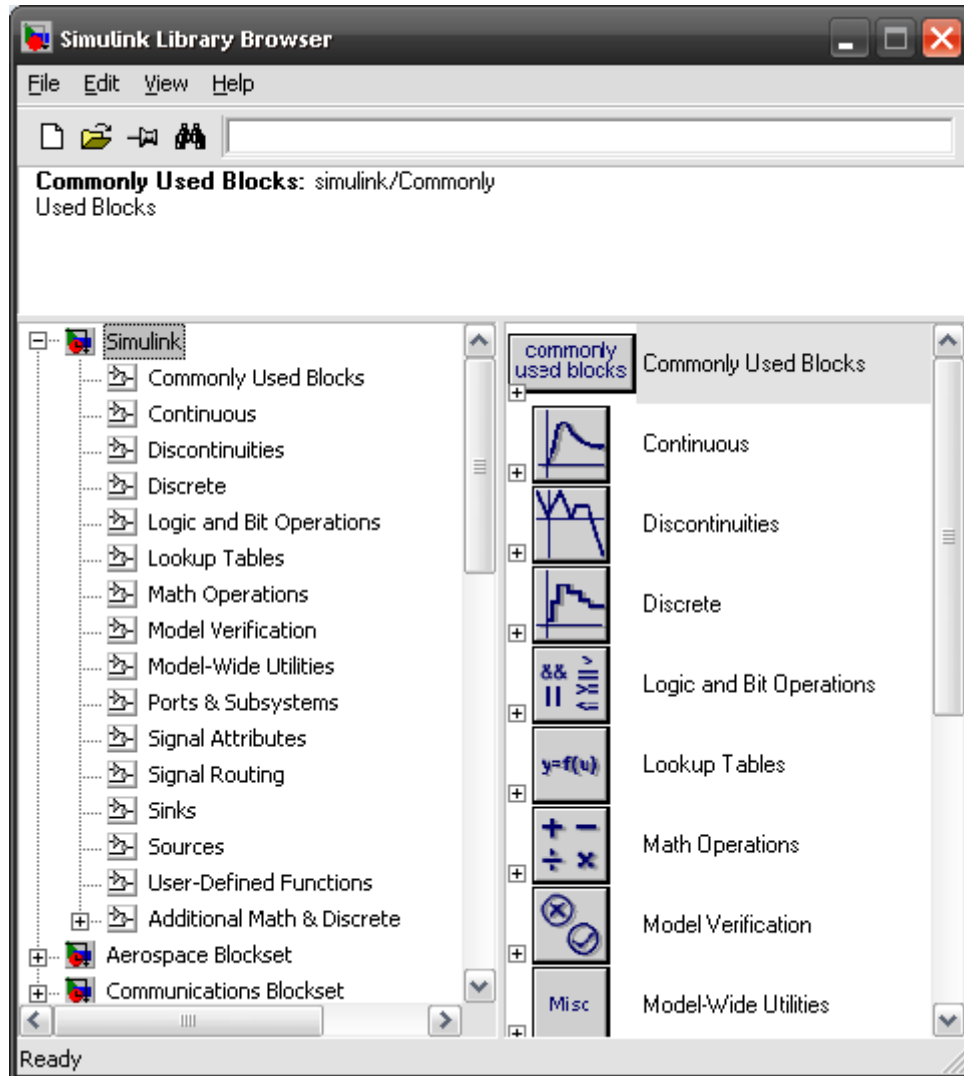
8.4.1 Ambiente de trabajo

Primero se debe ejecutar la herramienta Simulink desde la ventana de comandos de Matlab haciendo click en el ícono correspondiente: . Enseguida después emerge una ventana como la de la figura a continuación. La misma contiene todas las librerías que el entorno de Simulink bajo Matlab soporta.



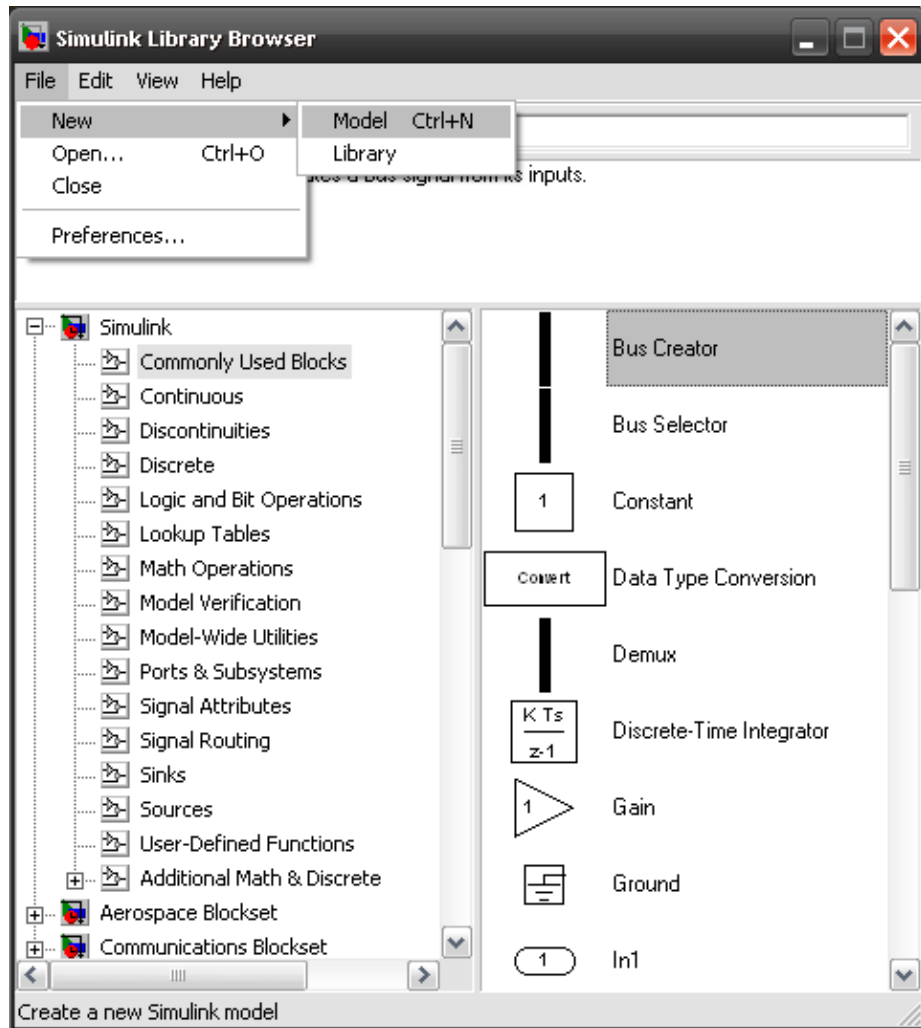
8.4.2 Ejemplo de implementación de un PI

Para este sencillo ejemplo solo se necesita la librería básica Simulink. Con un click se expande el menú 'Simulink' en la ventana anterior, y se observa la siguiente figura:

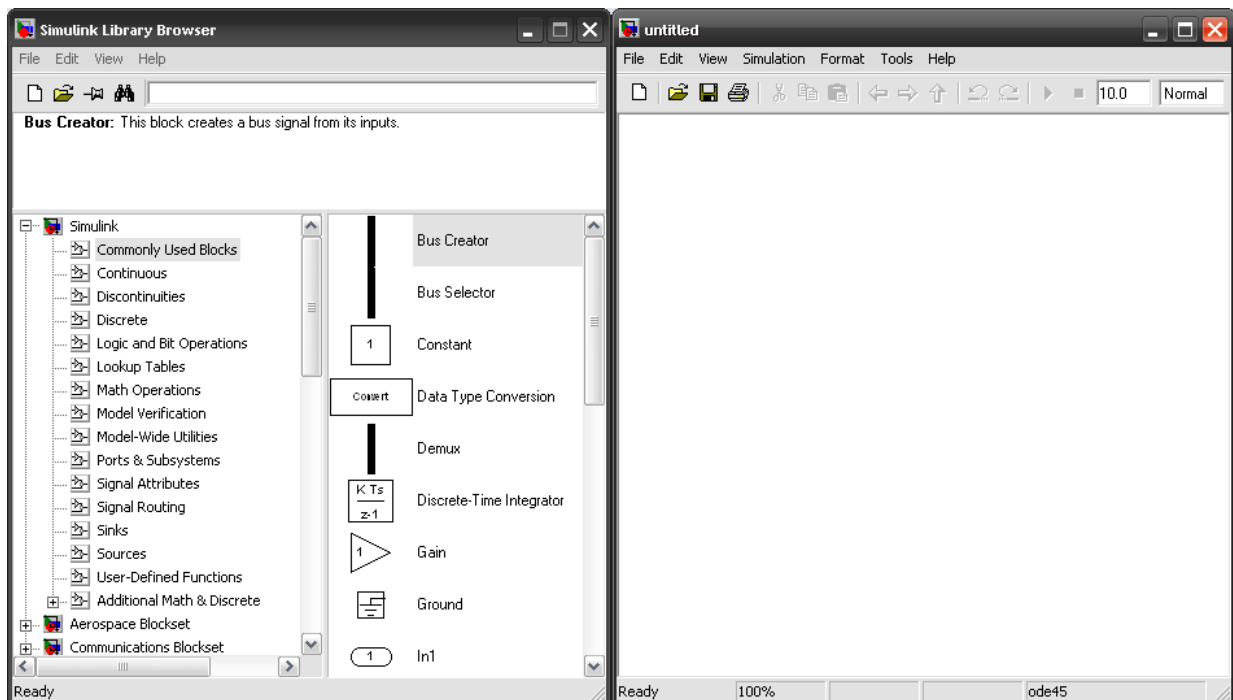


Esta ventana está dividida en dos partes. La de la izquierda es la correspondiente a las librerías y la de la derecha es el contenido de la librería seleccionada

Se elige un nuevo fichero donde se guardara el modelo: se selecciona en el menú **File**→**New**→**Model**



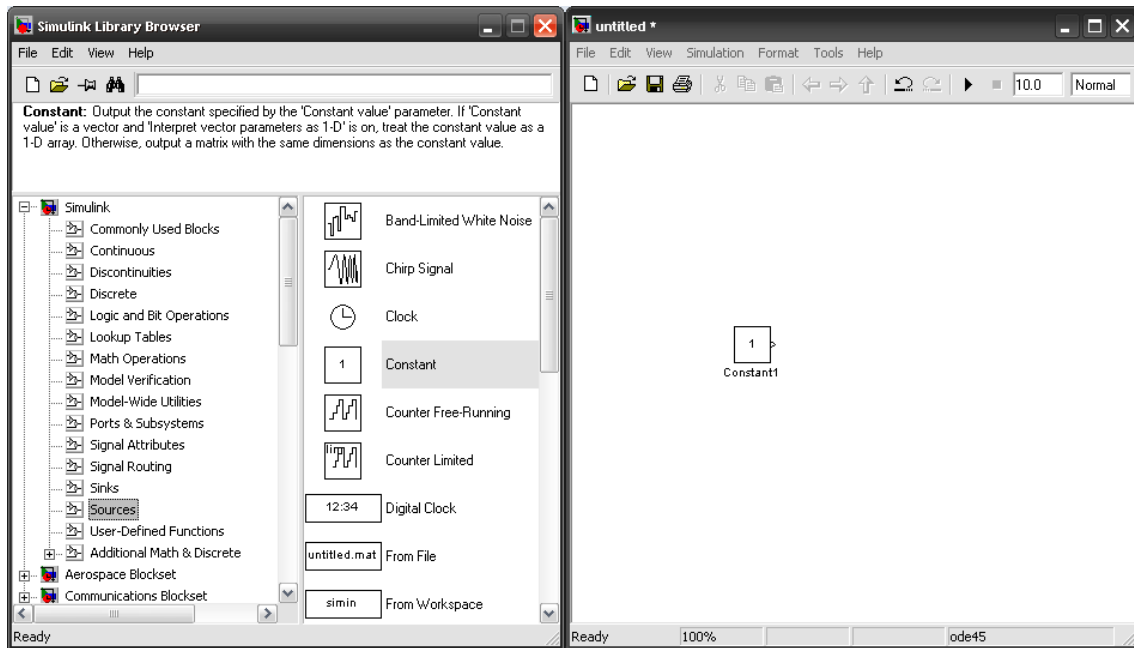
Se abrirá una ventana en blanco donde se creará el controlador:



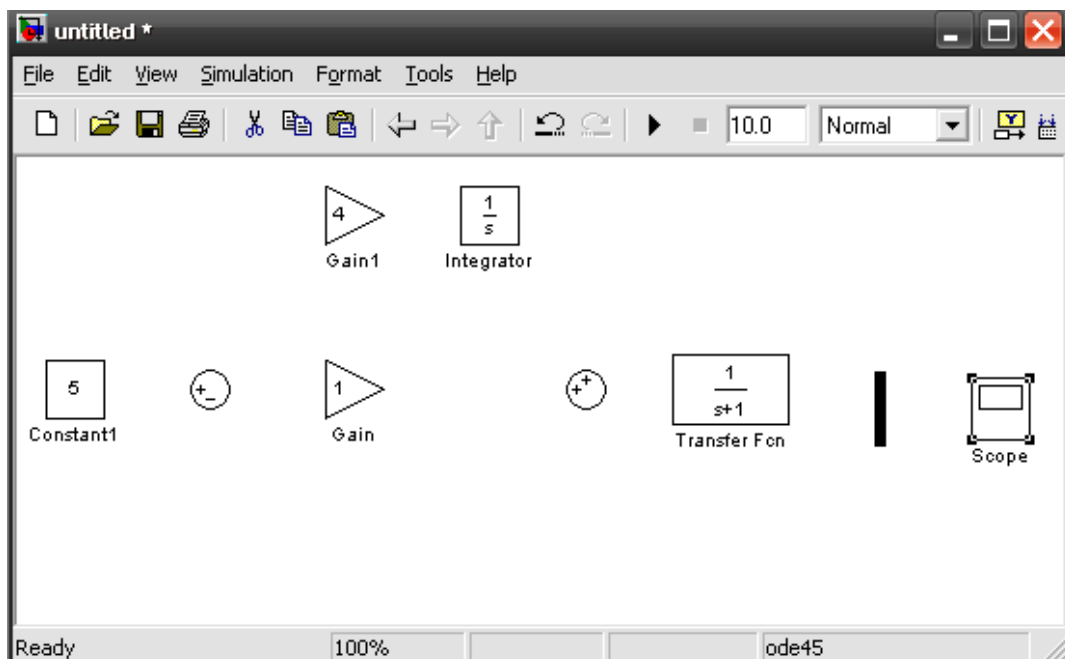
En esta ventana, que aun no se ha nombrado, se desarrollará el ejemplo. Lo primero que se hace es darle un nombre adecuado. Para ello, en el menú 'File' de la ventana se elije: File→Save.

Para crear el PI se van a necesitar un bloque integrador, un bloque sumador, dos bloques multiplicadores, un bloque que simule la planta, un bloque de entrada constante y un bloque que despliegue gráficas.

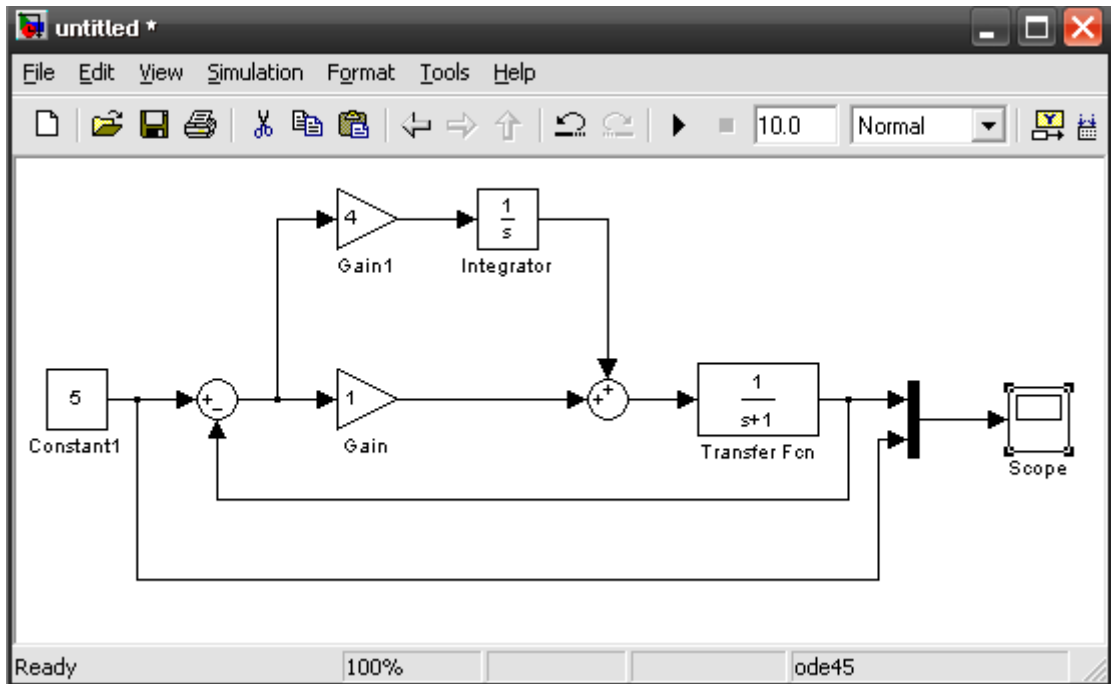
Lo primero que se debe seleccionar es el bloque que definirá la entrada constante (SP del controlador). Se selecciona primero 'Source' y a la derecha se selecciona el bloque 'Constant'. Con el botón izquierdo del ratón pulsado se lo arrastra hacia la ventana en blanco.



Se hace lo mismo con el resto de los bloques, además se pueden cambiar los atributos, como ganancias, función de transferencia, etc.

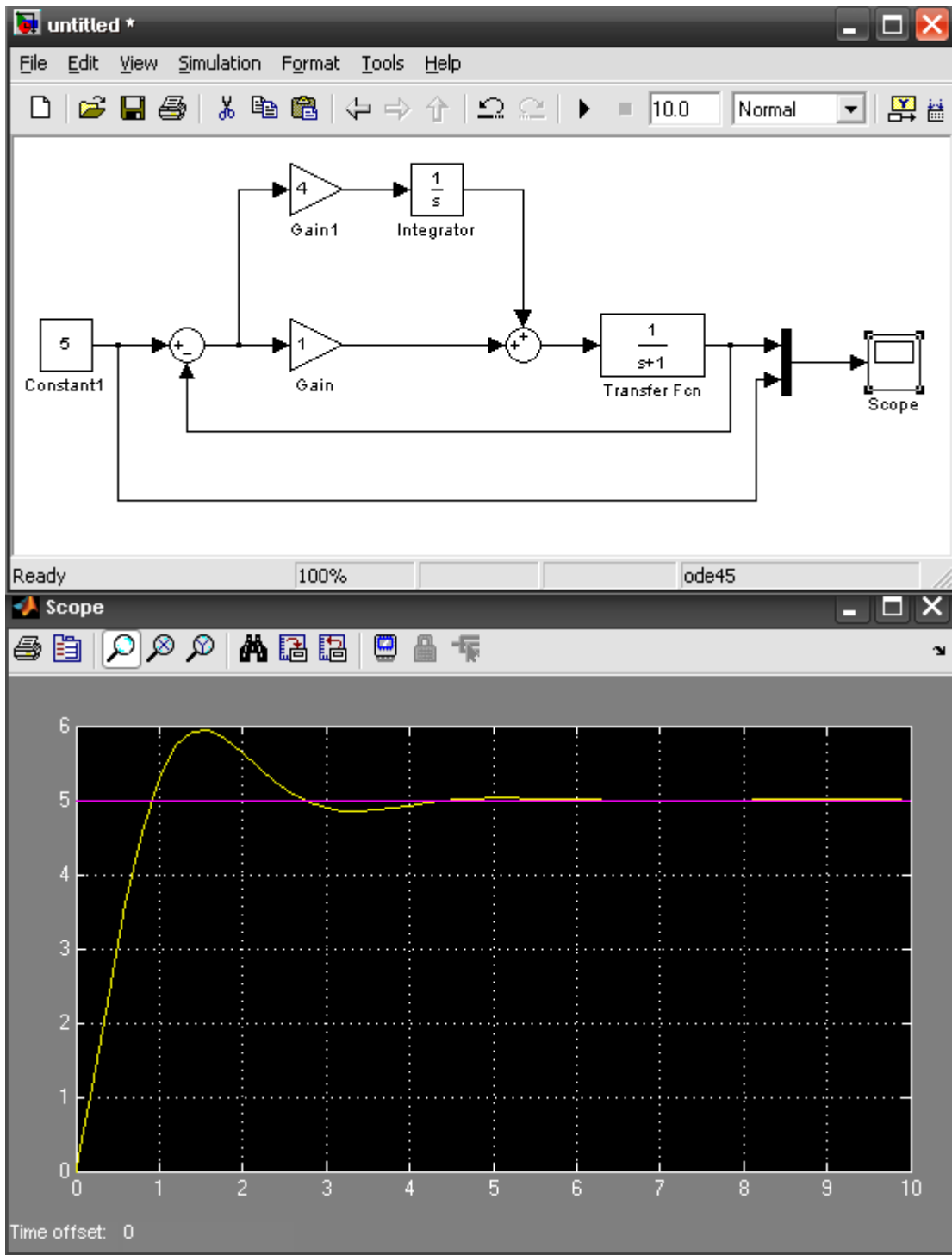


Finalmente se deben unir los bloques por medio de líneas (similar a cualquier programa de diseño de esquemáticos):



Se pulsa el botón de inicio de la simulación .

Una vez terminada la simulación el resultado se puede ver gráficamente haciendo doble click en el bloque 'Scope'



9 Conclusiones Finales

9.1 Introducción

Felizmente, se concluyó con todos los objetivos planteados cumplidos. Algunos objetivos tuvieron ciertas modificaciones, principalmente realizadas cuando se comenzó con las pruebas la planta, momento en el cual se tomó conocimiento del verdadero estado inicial del proyecto.

Al verse retrasada la culminación del proyecto, se plantearon nuevos objetivos menos exigentes que los iniciales, pero debido a la insistencia de los integrantes del mismo se pudo cumplir tanto con los objetivos iniciales como con los nuevos. Por lo que a modo de resumen se lograron más metas de las que inicialmente se plantearon.

9.2 Planificación

Como un hecho no menor, se debe resaltar la relevancia de una cuidadosa planificación y definición de todos los aspectos de cualquier proyecto. Para éste, en particular, existieron demoras en la obtención de algunos materiales, así como pruebas (iniciales), de los controladores diseñados, que resultaron no ser exitosas, motivos por los cuales se debieron replanificar ciertas actividades, poniéndose en riesgo la culminación del proyecto en la fecha estipulada.

9.3 Técnicas de Control

Uno de los puntos más importantes de este proyecto es el estudio intensivo del control PID. El mismo se realizó con un enfoque práctico, pues se evaluaron distintas implementaciones que toman en cuenta ciertas no idealidades que en el esquema del PID clásico no tienen en cuenta.

Sobre los controles avanzados, se comprobó que una vez que los parámetros de estos controladores convergen, se tienen mejores respuestas que las que se tendría incluso con un control PID. Además dichos controles tienen la gran ventaja de que, si los cambios en las condiciones de trabajo son graduales (no muy bruscos), en relativamente poco tiempo (cuestión de minutos), los mismos logran adaptarse a éstos y de ésta manera el control se continúa llevando en buena forma.

En base a la evaluación práctica de los controladores diseñados, se comprobó la gran robustez del control PID, que no por nada es de los más difundidos en la industria. Además de su bien ganada popularidad, hoy en día es objeto de estudio de muchas personas. Por dicho motivo este tipo de control tiene la ventaja extra de que existen técnicas prácticas que permiten obtener los parámetros para dichos controladores de forma muy sencilla. En cambio, para las técnicas de control adaptivo que se implementaron en este proyecto, se requiere de mucho estudio y el doble de parámetros a sintonizar. Por este motivo, un futuro proyecto puede basarse en crear reglas prácticas (empíricas) para sintonizar los controles adaptativos estudiados.

9.4 Experiencia

Con la realización del proyecto se ganó experiencia en evaluar y definir los medios de comunicación de datos, así como los protocolos a utilizar. Si este punto no se hubiese estudiado a fondo, evaluando los diferentes aspectos, la comunicación se hubiese convertido en un gran problema. Gracias a ese análisis, la misma se logró implementar sin mayores complicaciones.

Se debe destacar la importancia de la familiarización con un sistema de control moderno, ampliamente usado y muy completo, además del manejo y programación de controladores lógicos programables (PLCs), pues esta experiencia es muy valorada en la industria.

También se ganó experiencia en instrumentación industrial, pues fueron numerosas las empresas consultadas para tanto adquisición como posible donación de instrumentos y material.

9.5 Prácticas de Laboratorio

Se realizaron un conjunto de prácticas de laboratorio (basadas en la Planta Piloto v2) de manera que resulten un complemento de las prácticas actuales de la asignatura Introducción a la Teoría del Control.

9.6 Proyectos a Futuro

Dado la enorme flexibilidad de los programas hechos en Matlab y el manejo bastante intuitivo del DCS, queda abierta la posibilidad de probar cualquier tipo de controlador, por lo que se puede seguir profundizando en controles avanzados como: Redes Neuronales, GPC, MPC, etc.

Como ya se mencionó anteriormente, resultaría muy útil realizar un proyecto que se centre en crear reglas prácticas para sintonizar los controles adaptativos estudiados. Este punto puso en jaque por varias semanas el cumplimiento del objetivo referido a los controles avanzados.

Como mejoras en la física de la planta, se sugiere enfáticamente cambiar el intercambiador de calor actual, el mismo acumula mucho óxido, y más si por él no se hace circular agua en mucho tiempo. Finalmente, resultaría interesante realizar nuevas válvulas de control con la solución encontrada, pero con las esferas torneadas en aluminio o acero.

A. ANEXO: ‘Comunicación MODBUS’

Este anexo está basado en apuntes de la asignatura “Comunicaciones Industriales”, escritos por el profesor de Universidad Politécnica de Cartagena: M. Jiménez, en 2001.

A.1 Introducción

Para lograr la comunicación entre el DCS y el PLC se estudió el protocolo Modbus. Mediante el mismo se puede hacer referencia a las diferentes variables del PLC desde el DCS.

⁴²La designación Modbus Modicon corresponde a una marca registrada por Gould Inc. Como en tantos otros casos, la designación no corresponde propiamente al estándar de red, incluyendo todos los aspectos desde el nivel físico hasta el de aplicación, sino a un protocolo de enlace. Puede, por tanto, implementarse con diversos tipos de conexión física y cada fabricante suele suministrar un software de aplicación propio, que permite parametrizar sus productos.

⁴³El protocolo Modbus está situado en el nivel 7 del Modelo OSI, basado en la arquitectura maestro/esclavo o cliente/servidor, diseñado en 1979 por Modicon para su gama de controladores lógicos programables (PLCs). Convertido en un protocolo de comunicaciones estándar de facto en la industria es el que goza de mayor disponibilidad para la conexión de dispositivos electrónicos industriales. Las razones por las cuales el uso de Modbus es superior a otros protocolos de comunicaciones son:

- es público
- su implementación es fácil y requiere poco desarrollo
- maneja bloques de datos sin suponer restricciones

No obstante, se suele hablar de MODBUS como un estándar de bus de campo, cuyas características esenciales son las que se detallan a continuación.

A.2 ⁴⁴Estructura de la red

Medio Físico

El medio físico de conexión puede ser un bus semidúplex (half duplex) (RS-485 o fibra óptica) o dúplex (full duplex) (RS-422, BC 0-20mA o fibra óptica).

La comunicación es asíncrona y las velocidades de transmisión previstas van desde los 75 a los 19.200 baudios. La máxima distancia entre estaciones depende del nivel físico, pudiendo alcanzar hasta 1200 m sin repetidores.

⁴² (Jiménez, 2001)

⁴³ (Wik)

⁴⁴ (Jiménez, 2001)

Acceso al Medio

La estructura lógica es del tipo maestro-esclavo, con acceso al medio controlado por el maestro. El número máximo de estaciones previsto es de 63 esclavos más una estación maestra.

Los intercambios de mensajes pueden ser de dos tipos:

- Intercambios punto a punto, que comparten siempre dos mensajes: una demanda del maestro y una respuesta del esclavo (puede ser simplemente un reconocimiento (*acknowledge*)).
- Mensajes difundidos. Estos consisten en una comunicación unidireccional del maestro a todos los esclavos. Este tipo de mensajes no tienen respuesta por parte de los esclavos y se suelen emplear para mandar datos comunes de configuración, reset, etc.

A.3 ⁴⁵Protocolo

La codificación de datos dentro de la trama puede hacerse en modo ASCII o puramente binario, según el estándar RTU (Remote Transmission Unit). En cualquiera de los dos casos, cada mensaje obedece a una trama que contiene cuatro campos principales, según se muestra en la figura 1. La única diferencia radica en que la trama ASCII incluye un carácter de encabezamiento («»=3AH) y los caracteres CR (*Carriage Return*) y LF (*Line Feed*) al final del mensaje. Pueden existir también diferencias en la forma de calcular el CRC, puesto que el formato RTU emplea una fórmula polinómica en vez de la simple suma en módulo 16.

Con independencia de estos pequeños detalles, a continuación se da una breve descripción de cada uno de los campos del mensaje:

: (3AH)	Nº Esclavo (00-3FH)	Código de Operación	Subfunciones, Datos	LRC(16) H L	CR (0DH)	LF (0AH)
------------	---------------------------	---------------------------	---------------------	----------------	-------------	-------------

Codificación ASCII

Nº Esclavo (00-3FH)	Código de Operación	Subfunciones, Datos	CRC(P16) H L
---------------------------	---------------------------	---------------------	-----------------

Codificación RTU

Figura 1. Trama genérica del mensaje según el código empleado

Número de esclavo (1 byte):

Permite direccionar un máximo de 63 esclavos con direcciones que van del 01H hasta 3FH. El número 00H se reserva para los mensajes difundidos.

⁴⁵ (Jiménez, 2001)

Código de operación o función (1 byte):

Cada función permite transmitir datos u órdenes al esclavo. Existen dos tipos básicos de órdenes:

- Ordenes de lectura/escritura de datos en los registros o en la memoria del esclavo.
- Ordenes de control del esclavo y el propio sistema de comunicaciones (RUN/STOP, carga y descarga de programas, verificación de contadores de intercambio, etc.)

La tabla 1 muestra la lista de funciones disponibles en el protocolo MODBUS con sus correspondientes códigos de operación.

Función	Código	Tarea
0	00 _H	Control de estaciones esclavas
1	01 _H	Lectura de n bits de salida o internos
2	02 _H	Lectura de n bits de entradas
3	03 _H	Lectura de n palabras de salidas o internos
4	04 _H	Lectura de n palabras de entradas
5	05 _H	Escritura de un bit
6	06 _H	Escritura de una palabra
7	07 _H	Lectura rápida de 8 bits
8	08 _H	Control de contadores de diagnósticos número 1 a 8
9	09 _H	No utilizado
10	0A _H	No utilizado
11	0B _H	Control del contador de diagnósticos número 9
12	0C _H	No utilizado
13	0D _H	No utilizado
14	0E _H	No utilizado
15	0F _H	Escritura de n bits
16	10 _H	Escritura de n palabras

Tabla 1. Funciones básicas y códigos de operación

Campo de subfunciones/datos (n bytes):

Este campo suele contener, en primer lugar, los parámetros necesarios para ejecutar la función indicada por el byte anterior. Estos parámetros podrán ser códigos de subfunciones en el caso de órdenes de control (función 00H) o direcciones del primer bit o byte, número de bits o palabras a leer o escribir, valor del bit o palabra en caso de escritura, etc.

Palabra de control de errores (2 bytes):

En código ASCII, esta palabra es simplemente la suma de comprobación ("checksum") del mensaje en módulo 16 expresado en ASCII.

A.4 ⁴⁶Las funciones más importantes del protocolo

Función 0:

Esta función permite ejecutar órdenes de control, tales como marcha, paro, carga y lectura de programas de usuario del autómatas. Para codificar cada una de las citadas órdenes se emplean los cuatro primeros bytes del campo de datos. La trama resultante es la representada en la figura 3 y la interpretación de los códigos de subfunción se especifica en la tabla 2.

En caso de las órdenes de marcha y paro, el campo de «información» de la trama representada en la figura 3 está vacío y, por tanto, el mensaje se compone simplemente de 6 bytes de función más 2 bytes de CRC. La respuesta del esclavo a estas órdenes es un mensaje idéntico al enviado por el maestro. Cabe señalar, además, que después de un paro el autómatas sólo acepta ejecutar subfunciones de la función 00H.

Nº Esclavo (00-3F _H)	00 _H	Código Subfunción SF0 SF1	Datos Subfunción D0 D1	Información	CRC(16) H L
----------------------------------	-----------------	---------------------------	------------------------	-------------	-------------

Figura 3. Trama genérica de las subfunciones de control de esclavos (cód. función 00H)

Código subfunción SF0 SF1		Datos subfunción D0 D1		Tarea
00 _H	00 _H	00 _H	00 _H	Paro del esclavo sin inicializar
00 _H	01 _H	00 _H	00 _H	Marcha del esclavo sin inicializar
00 _H	02 _H	00 _H	00 _H	Marcha e inicialización del esclavo
00 _H	03 _H	00 _H	XX _H	Lectura de la secuencia XX de programa de usuario en el esclavo
00 _H	04 _H	YY _H	XX _H	Carga de una secuencia de programa de usuario en el esclavo Petición: YY = secuencia a cargar, XX = próxima secuencia Respuesta: XX = código error, YY = 00

Tabla 2. Subfunciones correspondientes a la función =00H

Funciones 1 y 2:

Lectura de bits del autómatas. La trama es la indicada en la figura 4. La forma de direccionamiento de los bits es a base de dar la dirección de la palabra que los contiene y luego la posición del bit. La respuesta es dada siempre en octetos completos.

⁴⁶ (Jiménez, 2001)

Petición del maestro

Nº Esclavo (00-3F _H)	01 _H o 02 _H	Dirección 1 ^{er} Bit PP PB	Nº de Bits NN NN	CRC H L
----------------------------------	---	-------------------------------------	------------------	---------

PPP = Dirección de la palabra (hex), B= Dirección del bit dentro de la palabra 0 a F_H.

Respuesta del esclavo

Nº Esclavo (00-3F _H)	01 _H o 02 _H	Nº Octetos leídos NN NN	1 ^{er} Octeto B7..B0	Otros Octetos Hasta máx. 256	CRC H L
----------------------------------	---	-------------------------	-------------------------------	------------------------------	---------

Figura 4. Petición y respuesta de la función: Lectura de bits (01H, 02H)

Funciones 3 y 4:

Lectura de palabras del autómatas. La trama es la indicada en la figura 5. La petición indica el número de palabras a leer, mientras que en la respuesta se indica el número de octetos leídos.

Petición del maestro

Nº Esclavo (00-3F _H)	03 _H o 04 _H	Dirección 1ª Palabra PP PP	Nº de Palabras NN NN	CRC H L
----------------------------------	---	----------------------------	----------------------	---------

PPPP = Dirección de la palabra (hex)

Respuesta del esclavo

Nº Esclavo (00-3F _H)	03 _H o 04 _H	Nº Octetos leídos NN NN	1 ^{er} Palabra H L	Otras Palabras Hasta máx. 128 H L, H L, HL,...	CRC H L
----------------------------------	---	-------------------------	-----------------------------	--	---------

Figura 5. Petición y respuesta de la función: Lectura de palabras (03H,04H)

Función 5:

Escritura de un bit. La trama es la indicada en la figura 6. El direccionamiento del bit se efectúa tal como se ha indicado para las funciones 1 y 2.

Petición del maestro

Nº Esclavo (00-3F _H)	05 _H	Dirección Bit PP PB	XX _H	00 _H	CRC H L
----------------------------------	-----------------	---------------------	-----------------	-----------------	---------

PPP = Dirección de la palabra (hex), B= Dirección del bit dentro de la palabra 0 a F_H.

Respuesta del esclavo

Nº Esclavo (00-3F _H)	05 _H	Dirección Bit PP PB	XX _H	00 _H	CRC H L
----------------------------------	-----------------	---------------------	-----------------	-----------------	---------

XX_H = 00H para bit = 0 y XX_H = FF_H para bit = 1

Figura 6. Petición y respuesta de la función: Escritura de un bit (05H)

Función 6:

Escritura de una palabra. La trama es la indicada en la figura 7.

Petición del maestro

Nº Esclavo (00-3F _H)	06 _H	Dirección Palabra PP PP	Valor Palabra DD DD	CRC H L
----------------------------------	-----------------	-------------------------	---------------------	---------

Respuesta del esclavo

Nº Esclavo (00-3F _H)	06 _H	Dirección Palabra PP PP	Nº de Palabras DD DD	CRC H L
----------------------------------	-----------------	-------------------------	----------------------	---------

Figura 7. Petición y respuesta de la función: Escritura de una palabra (06H)

Función 15:

Escritura de bits del autómeta. La trama es la indicada en la figura 11. La forma de direccionamiento es análoga a la indicada para las funciones 1 y 2.

Petición maestro

Nº Esclavo (00-3F _H)	0F _H	Dirección 1 ^{er} Bit PP PB	Nº de Bits NN NN	Nº de Octetos M	Valor de los bits 8xM valores	CRC H L
----------------------------------	-----------------	-------------------------------------	------------------	-----------------	-------------------------------	---------

Respuesta del esclavo

Nº Esclavo (00-3F _H)	0F _H	Dirección 1 ^{er} Bit PP PB	Nº de Bits NN NN	CRC H L
----------------------------------	-----------------	-------------------------------------	------------------	---------

Figura 11. Petición y respuesta: Escritura de bits (0FH)

Función 16:

Escritura de palabras del autómeta. La trama es la indicada en la figura 12.

Petición maestro

Nº Esclavo (00-3F _H)	10F _H	Dirección 1ª Palabra PP PP	Nº de Palabras NN NN	Nº de Octetos M	Valor de las palabras HL HL...	CRC H L
----------------------------------	------------------	----------------------------	----------------------	-----------------	--------------------------------	---------

Respuesta del esclavo

Nº Esclavo (00-3F _H)	10 _H	Dirección 1ª Palabra PP PP	Nº de Palabras NN NN	CRC H L
----------------------------------	-----------------	----------------------------	----------------------	---------

Figura 12. Petición y respuesta: Escritura de palabras (10H)

Mensajes de error:

Puede ocurrir que un mensaje se interrumpa antes de terminar. Cada esclavo interpreta que el mensaje ha terminado si transcurre un tiempo de silencio equivalente a 3,5 caracteres. Después de este tiempo el esclavo considera que el carácter siguiente es el campo de dirección de esclavo de un nuevo mensaje.

Cuando un esclavo recibe una trama incompleta o errónea desde el punto de vista lógico, envía un mensaje de error como respuesta, excepto en el caso de mensajes de difusión. La trama del mensaje de error es la indicada en la figura 13.

Respuesta del esclavo

Nº Esclavo (00-3F _H)	Código Función	Código Error	CRC H L
--	-------------------	-----------------	------------

Figura 13. Trama de mensaje de error

Código Función = Código función recibido + 80H

Código Error = 01 Código de Función erróneo

02 Dirección incorrecta

03 Datos incorrectos

06 Autómata ocupado

Si la estación maestra no recibe respuesta de un esclavo durante un tiempo superior a un límite establecido, declara el esclavo fuera de servicio, a pesar de que al cabo de un cierto número de ciclos hace nuevos intentos de conexión.

A.5 ⁴⁷Nivel de aplicación

Como se ha dicho a nivel general de buses de campo, el nivel de aplicación de MODBUS no está cubierto por un software estándar, sino que cada fabricante suele suministrar programas para controlar su propia red. No obstante, el nivel de concreción en la definición de las funciones permite al usuario la confección de software propio para gestionar cualquier red, incluso con productos de distintos fabricantes.

⁴⁷ (Jiménez, 2001)

B. ANEXO: ‘Comunicación OPC’

⁴⁸OPC fue desarrollado para estandarizar los sistemas propietarios de Drivers de Control y Automatización de redes industriales y Buses de Campo de múltiples fabricantes, generando así, interoperabilidad entre estos sistemas.

El problema histórico de los sistemas de control SCADA y DCS, de distintos fabricantes, ha sido que sus sistemas de datos tienen diferentes formatos y protocolos de comunicación. A ello se suma que redes industriales (SimaticNet, DeviceNet, etc.) y Buses de Campo (Modbus, Profibus, etc.), son incompatibles en comunicación originando problemas como:

- a. Arquitectura propietaria centralizada, con baja flexibilidad y características de integración con otros fabricantes.
- b. Ninguna garantía de interoperabilidad entre sistemas propietarios.
Al ser sistemas propietarios presentan un elevado costo de integración en Software y Hardware (Duplicación de esfuerzos).

B.1 ⁴⁹Introducción

OPC es la forma abreviada de “OLE *for Process Control*” y significa tecnología OLE para control de procesos. OPC es una interface estándar basada en los requerimientos de la tecnología OLE (ahora ActiveX) / COM (*Component Object Model*) y DCOM (*Distributed COM*) de Microsoft, que facilita el intercambio de datos de forma simple entre aplicaciones de control y automatización, entre dispositivos y sistemas de campo, y entre aplicaciones administrativas y de oficina. La tecnología ActiveX/COM se define como un conjunto de componentes individuales de software que pueden interactuar y repartir datos. OPC simplifica la interface entre componentes de automatización de distintos fabricantes, así como entre programas y aplicaciones, y entre sistemas administrativos y de visualización. El estándar, gobernado por la Fundación OPC, es de dominio público y está disponible para cualquiera que quiera usarlo.

B.2 Motivos para el desarrollo de la especificación OPC

La arquitectura informática para la industria de proceso incluye los siguientes niveles:

- Gestión de campo: información sobre los dispositivos de instrumentación (estado, Constitución, configuración, etc.).
- Gestión de proceso: datos sobre el proceso productivo adquiridos y procesados por sistemas SCADA y DCS.
- Gestión de negocio: integración de la información de planta en los sistemas que gestionan los aspectos financieros de la fabricación.

Se trata de que en la industria se puedan utilizar herramientas estándar (paquetes SCADA, bases de datos, hojas de cálculo) para construir un sistema que responda a sus necesidades de mejora de la productividad. Para ello es necesario desarrollar una arquitectura de comunicaciones abierta y efectiva que se centre en el acceso a los datos, no en los tipos de datos.

⁴⁸ (Vidal & Figueroa, 2002)

⁴⁹ (Blanco, 2005)

Tradicionalmente, los fabricantes de software para acceso de datos de proceso tenían que desarrollar *drivers* específicos para cada tipo de hardware al que querían acceder. Cada software requería un *driver* distinto para cada hardware, a lo que hay que añadir el esfuerzo de las continuas actualizaciones. Por tanto, se estaba forzado a volver al mismo fabricante cada vez que se necesitaba cambiar o ampliar los sistemas de control.

En resumen, los fabricantes de hardware no pueden desarrollar un *driver* eficiente utilizable por todos los clientes a causa de las diferencias de protocolos entre ellos. En contraste, OPC estandariza la tecnología más que los productos. Con OPC, los fabricantes de hardware sólo tienen que preparar un conjunto de componentes de software para que los clientes los usen en sus aplicaciones, y así los usuarios finales tienen más alternativas de integrar distintos sistemas.

La clave de este estándar es el *plug-and-play*, un concepto desarrollado por Microsoft y otras compañías hace unos años con la finalidad de usar un estándar común para la configuración del hardware automáticamente, permitiendo así que un recurso se conectase fácilmente a otro e inmediatamente trabajasen sin la necesidad de un extenso procedimiento de instalación o una compleja configuración.

En lugar de tener que aprender cómo se usan 100 o más paquetes de herramientas, los usuarios tendrán que aprender una serie de herramientas, ya que todos los *drivers* de OPC trabajan de la misma manera. El propósito de OPC es obligar a las industrias suministradoras de automatismos a llevar todos los *drivers* hacia una forma estándar.

Esencialmente, OPC define una interface común que permite desarrollar una interface de trabajo para ser hecha una vez y entonces reutilizarla fácilmente

B.3 Ventajas de OPC

El futuro inmediato de los sistemas de control pasa por la flexibilidad tanto en el hardware como en el software. Se podrán conectar entre sí aquellos componentes que más se adapten a las necesidades de cada punto concreto del proceso. La información será gestionada de manera que a cada usuario se le presenten los datos que necesite y de la forma en que los necesite. Las señales se capturan directamente de los PLC's de las máquinas, equipos, procesos, etc. utilizando tecnología de Bus de Campo y OPC.

Esto implica sistemas con gran capacidad de “apertura”. Frente a otros sistemas de control, OPC presenta, entre otras, las siguientes ventajas:

- Con OPC se puede intercambiar datos a través de una interface común entre dispositivos de hardware y aplicaciones de software desarrolladas por una variedad de fabricantes. Respaldada por la tecnología NT de Microsoft, esta interface permite comunicarse con diversos controladores del proceso, con independencia del software de control o los recursos que haya en el proceso.
- La tecnología de OPC y Windows hacen posible la combinación de hardware de control programable y software sin necesidad de *drivers* especiales. OPC minimiza el número de *drivers* necesarios, éstos son públicos y se reduce también la carga de dispositivos.
- OPC facilita el acceso flexible y desde productos de MS-Office a datos del proceso. Usando el set de OPC, los datos pueden pasarse desde cualquier fuente de datos a cualquier aplicación conforme a OPC. Estas aplicaciones incluyen HMI's, tendencias, hojas de cálculo, bases de datos, etc.

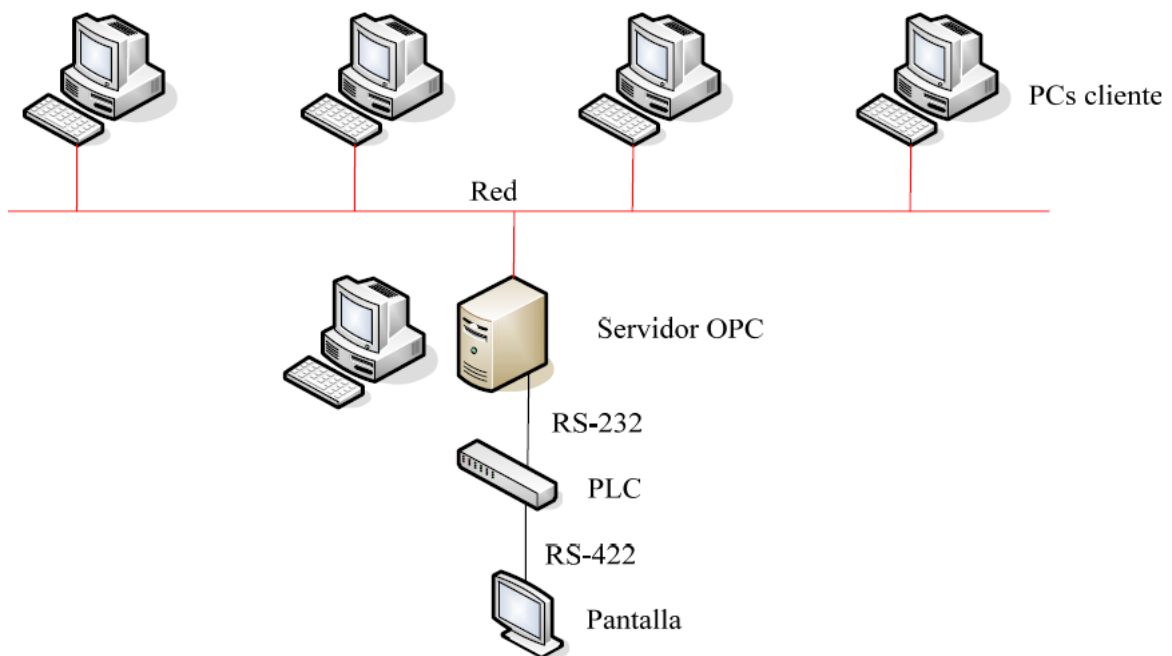
- OPC representa un complemento importante en las actividades relacionadas con buses de campo. El propósito principal de la estandarización de los buses de campo es la transferencia de datos de forma rápida y fiable.
- OPC estandariza la comunicación de modo que cualquier servidor OPC y cualquier aplicación OPC pueden trabajar juntos sin ningún problema. OPC es un estándar industrial público. La Fundación OPC controla las especificaciones que están basadas en la tecnología OLE de Microsoft para el intercambio de datos usando una arquitectura de cliente-servidor.

El diseño de las interfaces OPC soporta arquitecturas distribuidas en red. El acceso a servidores OPC remotos se hace empleando la tecnología DCOM de Microsoft. OPC es un estándar de comunicaciones que proporciona verdadera interoperabilidad y escalabilidad. Esto permite visualizar, analizar, importar o lo que se quiera hacer con aplicaciones de casi todos los vendedores usando una o más de las especificaciones de OPC.

Seleccionando los estándares basados en OPC se obtiene verdadera interoperabilidad, se reducen los costos y tiempo de implantación y se construye un completo sistema escalable para el futuro.

B.4 Integración de sistemas heterogéneos con OPC

Como ya se ha dicho, OPC permite interactuar con una gran variedad de sistemas distintos de todo tipo: diferentes lenguajes de programación, PCs, PLCs, etc. En la siguiente figura se muestra el esquema de esta interconexión en el caso objeto del presente proyecto.



Integración de diferentes sistemas con OPC

B.5 Componentes OPC

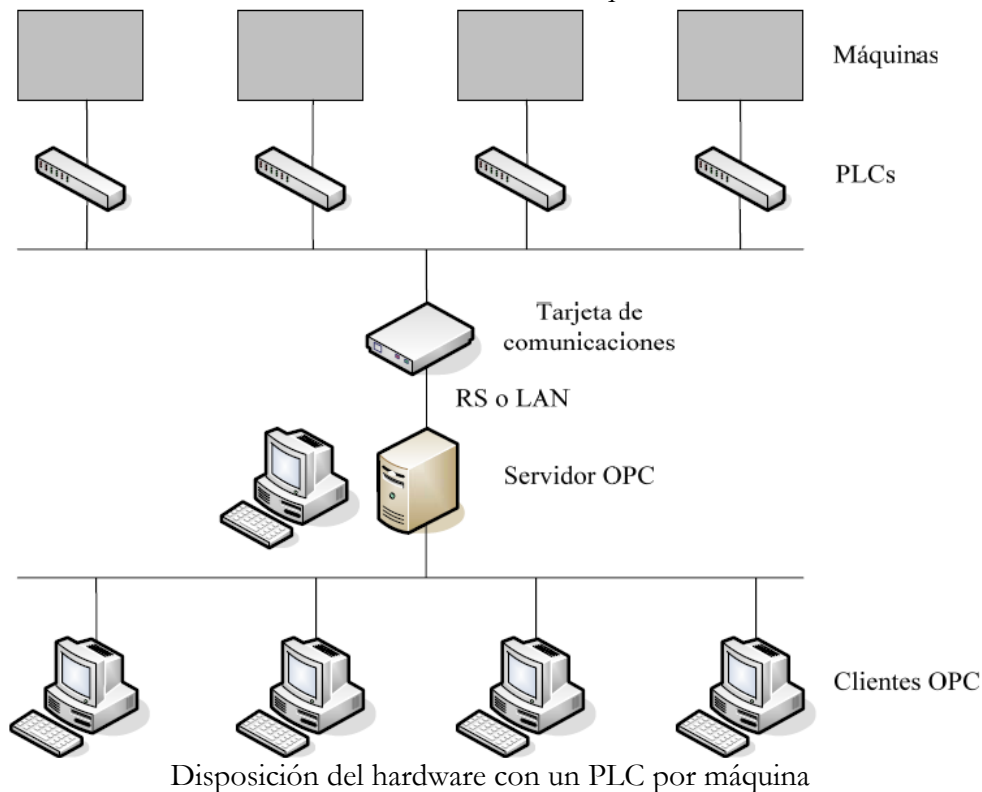
Un servidor OPC se compone de varios objetos que se ajustan a la norma COM:

- **El objeto servidor:** Contiene información sobre la configuración del servidor OPC y sirve de contenedor para los objetos tipo grupo.
- **El objeto grupo:** Sirve para organizar los datos que leen y escriben los clientes (por ejemplo, valores en una pantalla MMI5 o en un informe de producción). Se pueden establecer conexiones por excepción entre los clientes y los elementos de un grupo. Un grupo puede ser público, es decir, compartido por varios clientes OPC.
- **El objeto ítem:** Representa conexiones a fuentes de datos en el servidor (no son las fuentes de datos en sí). Tiene asociados los atributos *Value*, *Quality* y *Time Stamp*. Los accesos a los ítems OPC se hacen a través de los grupos OPC y los clientes pueden definir el ritmo al cual el servidor les informa sobre cambios en los datos.

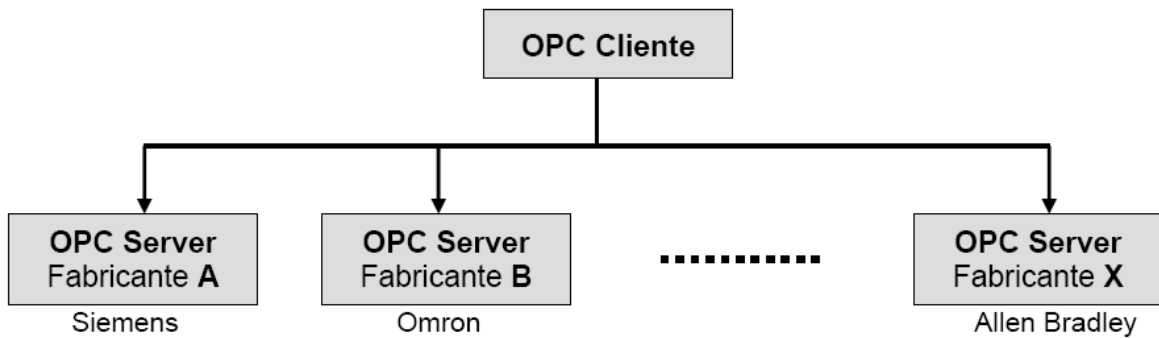
El acceso a los objetos COM se hace a través de interfaces, que son lo único que ven los clientes OPC. Los objetos descritos son representaciones lógicas, que no tienen por qué coincidir con la implementación que se haga del servidor OPC.

B.6 Hardware

Cada máquina de la línea dispone de un PLC, con su correspondiente módulo I/O, y todos los PLCs están conectados a una unidad central, una tarjeta de comunicaciones (una por fabricante) que, a su vez, puede situarse dentro de un PC o conectarse al mismo a través de RS o LAN. Este PC será el servidor OPC, al que se conectan los clientes OPC.



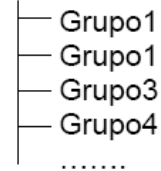
B.6.1 Tecnología Cliente-Servidor



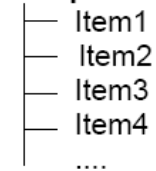
El objeto **Servidor** es una estructura en la que se almacena información del Server y es un contenedor de objetos **Grupo**.

El objeto **Grupo** es una estructura en la que se almacena información del Grupo y es un contenedor de objetos **Item**.

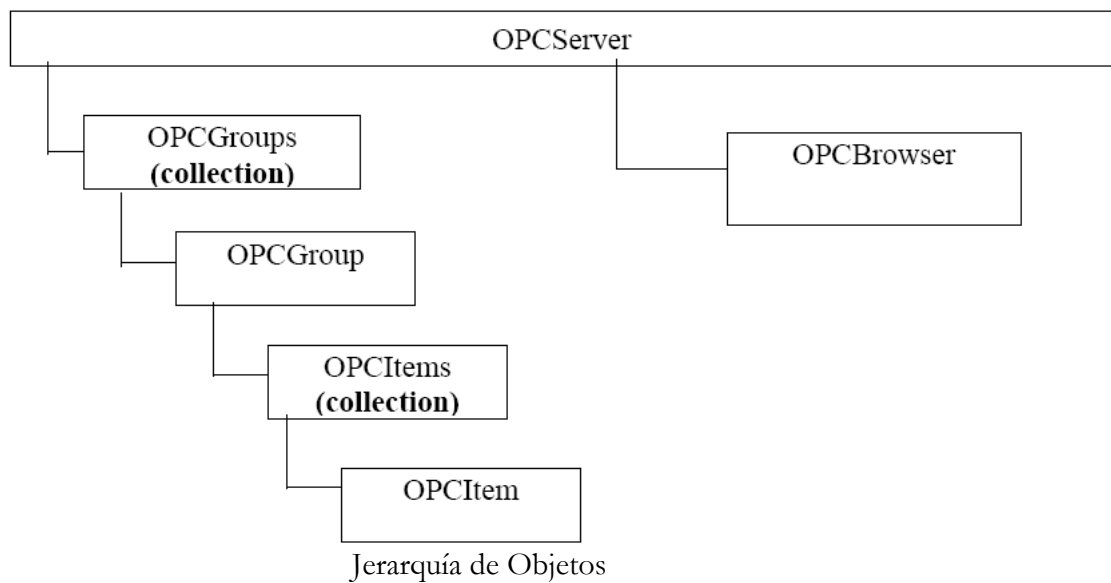
OPCSERVER



Grupo1



Modelo de Objetos de OPC Server



Object	Description
OPCServer	An instance of an OPC Server. You must create an OPCServer object before you can get references to other objects. It contains the OPCGroups Collection and creates OPCBrowser objects.
OPCGroups	An Automation collection containing all of the OPCGroup objects this client has created within the scope of the OPCServer that the Automation Application has connected to via the OPCServer.Connect()
OPCGroup	An instance of an OPCGroup object. The purpose of this object is to maintain state information and provide the mechanism to provide data acquisition services for the OPCItem Collection object that the OPCGroup object references.
OPCItems	An Automation collection containing all of the OPCItem objects this client has created within the scope of the OPCServer, and corresponding OPCGroup object that the Automation Application has created.
OPCItem	An automation object that maintains the item's definition, current value, status information, last update time. Note the Custom Interface does not provide a separate Item Object.
OPCBrowser	An object that browses item names in the server's configuration. There exists only one instance of an OPCBrowser object per instance of an OPC Server object.

B.6.2 Modo De Operación

1. Establecer la conexión con el OPC Server.

a) Definir el nombre del OPC Server

Ejemplos de nombre de servidor:

```
Set ServerReactor = New OPCServer
Set ServerAutoclave1 = New OPCServer
Set ServerLinea1 = New OPCServer
```

b) Preparar la cadena de conexión y conectarse con el OPC Server

Las cadenas de Conexión dependen del fabricante del OPC Server. Ejemplos:
ServerLinea1.Connect "PhoenixContact.Interbus.2", "IQ3-6" 'Phoenix Contact

ServerReactor.Connect "HitachiOPC.DA2", "UVIGO-C5BFBE7C7"
'Hitachi

2. Definir los Ítems necesarios para la Lectura / Escritura.

Los Ítems equivalen a variables de proceso: Marcha/Paro de bombas, Apertura/Cierre de válvulas, señales de control de válvulas de control (4-20 mA, 1-5 V, etc.), lectura de variables temperatura, caudal, nivel, presión, pH, etc., Activar/Desactivar alarmas, etc.

Las variables pueden ser:

Boolean Marcha/Paro, Activar/Desactivar, *Trigger*, etc.

Integer Valores enteros desde **-32767** , **0** , **+32767**. 16 bits.

El bit 15 se reserva para el signo.

Long Integer Enteros -2147483647 , 0 , +2147483647

Real / Double Cualquier valor con decimales.

Alfanumérico / Texto Texto.

String Texto.

Una vez definida una variable sólo puede contener valores dentro de su rango, excepto las variables definidas como Variant.

Variant: es una estructura que puede contener cualquier variable de las anteriores. Se compone de dos campos, el primero indica el tipo y el segundo contiene el valor.

Además las variables pueden ser de Lectura y/o Escritura. *Read*, *Write* o *ReadWrite*.

Ejemplo de nombre de Ítems:

Primero crear el *Array* de ítems. Se fija la dimensión igual número máximo de variables.

```
Private NombreItems(1 To 6) As String 'Array con los nombres de los ítems cargados
```

```
NombreItems(1) = "PLC1.M0200" 'Marcha / Paro
```

```
NombreItems(2) = "PLC1.WM0050" 'Velocidad
```

```
NombreItems(3) = "PLC1.WM0010" 'Contador Total de Piezas NO  
CONFORMES
```

```
NombreItems(4) = "PLC1.WM0060" 'Contador Total de Piezas  
CONFORMES
```

```
NombreItems(5) = "PLC1.WM0030" 'Contador Parcial de Piezas NO  
CONFORMES
```

```
NombreItems(6) = "PLC1.WM0040" 'Contador Parcial de Piezas  
CONFORMES
```

3. Crear el contenedor de Grupos

```
Set GrupoLinea1 = MyServer.OPCGroups
```

4. Crear un Grupo dentro del contenedor de grupos

```
Set Grupo1 = GrupoLinea1.Add("Grupo") ‘ “Grupo” es la etiqueta del  
contenedor de grupos
```

5. Definir los Ítems dentro del grupo

```
Set Items1 = Grupo1.OPCItems
```

6. Añadir los ítems (variables) al contenedor de ítems

```
Items1.AddItem 6 , NombreItems ‘Se han excluido otras variables: Errores y  
Calidad’.
```

7. Indicar la velocidad de actualización de los valores del grupo.

```
Grupo1.UpdateRate = 100 ‘100 milisegundos’. Actualizar los valores 10 veces por  
segundo
```

8. Suscribirse el grupo Grupo1 para recibir notificaciones de las variables del OPC Server.

```
Grupo1.IsSubscribed = True
```

Las notificaciones de las variables se reciben de manera asíncrona.

9. Activar el Grupo1 para que reciba las notificaciones a las que está suscrito.

```
Grupo1.IsActive = True
```

Si un grupo no está activo no recibe los valores de las variables.

Para Leer o Escribir las variables se procede de la siguiente forma:

10. Leer el estado de las seis variables de la línea.

Grupo1.SyncRead OPCDevice, 6, ItemServerHandles, ItemValores, ItemServerErrors, ItemCalidades, ItemTimeStamps

ItemServerHandles es un puntero a cada uno de las variables. Se usa para acceder rápidamente. En este caso va desde 1 hasta 6. **ItemServerHandles(1 a 6)**. Los valores se devuelven en **ItemValores**, que también va de 1 a 6. **ItemValores(1)** será el estado de la línea, **ItemValores(2)** la velocidad, etc.

11. Leer el estado de una variable

Grupo1.AsyncRead OPCDevice, 1, ItemServerHandles(1), ItemValores
Se han eliminado otras variables para facilitar la comprensión.

12. Escribir una variable.

Por ejemplo, arrancar la Línea:

ItemValores = 1

Grupo1.AsyncWrite 1, ItemServerHandles(1), ItemValores

Se han eliminado otras variables para facilitar la comprensión.

Cambiar la velocidad, fijar al 60 % de la velocidad máxima:

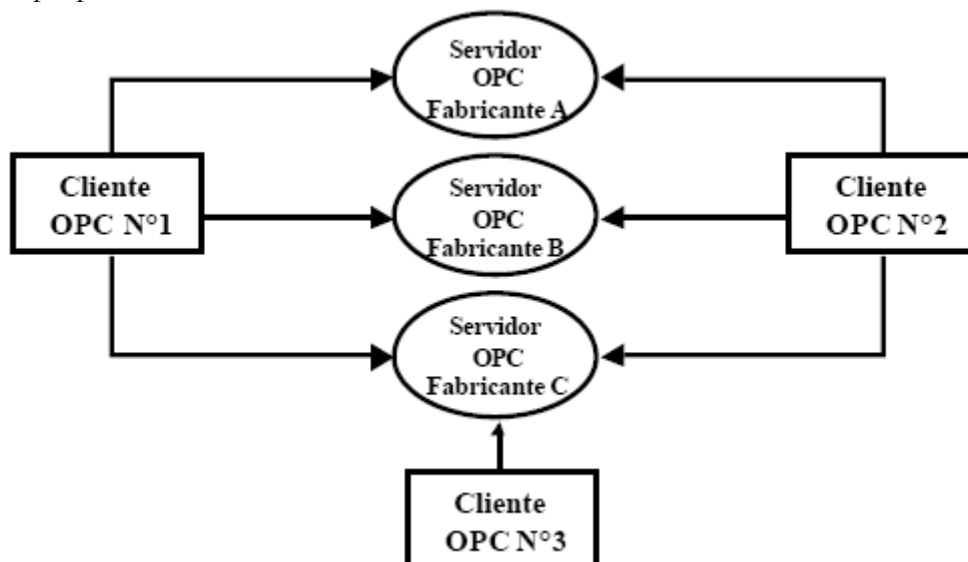
ItemValores = 60

Grupo1.AsyncWrite 2, ItemServerHandles(2), ItemValores

Se han eliminado otras variables para facilitar la comprensión.

B.7 ⁵⁰ Servidores OPC

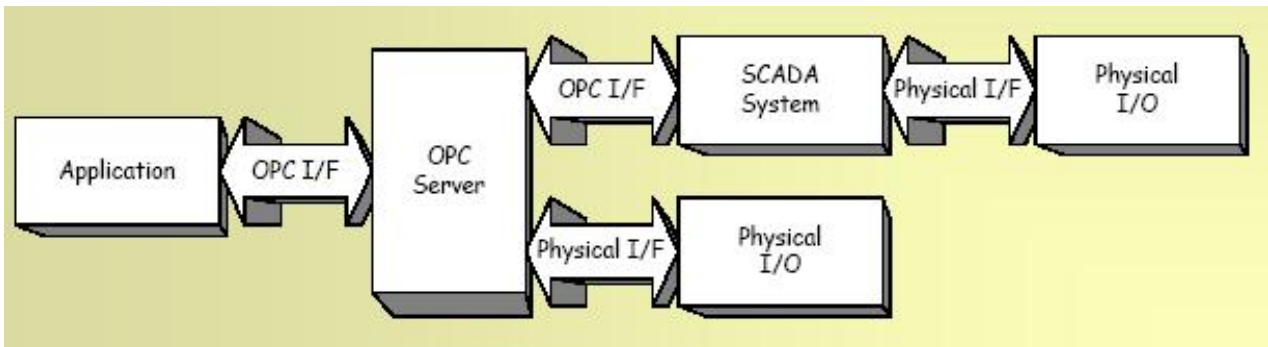
Una aplicación cliente OPC, puede conectarse por medio de una red, a varios servidores OPC proporcionados por uno o más fabricantes (Ver siguiente figura). De esta forma no existe restricción por cuanto a tener un Software Cliente para un software Servidor, lo que es un problema de interoperabilidad que hoy en día se aprecia con sistemas del tipo propietario.



Relación de trabajo Cliente/Servidor.

⁵⁰ (Vidal & Figueroa, 2002)

Sistemas de control supervisorio como lo son SCADA o DCS pueden comunicarse con un Servidor OPC (Ver siguiente figura) y proveer a este, información de los dispositivos de campo asociados. De esta forma, aplicaciones cliente OPC de otros fabricantes tendrán acceso a estos datos por medio del Servidor:



⁵¹Relación Cliente/Servidor y Sistemas SCADA /DCS

Servidor de Acceso a datos OPC (OPC DA)

A un alto nivel, está compuesto por los objetos:

Servidor: Mantiene la información sobre sí mismo, y unifica los Datos dentro de un Grupo.

Grupo: Dota de un mecanismo que contiene en forma lógica los ítems. Se clasifican en público o Local.

Ítem: Es un valor, una condición y permanece o varía en el tiempo. Es una dirección específica de los datos y no la fuente de datos.

Servidor de Alarmas, Condiciones y Eventos OPC (OPC AE)

Provee de Interfaces donde Clientes OPC son notificados de Sucesos. Estos mecanismos se definen como:

Alarma: Condición anormal del sistema, por lo que es un caso especial de esta.

Condición: Estado, nombrado evento por contener condiciones asociadas a una etiqueta como HighAlarm, Normal, LowAlarm.

Evento: Ocurrencia perceptible, de importancia al servidor OPC, de los dispositivos que representa o de sus dispositivos OPC.

Servidor de Acceso a Datos Históricos OPC (OPC HDA)

Provee de una interfaz Cliente OPC de Acceso a Datos Históricos, que facilita el uso de aplicaciones de acceso a datos de “Piso- Planta” industrial. Por su proceso, a esta interface se le denomina “Automatización Amigable”.

⁵¹ (Wik)

C.ANEXO: ‘DCS Vs SCADA’

A lo largo del proyecto se utiliza la denominación DCS, pero nunca se introdujo formalmente su significado.

Un DCS refiere a un sistema de control de procesos o sistemas dinámicos de cualquier tipo, en el cual los elementos de control no están en una ubicación central (cerebro), sino que se encuentran distribuidos por todo el sistema. Cada componente de los sub-sistemas es controlado por uno o más controladores, mientras que el sistema entero de controladores está conectado por una red de comunicaciones y monitoreo.

⁵² Tipo de Arquitectura	Centralizada	Distribuida
Tipo de control predominante	Supervisorio: Lazos de control cerrados por el operador. Adicionalmente: control secuencial y regulatorio.	Regulatorio: Lazos de control cerrados automáticamente por el sistema. Adicionalmente: control secuencial, batch, algoritmos avanzados, etc.
Tipos de variables	Desacopladas	Acopladas
Área de acción	Áreas geográficamente distribuidas.	Área de la planta.
Unidades de adquisición de datos y control	Remotas, PLCs.	Controladores de lazo, PLCs, DCS
Medios de comunicación	Radios, satélites, líneas telefónicas, conexión directa, LAN, WAN.	Redes de área local, conexión directa.
Base de datos	Centralizada	Distribuida

Por otro lado un sistema SCADA refiere solo a una interfaz HMI (programación en el PLC), mientras que en un DCS se pueden programar los algoritmos de control necesarios así como fabricar su propia HMI (programación en el PLC así como en la PC supervisora).

En este proyecto se utilizó al sistema DCS 800xA como un SCADA, esto se debe a que el PLC del que se dispone no es el diseñado para interactuar en forma de DCS con el sistema 800xA. Si se usaran los controladores AC800M la programación del mismo se realiza directamente desde el DCS y no es necesario que linkear las variables en el PLC con variables en el DCS, cómo se debió hacer en este proyecto, pues el sistema se encarga de esa tarea.

⁵² (Ferrari, 2005)

D. ANEXO: ‘MatLab-Simulink’

Una de las partes centrales del proyecto es MatLab y en particular Simulink. Como toda nueva herramienta también presentó gran complejidad, ya que solo uno de nosotros había tenido contacto directo con Simulink. Para superar esta adversidad se decidió se decidió separar el problema en dos partes:

1. Estudio general de Simulink
2. Estudio de las herramientas:
 - StateFlow
 - Truth Table
 - S-Functions
 - OPC ToolBox

Simulink es una herramienta de programación grafica de Matlab. La misma tiene un conjunto de herramientas y librerías que la hacen muy interesante y muy sencilla de usar a la hora de simular, crear y testear tanto modelos dinámicos lineales y no lineales como sistemas de control.

Algunas de dichas herramientas son:

- StateFlow – Esta herramienta permite generar máquinas de estados las cuales son muy útiles a la hora de programar tomas de decisiones en distintas situaciones y cambios de modos de operación.
- Truth Table – Con esta herramienta se pueden realizar tablas de verdad de un conjunto de variables que corren en Simulink. Por ejemplo con estas tablas de verdad se pueden crear interlocks para las distintas acciones programadas de manera muy sencilla y optima.
- OPC – Tanto Matlab como Simulink contienen librerías para el manejo de OPC (OPC Cliente). Esta librería es de muy fácil uso en Simulink, además cuenta con la ventaja de que si se usa esta librería, Simulink, se ve obligado a correr en tiempo real, siendo una de las principales restricciones a la hora de crear sistemas de control.
- S-Functions – La principal ventaja de S-Functions es que permite escribir ecuaciones diferenciales de manera muy sencilla e intuitiva. Esta es una herramienta muy interesante a la hora de resolver ecuaciones diferenciales ya que genera código en C para optimizar el tiempo y uso de recursos de la PC.

Estas son las herramientas de Simulink a las cuales dimos mayor importancia porque fueron utilizadas para realizar la programación nuestro proyecto.

Para el uso de las herramientas antes descritas se adjuntan los manuales para aquellos que quieran aprender a usar y a profundizar en estas herramientas:

- [-Getting Started Simulink](#)
- [-Using Simulink](#)
- [-Getting Started StateFlow](#)
- [-Using stateFlow](#)
- [-Using Excel Link](#)
- [-Writing S-Functions](#)
- [-Using OPC ToolBox](#)

E.ANEXO: ‘Procedimiento de instalación del sistema 800xA en una PC’

E.1 Usuarios del Sistema

Los siguientes usuarios han sido definidos por requerimientos del sistema de manera tal de implementar políticas de seguridad:

Usuarios de administración:

Nombre	Logon	Clave	Grupos Windows	Roles IIT
Usuario para servicios 800xA	Engineer	engineer	IndustrialITUser IndustrialITAdmin	Everyone Application Engineer System Engineer
Usuario de ingeniería ABB	EngineerPP	engineerpp	IndustrialITUser IndustrialITAdmin	Everyone Application Engineer System Engineer Operator

Usuarios de operación:

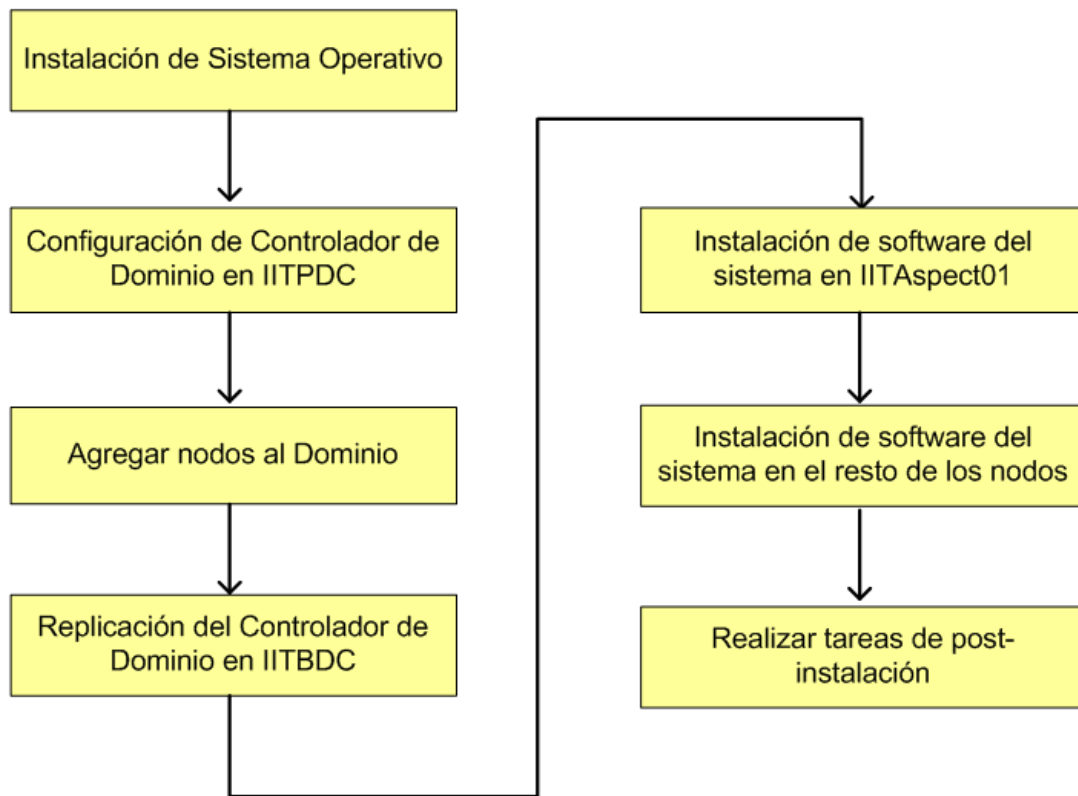
Nombre	Logon	Clave	Grupos Windows	Roles IIT
Usuario de ITC	ITC	itc	IndustrialITUser	Everyone Operator

E.2 Instalación

A la hora de instalar los nodos para el sistema, y considerando la arquitectura del sistema (1 sólo PC), existe una posibilidad de acuerdo al software requerido:

1. Servidor de aspectos primario y ‘connectivity’

Se indicarán los requerimientos y procedimientos necesarios, así como el siguiente procedimiento global para la instalación del conjunto.



E.2.1 Sistema Operativo

Según las recomendaciones, los sistemas operativos a instalar son Windows 2003 Server SP1 o Windows XP Professional SP2. En la siguiente tabla se indica el sistema operativo requerido según el tipo de nodo.

La instalación del sistema operativo es el primer paso en la instalación de cualquier tipo de nodo. Previo a ello, mediante una herramienta de particionamiento de discos como el Partition Magic, se deberán crear dos particiones: una primera partición NTFS donde se instalará el sistema operativo y una segunda partición FAT32 donde se guardarán imágenes Ghost de respaldo, etc. Es recomendable asignar la mitad del espacio del disco a cada partición (suponiendo discos de capacidad mayor o igual a 20 Gb).

Algunas consideraciones generales a tener en cuenta durante la instalación del sistema operativo:

- El sistema 800xA soporta únicamente sistemas operativos en inglés.
- Durante la instalación de Windows 2003 Server, seleccionar el tipo de licenciamiento 'Per Device' o 'Per User'.
- Configuración de Regional Settings: US English Locale, el punto (.) como símbolo decimal.
- No preocuparse en este punto por la configuración de los dispositivos de red o el agregado de un nodo a un dominio.
- Como cuenta de administrador local de la máquina elegir 'administrator' con clave 'aseabrown'.

- Tras finalizar la instalación de Windows 2003 Server, cerrar el diálogo: 'Manage your Server' que se despliega en el primer log-on.

Los drivers específicos de cada máquina, que no hayan podido instalarse automáticamente, deberán cargarse luego de la instalación del sistema operativo.

Algunas consideraciones a tener en cuenta luego de haber finalizado la instalación del sistema operativo:

- Configuración de memoria virtual. Para el caso de nodos cliente, setearla en 1.5 Gb como mínimo. Para servidores, la regla usual es configurar la memoria virtual en 2.5 veces la memoria RAM física disponible.
- Deshabilitar: 'Show Window Contents While Dragging' en el menu: 'Display Properties' > 'Effects'.
- Deshabilitar las configuraciones de 'Energy Saving' y el 'Screen Saber'. Estas configuraciones son por usuario, por lo que tendrán que ser realizadas nuevamente cuando los usuarios del sistema hayan sido creados.
- Deshabilitar el 'Windows Firewall'. Puede realizarse desde el 'Control Panel'. Esta configuración es por usuario, por lo que tendrá que ser realizada nuevamente cuando los usuarios del sistema hayan sido creados.
- Deshabilitar los "Automatic Updates". Puede realizarse desde el 'Control Panel'.
- Para instalaciones de Windows XP Professional SP2:
 1. Abrir el Control Panel
 2. Seleccionar 'Administrative Tools'
 3. Seleccionar 'Local Security Policy'
 4. Seleccionar 'Security Options'
 5. Seleccionar 'Network Access: Sharing and security model for a local account' y elegir la opción 'Classic – local users authenticate as themselves'
- Instalación de 'Internet Information Services'. Se requiere en todos los casos salvo para los clientes y el servidor de dominio.

Para Windows XP Professional SP2 debe instalarse desde: Control Panel > Add/Remove Programs > Add/Remove Windows Components. Seleccionar 'Internet Information Services (IIS)' y verificar que las siguientes opciones estén seleccionadas: 'Common Files', 'Internet Services Manager', 'World Wide Web Service'.

Para Windows 2003 Server SP1 debe instalarse desde: Control Panel > Add/Remove Programs > Add/Remove Windows Components. Seleccionar 'Application Server' y verificar que las siguientes opciones estén seleccionadas: 'ASP.NET', 'Enable network COM+ Access' e 'Internet Information Services (IIS)'. Luego de instalar dichas componentes, ir a Control Panel > Administrative Tools > Internet Information Services Manager. En el panel izquierdo seleccionar 'Services Extensions' y en el panel derecho seleccionar: 'Active Server – Allow', 'Server Side Includes – Allow' y 'ASP.NET – Allow'.

- Instalación de 'Support Tools' y 'Administration Pack' de Windows 2003 Server. Se requiere solamente para los controladores de dominio.

Para la instalación de las 'Support Tools', ejecutar el archivo 'suptools.msi' en la carpeta SUPPORT/TOOLS del CD de instalación de Windows 2003 Server

Para la instalación del 'Administration Pack', ejecutar el archivo 'adminpack.msi' en la carpeta c:\windows\system32

Microsoft Hotfixes – diversos hotfixes deberán instalarse tanto para Windows 2003 Server SP1, como para Windows XP Professional. Los hotfixes necesarios en cada caso, son continuamente actualizados y publicados en el documento: '*Microsoft Security Updates Validation Status for IIT System 800xA – 3BSE041902*'. El documento se encuentra disponible en 'ABB Solutionsbank' y las actualizaciones son periódicamente reportadas en el Auto-notification de 'ABB Solutionsbank'.

Configuración de nombre del nodo – configurar el nombre de cada nodo de acuerdo a la asignación descripta anteriormente.

Configuración de interfaz de red – para cada nodo se requiere la configuración de la interfaz de red. Dicha configuración consiste en los siguientes parámetros TCP/IP:

1. Dirección IP :172.16.4.11
2. Máscara de subred : 255.255.252.0
3. DNS Primario :
4. DNS Alterno :

E.2.2 Instalación del software de terceros

Algunos elementos de software de terceros, deberán instalarse previo a la instalación del software del sistema. Podrán haber ciertas variaciones entre el software necesario para Windows XP Professional SP2 y Windows 2003 Server SP1, que se indicarán cuando corresponda.

.Net Framework

En todos los nodos del sistema, salvo en el controlador de dominio, deberá seguirse el siguiente procedimiento para la instalación de componentes de .Net Framework. El procedimiento es el mismo para ambos sistemas operativos.

1. Instalar .Net Framework 1.0 ejecutando 'dotnetfx.exe' en: DVD > System1 > Software > Microsoft
2. Instalar el SP3 para .Net Framework 1.0 ejecutando 'NDP1.0sp3-KB867461-X86-ENU.exe' en: DVD > System1 > Software > Microsoft
3. Reiniciar el nodo
4. Instalar .Net Framework 1.1 ejecutando 'dotnetfx1_1.exe' en: DVD > System1 > Software > Microsoft
5. Instalar el SP1 para .Net Framework 1.1 ejecutando 'NDP1.1sp1-KB867460-X86.exe' en: DVD > System1 > Software > Microsoft
6. Reiniciar el nodo

Microsoft Word y Excel

Es recomendable instalar el Word y Excel del paquete Microsoft Office 2003 en todos los nodos del sistema salvo en el controlador de dominio. Luego de instalados, desde cada nodo y para cada usuario configurar el nivel de seguridad: Tools > Macro > Security en: ‘Security Level Medium’.

Visual Basic 6 con SP6

En todos los nodos del sistema, salvo en el controlador de dominio, deberá seguirse el siguiente procedimiento para la instalación del Visual Basic 6 con SP6. El procedimiento es el mismo para ambos sistemas operativos.

1. Instalar Visual Basic 6 con las opciones por defecto. Cuando lo solicite, reiniciar el nodo.
2. Completar la instalación y no instalar las librerías MSDN.
3. Instalar SP6 para Visual Basic 6 ejecutando: ‘setupsp6.exe’ en: DVD > System1 > Software > Microsoft > VisualBasic ServicePack

MSXML

En todos los nodos del sistema, independientemente del sistema operativo, deberá instalarse MSXML 4.0 SP2. Para ello ejecutar: ‘msxml.msi’ en: DVD > System1 > Software > Microsoft.

Acrobat Reader

En todos los nodos del sistema, independientemente del sistema operativo, deberá instalarse el Acrobat Reader 7.0. Para ello ejecutar: ‘AdbeRdr70_ene_full.exe’ en: DVD > System1 > Software > Adobe.

E.2.3 Grupos y usuarios necesarios para correr el sistema

1. Crear grupos de usuarios IndustrialITUser e IndustrialITAdmin, ambos pertenecientes a dicha Organizational Unit.
2. Hacer el grupo IndustrialITAdmin miembro del grupo Administrators.
3. Crear usuarios de acuerdo a los usuarios del sistema definidos anteriormente en este documento.
4. Hacerlos miembros de los grupos IndustrialITUser e IndustrialITAdmin según lo indicado.
5. Crear las Organizational Units – Operadores y Monitoreo dentro de Industrial IT.

E.2.4 Instalación del software del sistema

A continuación se describe el software del sistema 800xA requerido para cada uno de los seis tipos de nodos. Algunos puntos importantes a tener en cuenta:

- La numeración indica el orden en el que deberán ser instalados.
- Nunca reiniciar la máquina por más que el proceso de instalación así lo indique. Sólo deberá reiniciarse al comienzo, luego de instalar el ‘Central Licensing System’

en cada nodo. Para el caso particular del 'Primary Aspect Server', luego del reinicio, conectar el dongle USB para el licenciamiento del sistema e instalar sus drivers abriendo: Start > Programs > ABB Industrial IT 800xA > System > Licensing > License Entry y luego seleccionando: Machine IDs > Install SPro Driver. Se podrá verificar luego que el Machine ID correspondiente al dongle se despliega en la lista Machine IDs > Available IDs.

- En todos los casos en que exista la posibilidad, seleccionar el tipo de instalación 'Complete'. En los otros casos elegir 'Typical'.
- Al instalar el 'Central Licensing System', ingresar la cuenta del usuario creado para correr los servicios del sistema: PPiloto\engineer con su respectiva contraseña.
- Antes de instalar Microsoft SQL Server 2000 Desktop Engine cuando corresponda, también instalar el hotfix para MDAC Q823490.
- Tras instalar el Microsoft SQL Server 2000 Desktop Engine cuando corresponda, también instalar el patch MS03-01.

Deberá instalarse en primer lugar el nodo correspondiente al Primary Aspect Server. Esto se debe a que dicho nodo actuará como 'Central Licensing System Server' distribuyendo licencias al resto. Los demás nodos deberán instalarse como 'Central Licensing System Clients' y en el proceso de instalación se solicita seleccionar el nodo que actúa como servidor y deberá ser accesible a los mismos.

Primary Aspect Server

1. 'Central Licensing System' (DVD\System1\Licensing)
 - a. Instalar el software básico para cliente y servidor
 - b. Instalar la 'Central Licensing System Extension'
2. 'Process Portal' (DVD\System2\Process Portal)
 - a. Instalar el Process Portal A 4.0.0.1
 - b. Instalar el Process Portal A 4.0.0.1 SP1
 - c. Instalar la 'Legacy Graphics Library' (DVD\System2\Legacy Graphics Library)
3. '800xA Instructions' (DVD\Documentation)
4. 'Diagnostics Collection Tool' (DVD\System1\Diagnostic Collection Tool)
5. AC800M (DVD\System3)
 - a. Instalar 'AC800M Connect'
6. 'Engineering Studio' (DVD\Engineering)
 - a. Instalar 'Microsoft SQL Desktop Engine'
 - b. Instalar 'Engineering Platform'
 - c. Instalar 'Topology Designer'
 - d. Instalar 'Function Designer'
7. 'Asset Optimization' (DVD\AssetOpt\Asset Optimization)
 - a. Instalar el software básico para servidor
8. 'PLCC Connect'
 - a. Basic

9. Softpoint
10. IM
 - a. 'Client toolkit'

E.3 Post-instalación

E.3.1 Licenciamiento del sistema

Luego de que todo el software del sistema ha sido cargado en los diferentes nodos, la primera tarea de post-instalación consiste en el licenciamiento del sistema. Como prerrequisito para ello se deberá:

- disponer del archivo xxxxxxxxxxxxxx.sla que se genera en el sitio del 'ABB Software Factory' mediante las licencias asignadas para el sistema 800xA y el número del Machine ID disponible.
- haber instalado correctamente los drivers para el 'USB Rainbow Superpro Dongle'.

El procedimiento a seguir para licenciar el sistema es el siguiente:

1. Copiar el archivo xxxxxxxxxxxxxx.sla en la nueva carpeta Licencias 800xA de la partición FAT32 del 'Primary Aspect Server'
2. Abrir: Start > Programs > ABB Industrial IT 800xA > System > Licensing > License Entry
3. Verificar que en: Machine IDs > Available IDs se encuentra el número escrito en dongle USB
4. Seleccionar: File > Load/Replace Licenses y buscar el archivo: xxxxxxxxxxxxxx.sla en la carpeta correspondiente. Confirmar la selección.
5. Verificar que las licencias que se despliegan corresponden a las asignadas en 'ABB Software Factory'.

En cualquier momento y desde cualquier nodo se podrá verificar si existen problemas de licenciamiento, es decir falta de disponibilidad de licencias. Para ello abrir Start > Programs > ABB Industrial IT 800xA > System > Licensing > License Status Viewer. En caso de alguna violación de licencia o asignación de licencia temporal debido a que el nodo no tiene conectividad con el 'Central Licensing Server', se listarán allí dichas violaciones. Una pantalla blanca indicará que no hay problemas de licenciamiento.

E.3.2 Creación del sistema

Desde el 'Configuration Wizard' en el 'Primary Aspect Server', deberá crearse el sistema sobre el que luego se realizará el upgrade del sistema existente. Para ello, llevar a cabo el siguiente procedimiento:

1. Seleccionar: Configuration Wizard > Create System
2. Definir los atributos del nuevo sistema. 'Name': PPilotoxA, 'Description': Sistema de Control de la Planta Piloto, 'Server Type': Aspect and connect Server
3. Confirmar los directorios asignados para el almacenamiento de los archivos del sistema

4. Habilitar el uso de RNRP con los siguientes parámetros. 'Number of areas':1, 'Primary Network Address': 172.16.4.0, 'Network Address Mask':255.255.252.0, 'Secondary Network Address':...
5. Confirmar y guardar que se complete el procedimiento de creación del sistema. Verificar en el log de que no se hayan producido errores.

E.3.3 Carga de extensiones

Una vez creado el sistema, deberán cargarse las extensiones requeridas de acuerdo al software instalado y la funcionalidad buscada.

Las extensiones se cargan solamente una vez y en un único nodo, el 'Primary Aspect Server'. Para ello, desde el Configuration Wizard > System Administration > PPilotoxA > System Extension Load.

Las extensiones pueden cargarse todas juntas en un único proceso. Para ello deberán seleccionarse todas las extensiones que se listan a continuación, y en dicho orden para cumplir con las dependencias, antes de confirmar la selección.

1. AC800M Connect
2. AC800M SB2 Libraries
3. Asset Optimizer Server
4. Asset Monitor Enviroment
5. Engineering Base
6. DM & PM Application
7. AC800M Signal Extensions
8. Function Designer
9. Topology Designer
10. Function Designer for AC800M
11. Topology Designer for AC800M
12. Function Designer for AC800M SB2 Libraries
13. Central Licensing System
14. PLCC
15. Softpoint
16. Scheduling
17. Calculation
18. System Instructions

E.4 Configuración y puesta a punto del sistema

E.4.1 Registro de usuarios del sistema

Los diferentes usuarios Windows indicados anteriormente, que se utilizarán en el sistema, deben registrarse como usuarios del mismo. Para ello, llevar a cabo el siguiente procedimiento desde el 'Primary Aspect Server'.

1. Abrir el 'Configuration Wizard'
2. Seleccionar: System Administration > PPilotoxA > Users

3. Presionar el botón 'Add Windows Accounts' para agregar los usuarios Windows al sistema
4. Seleccionar la lista de usuarios a agregar. Además tener en cuenta que los usuarios Engineer y EngineerPP ya han sido agregados al sistema automáticamente.
Confirmar la selección
5. Para cada uno de los nuevos usuarios que se agregan, seleccionar los roles que le correspondan, pasándolos de la ventana 'Not member of' hacia 'Member'. También asignar el rol 'Operator' al usuario 'TTC'.
6. Seleccionar 'Next' y luego "Finish" para aplicar los cambios

E.4.2 Configuración de servicios

Servicio de Asset Optimization

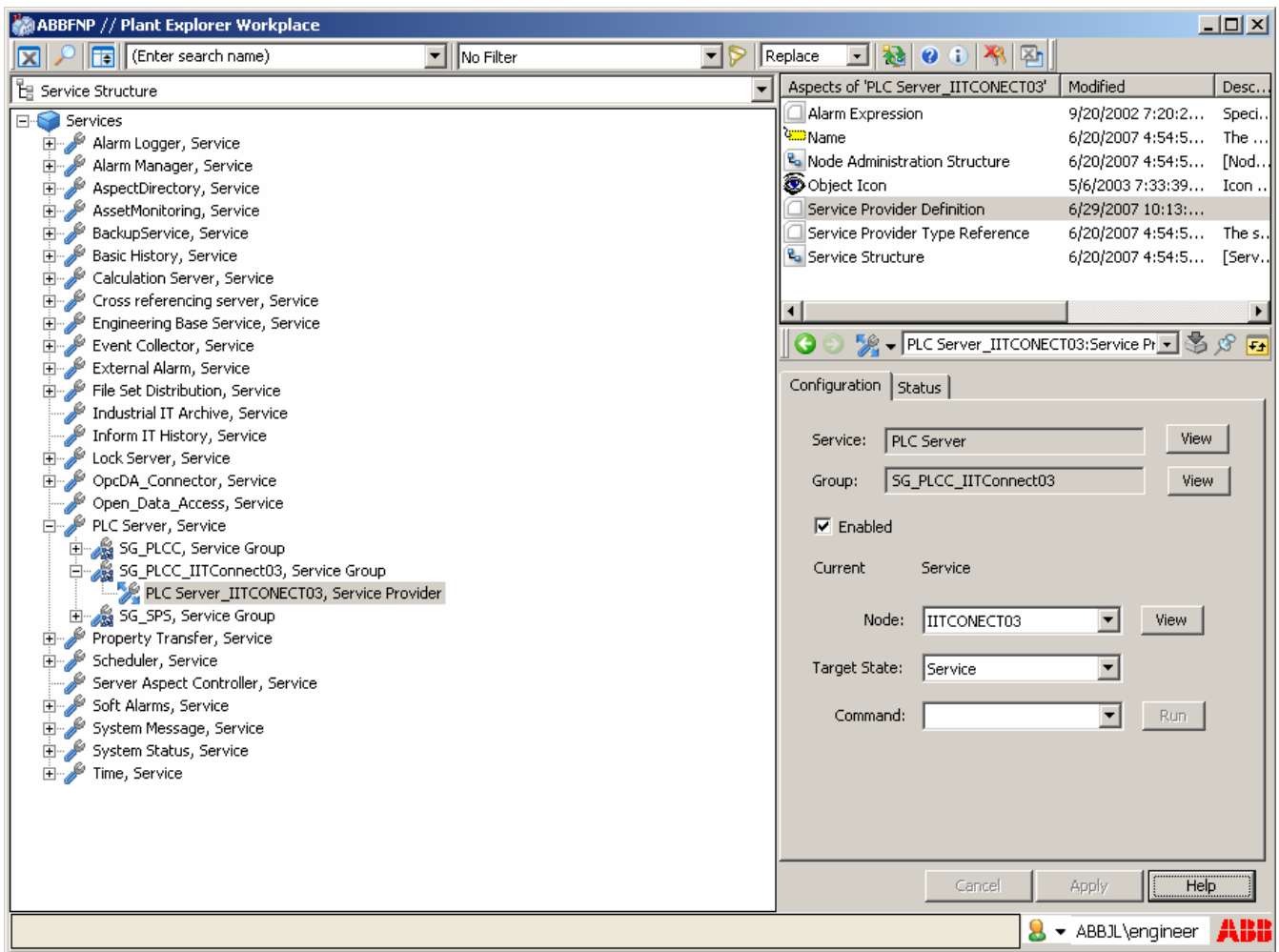
El proveedor del servicio Asset Monitoring debe configurarse manualmente. Para ello, seguir el siguiente procedimiento que puede realizarse desde la estación de ingeniería:

1. En la 'Service Structure' acceder al aspecto 'Service Provider Definition' del objeto del tipo 'Service Provider' bajo el servicio 'AssetMonitoring'
2. En la solapa 'Configuration' seleccionar: PPiloto en 'Node' y confirmar los cambios
3. Verificar que el servicio entra en 'Service State' luego de dicho cambio

Servicio PLCC

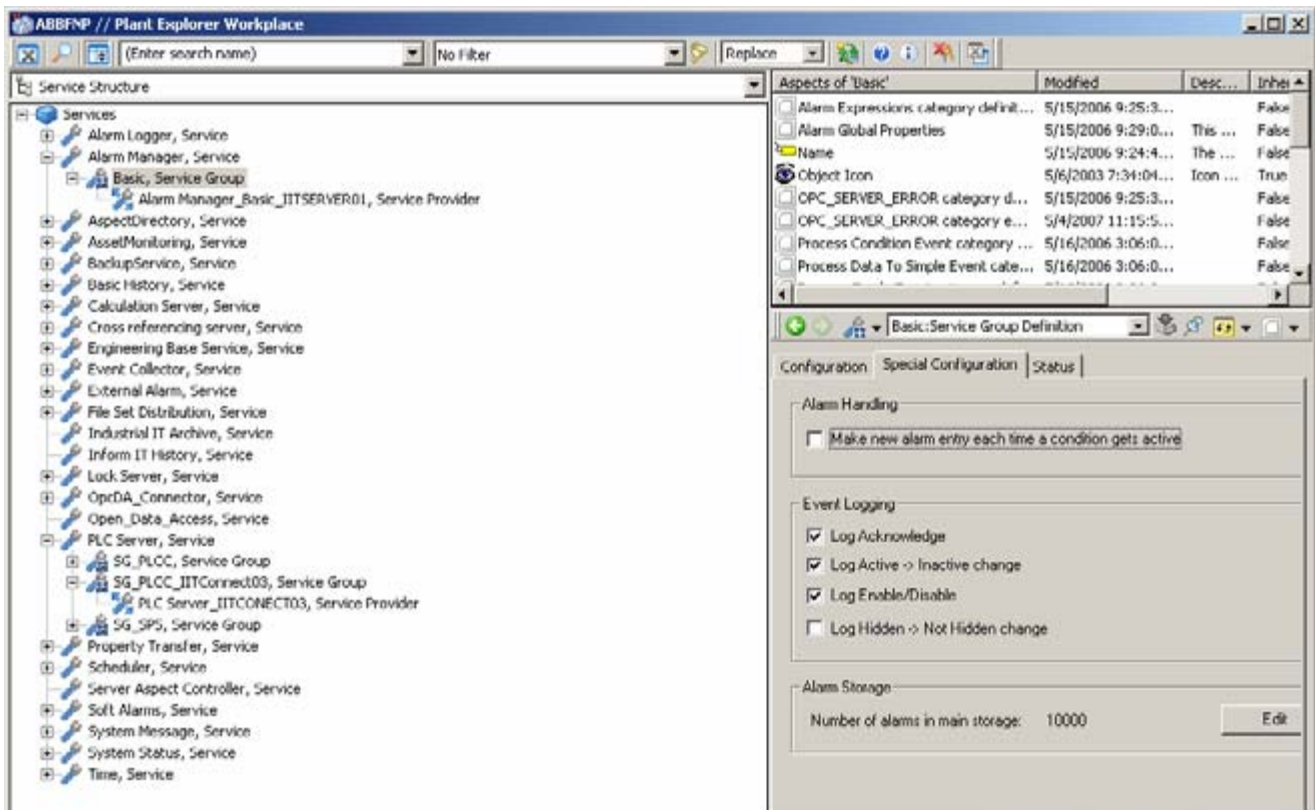
Verificar que se haya creado un 'Service Group' y 'Service Provider' en el servicio 'PLC Server' que corra en la maquina PPiloto

- 1- Si están creados, renombrarlos de la siguiente manera
 - a. Service Group: SG_PPiloto
 - b. Service Provider: SP_PPiloto



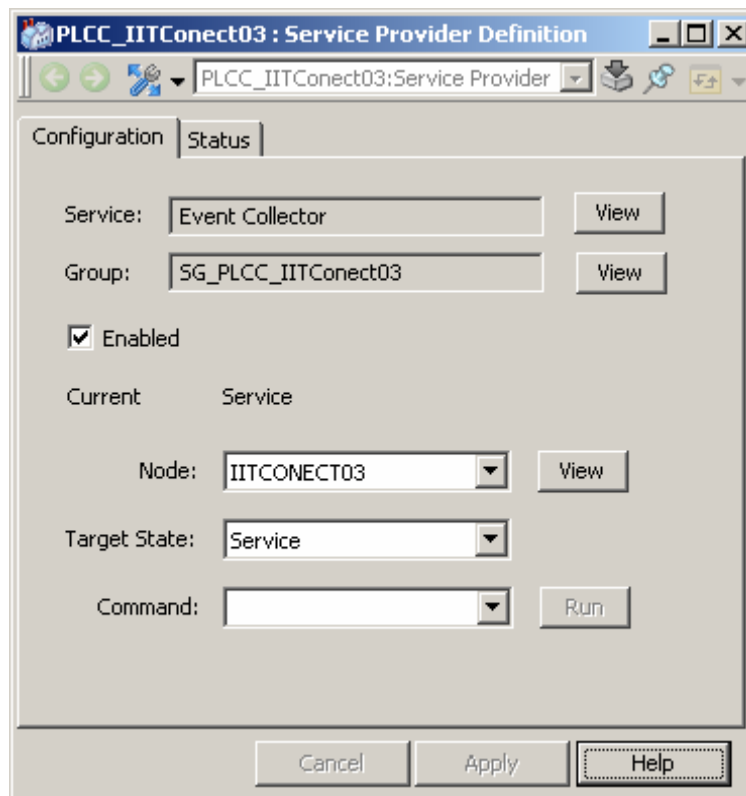
2- Configurar los servicios de alarma.

- a. En el servicio 'Alarm Manager' hacer la configuración que se muestra en la figura de abajo:

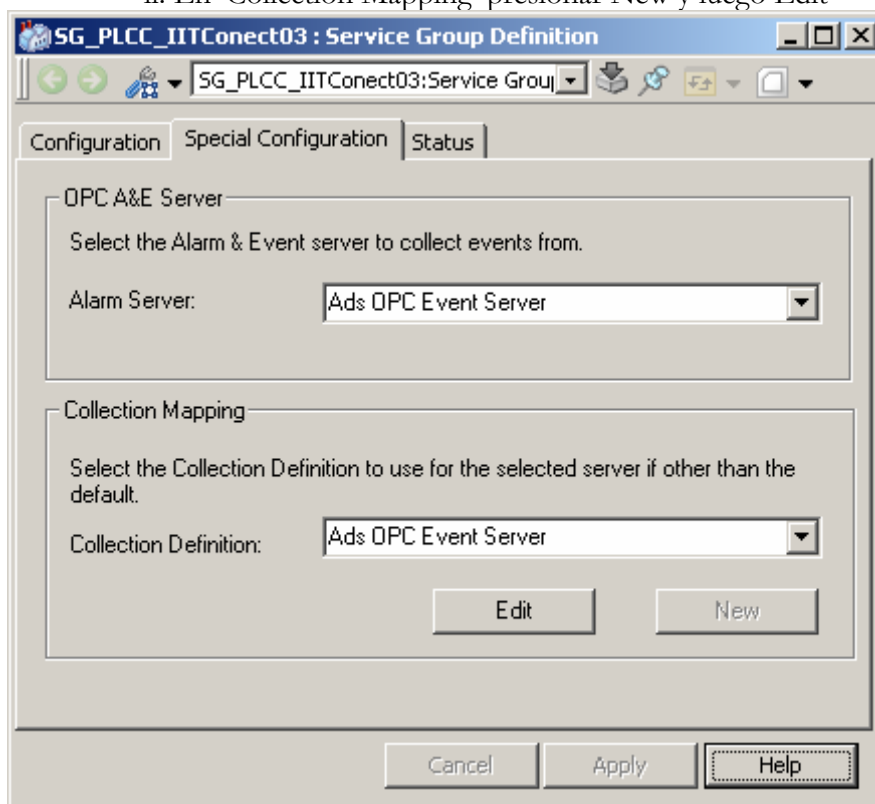


3- Configurar el servicio 'Event Collector' de la siguiente manera:

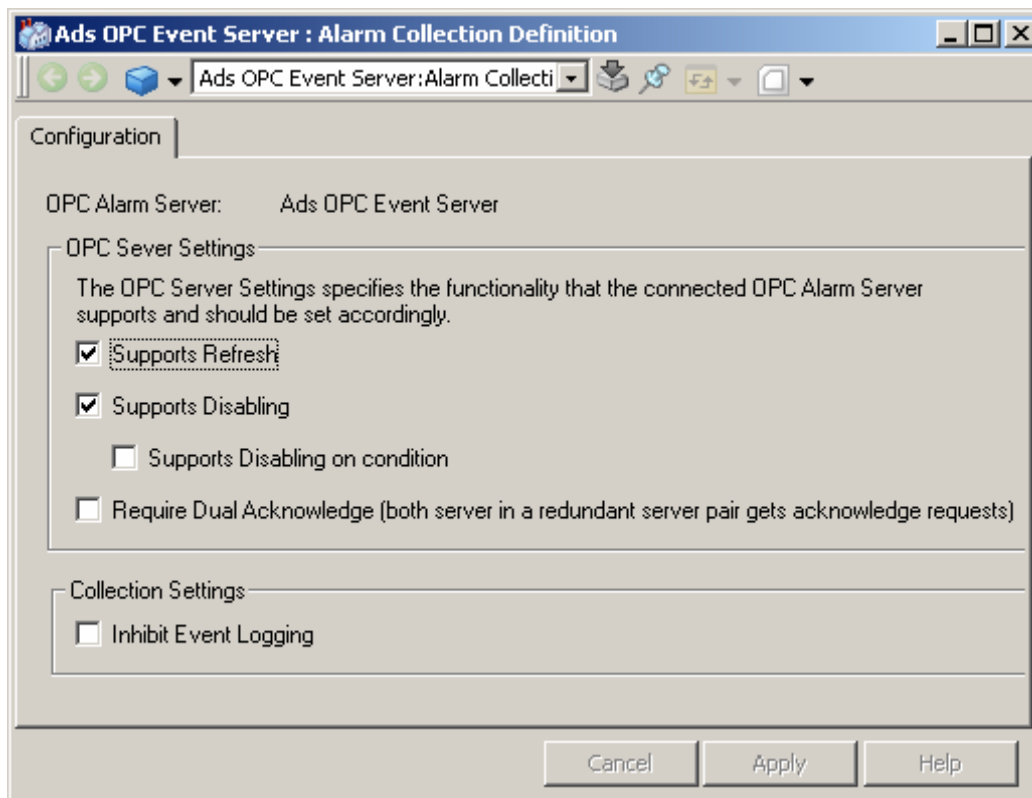
- Crear un 'Service Group' con el Nombre SG_PPiloto(SG_IITAspect03)
- Crear un 'Service Provider' con el nombre SP_PPiloto(SG_IITAspect03)
- Configurar el 'Service Provider' de la siguiente manera:



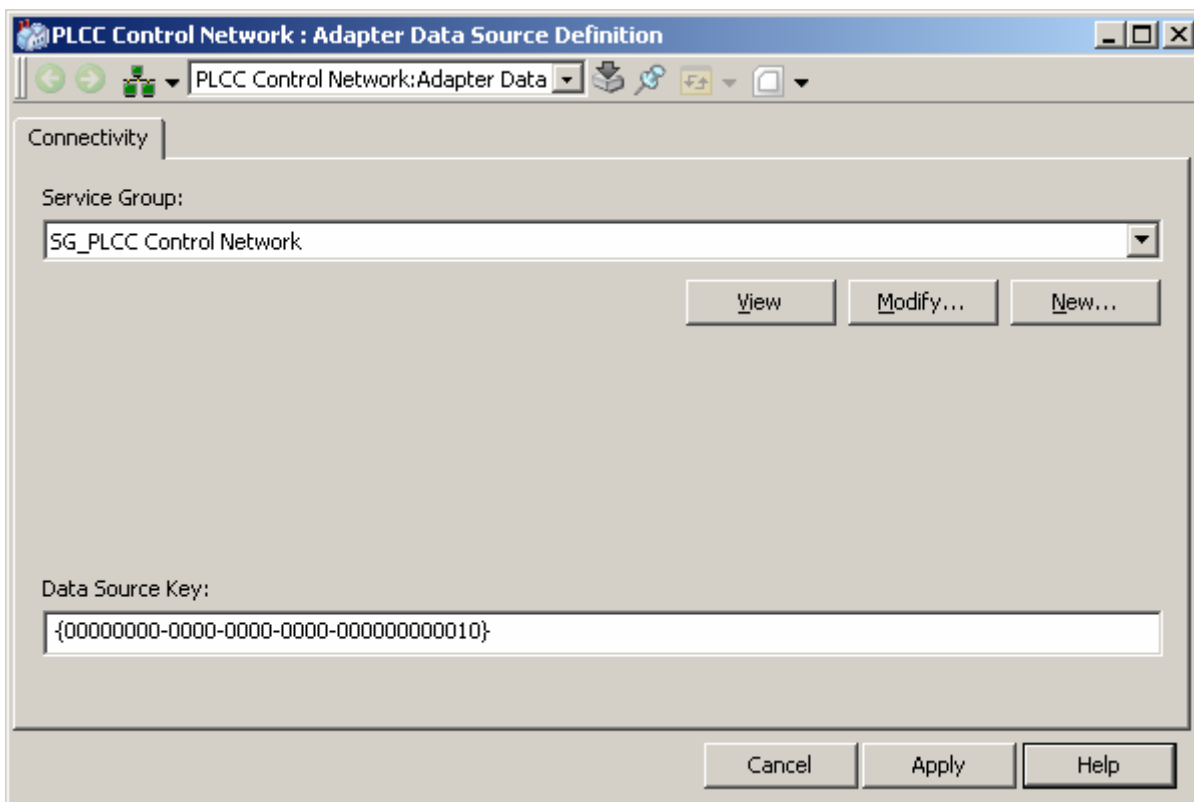
- d. Realizar la siguiente configuración en la solapa ‘Special Configuration’ del ‘Service Group’ antes creado:
- i. En ‘Alarm Server’ seleccionar: ‘Ads OPC Event Server’
 - ii. En ‘Collection Mapping’ presionar New y luego Edit



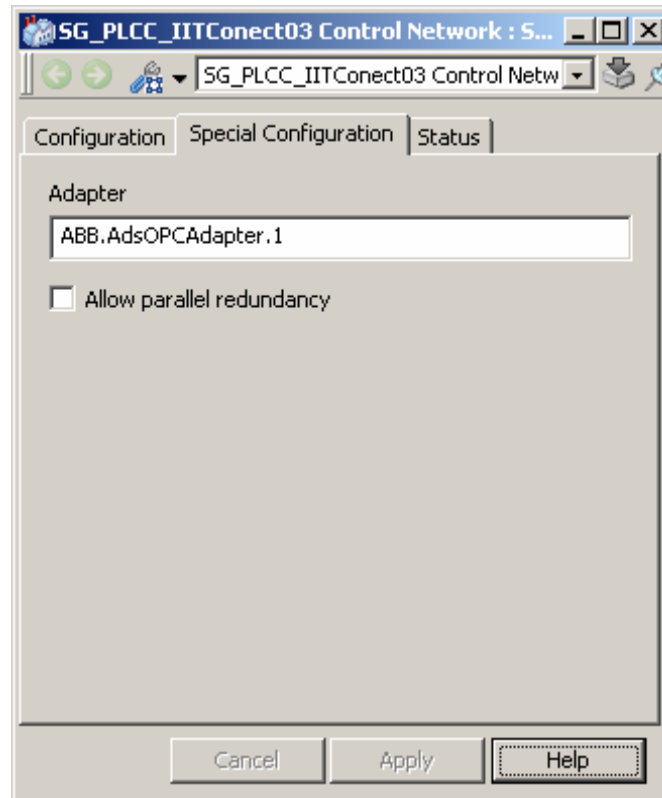
- iii. Seleccionar solamente: ‘Supports Refresh’ y ‘Support Disabling’



- 4- En la 'Control Structure' crear un nuevo objeto 'PLC Generic Control Network' y nombrarlo: 'PLCC PPiloto Control Network'.
- 5- En el objeto 'PLCC PPiloto Control Network' crear un aspecto 'Adapter Data Source Definition' y configurarlo como muestra la siguiente figura:

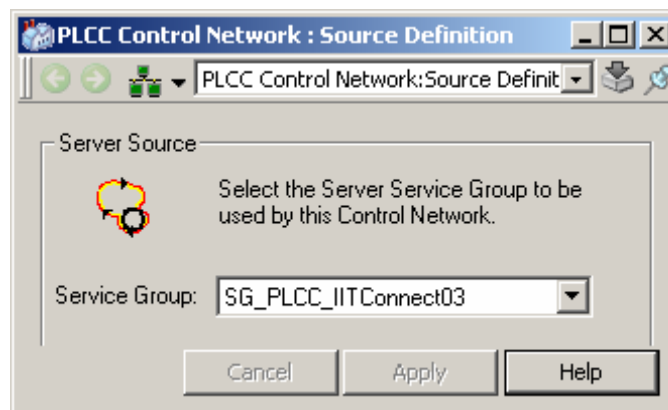


- 6- En el nuevo 'service provider' del servicio de 'OpcDA_Connector' hacer la siguiente Configuración
 - a. En la solapa 'Special Configuration' cambiar el ProgID de {00..0000} a {00..0010} ({00..001})y presionar Apply
 - b. En la solapa 'Configuration' seleccionar PPiloto en Node y presionar Apply
- 7- En el nuevo 'Service Group' creado hacer la configuración que se muestra en la siguiente figura:



8- Una vez terminado el paso anterior renombrar el 'Service Group' como:
 'SG_PLCC_PPiloto Control Network'

9- En el objeto 'Piloto Control Network' de la 'Control Structure' crear un aspecto:
 'Source Definition' y configurarlo como muestra la siguiente figura:



10- Configurar el servicio de históricos correspondiente al PPiloto

11- Crear y configurar el aspecto: 'History Source' en las dos nuevas 'Control Networks'

Bibliografía

- (ABB, 2001) ABB, “*Technical Manual Advant Controller 31*”, www.abbplc.com/pdf/series4050hard.pdf, 25/09/2001.
- (Alfaro Ruiz, 2001) Alfaro Ruiz V. M., “Identificación de Procesos Sobreamortiguados Utilizando Técnicas de Lazo Cerrado”, www.eie.ucr.ac.cr/uploads/file/documentos/pub_inv/articulos/valfaro01A.pdf, Facultad de Ingeniería de San José, Vol. 11, Costa Rica, 2001.
- (Alfaro Ruiz, 2005) Alfaro Ruiz V. M., “Actualización del Método de Sintonización”, www.eie.ucr.ac.cr/uploads/file/documentos/pub_inv/articulos/valfaro05A.pdf, Facultad de Ingeniería de San José, Vol. 15, Costa Rica, 2005.
- (Antonio Visioli, 2006) Antonio Visioli, “*Practical PID Control*”, Springer-Verlag, 2006.
- (Åström, 1995) Åström Karl Johan & Tore Hagglund, “*PID Controllers: Theory, Design, and Tuning.*”, International Society for Measurement and Con., 2nd Edition, 1995.
- (Åström, 2002) Åström Karl Johan, “*Control System Design Lecture notes for ME 155A*”, <http://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom.html>, Department of Mechanical & Environmental Engineering University of California, 2002.
- (Ballester, y otros, 2001) Ballester Grettel y Mora Erick, “Introducción al Control Adaptivo”, Universidad de Costa Rica : Facultad de Ingeniería, 2001.
- (Berenfus, Casas, & Rezk, 2006) Berenfus Viviana, Casas Natalia y Rezk Gabriela, “*Planta Piloto para Control de Procesos*”, Facultad de Ingeniería del Uruguay, 2006.
- (Blanco, 2005) Blanco Rocío, “Tecnologías Para La Captura De Información De Líneas De Producción – OPC”, www.eueti.uvigo.es/files/material_docente/355/opc_2.pdf, 2005.
- (Jiménez, 2001) Jiménez Buendía M., “Comunicaciones Industriales”, http://www.dte.upct.es/personal/manuel.jimenez/docencia/GD6_Comunic_Ind/pdfs/Tema%207.pdf, Universidad Politécnica de Cartagena, Departamento de Tecnología Electrónica, 2001.
- (Crane, 1999) Crane, “*Flow Of Fluids-Technical Paper No.410M*”, Mc. Graw Hill, 1999.
- (D’Azzo, 1965) D’Azzo J., “*Sistemas Realimentados de Control*”, Parainfo, 2da Edición, 1965.

- (Espinosa, 2006) J. M. Espinosa, “Historia y actualidad de los mecanismos Antiwindup”, Universidad Politécnica de Madrid, 1/06/2006.
- (Ferrari, 2005) Ferrari Juan Pablo, “Sistemas de Control Distribuido”, <http://www.monografias.com/trabajos-pdf/sistemas-de-control-distribuido/sistemas-de-control-distribuido.pdf>, Universidad Nacional de Rosario, 2005
- (Moreno, 2001) Moreno Mauricio A., “*Apuntes de control PID*”, www.alumnos.usm.cl/~ignacio.morande/descargas/apuntes_de_control_pid.pdf, Universidad Mayor de San Andrés, La Paz, Bolivia, 2001.
- (Levine, 1996) Levine William S., “*The Control Handbook*”, CRC PRESS in cooperation with IEEE PRESS, 1996.
- (Ogata, 2001) Ogata Katsuhiko, “*Modern Control Engineering*”, Prentice Hall, 4th Edition, 2001.
- (Rasmussen, 2002) Rasmussen H, “Automatic Tuning of PID-regulators”, www.control.auc.dk/~hr/elite_2007/auto.pdf, Aalborg University (Dept. of Control Engineering), Dinamarca, 6/9/2002.
- (Resnick y otros, 2001) R. Resnick, D. Halliday & K. S. Krane, “*Física Vol. 1*”, Tercera Edición, Ed. Continental, 2001.
- (Sastry & Bodson, 1989) Sastry Shankar y Bodson Marc, “*Adaptive Control- Stability, Convergence, and Robustness*”, Prentice-Hall, 1989.
- (Vidal, y otros, 2002) Vidal Pablo y Figueroa Sergio, “*OPC: Un Estandar En Las Redes*”, www.senacitel.cl/downloads/senacitel2002/TE001.pdf, 12/06/2002.
- (Wik) Wikipedia, www.wikipedia.org.

Prácticas de Laboratorio

Introducción a la Teoría de
Control

Planta Piloto v2

Facultad de Ingeniería
Universidad de la República

INDICE

1 INTRODUCCIÓN

1 Introducción.....	3
1.1 Objetivos y descripción del Laboratorio	3
1.2 Descripción de la planta.....	3
1.3 Comunicación de datos Planta-Matlab.....	7

2 GUÍA de Operación de la PLANTA

2 Guía de operación de la Planta	9
2.1 Pasos a seguir para iniciar la planta	9
2.2 Guía rápida de utilización del Sistema	10
2.2.1 Como ingresar al sistema de control	10
2.2.2 Uso de los Controladores.....	12
2.2.2.1 Objeto Controlador (LIC o TIC).....	12
2.2.2.2 Faceplates de los Controladores.....	12
2.2.2.3 Modos de funcionamiento del Controlador	13
2.2.2.3.1 Modo Manual	13
2.2.2.3.2 Modo PID Automático	14
2.2.2.3.3 Modo Control por MR.....	15
2.2.2.4 Autotuning PID.....	17
2.2.2.5 Identificación	18
2.2.2.6 Botones Auxiliares	19
2.2.3 Otros objetos implementados	19
2.2.3.1 Bomba	19
2.2.3.2 Resistencia	20
2.2.3.3 Válvula de realimentación	20
2.2.3.4 Indicaciones	20
2.3 Exportación de Históricos a planillas de Excel	21

3 PRÁCTICAS

3 Prácticas.....	22
3.1 Práctica 0: Uso de Matlab y de la Planta Piloto	22
3.1.1 Objetivo.....	22
3.1.2 Ejercicios	22
3.2 Practica 1: Modelado e identificación de parámetros.....	23
3.2.1 Objetivo.....	23
3.2.2 Modelado de la planta	23
3.2.3 Ensayos a realizar.....	24
3.2.4 Identificación	25
3.3 Práctica 2: Simulación y diseño de controladores asistido por computadora.....	26
3.3.1 Objetivo.....	26
3.3.2 Simulación de los controladores.....	26
3.4 Práctica 3: Implementación y evaluación de controladores.....	29
3.4.1 Objetivo.....	29
3.4.2 Implementación de Controladores.....	29
3.4.3 Ensayo.....	29
Bibliografía	31

1 Introducción

1.1 Objetivos y descripción del Laboratorio

Este laboratorio tiene como objetivo familiarizar al estudiante con técnicas experimentales básicas del control e introducirlo en el manejo de un sistema de control comercial utilizado ampliamente en el mercado.

El laboratorio se divide en varias prácticas, integrando un único proceso de ingeniería en el cual se busca controlar una planta física.

Como objetivo final de este proceso se busca que el estudiante pueda poner en práctica controladores y comparar su desempeño con la teoría.

También se busca que el estudiante compare métodos de auto sintonía con los métodos aplicados en ITC.

Las diferentes prácticas buscarán completar, de forma ordenada, diferentes instancias del proceso de ingeniería de control:

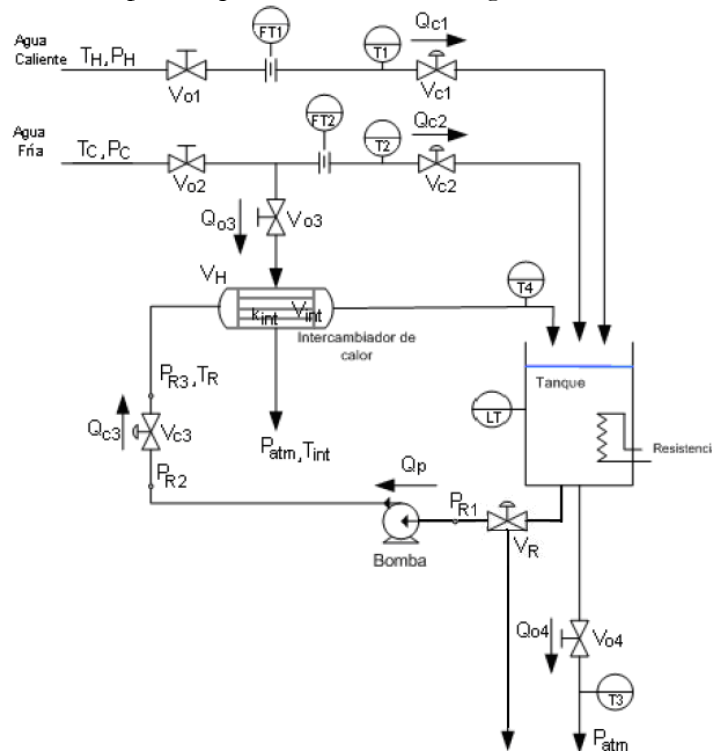
- Practica 0: Uso de Matlab y de la Planta Piloto
- Practica 1: Modelado (del proceso de nivel dentro del tanque) e identificación de parámetros
- Práctica 2: Simulación y diseño de controladores asistido por computadora
- Práctica 3: Implementación y evaluación de controladores

Material a traer por el estudiante:

- Pendrive (verificar antes de conectar cualquier dispositivo al PC que el mismo se encuentre libre de software malicioso)

1.2 Descripción de la planta

El sistema a modelar es la planta que se muestra en la figura a continuación.



Esquema de la Planta Piloto

La planta consta de un tanque, un intercambiador de calor, una bomba, tres válvulas de control, una válvula ON-OFF de 3 vías, dos caudalímetros, cuatro sensores de temperatura y un sensor de nivel. Dichos componentes están interconectados por medio de cañerías y accesorios.

El tanque es un recipiente abierto, con una resistencia interna de 1200W. En dicho tanque ingresan agua fría, agua caliente y agua tibia.

El agua fría proviene directamente de OSE, por lo tanto la presión a la que es suministrada está determinada por dicha institución en aproximadamente 3bar. Por otro lado el agua caliente es agua proveniente de OSE calentada en un termotanque a 70°C aproximadamente. El agua tibia es agua que sale del tanque y es enfriada mediante el intercambiador de calor por el cual circula agua fría con este propósito, también proveniente de la red de OSE. Dicha recirculación se realiza mediante la bomba.

Los tres caudales que ingresan al tanque son controlados por las válvulas de control correspondientes a cada uno de ellos. El caudal de agua caliente se controla con la servoválvula Vc1, el frío con Vc2 y la realimentación con Vc3.

Además de las válvulas de control existe en la planta una válvula ON-OFF (VR) de 3 vías encargada de habilitar la recirculación (si la bomba está encendida) o permitir el desagotar del tanque.

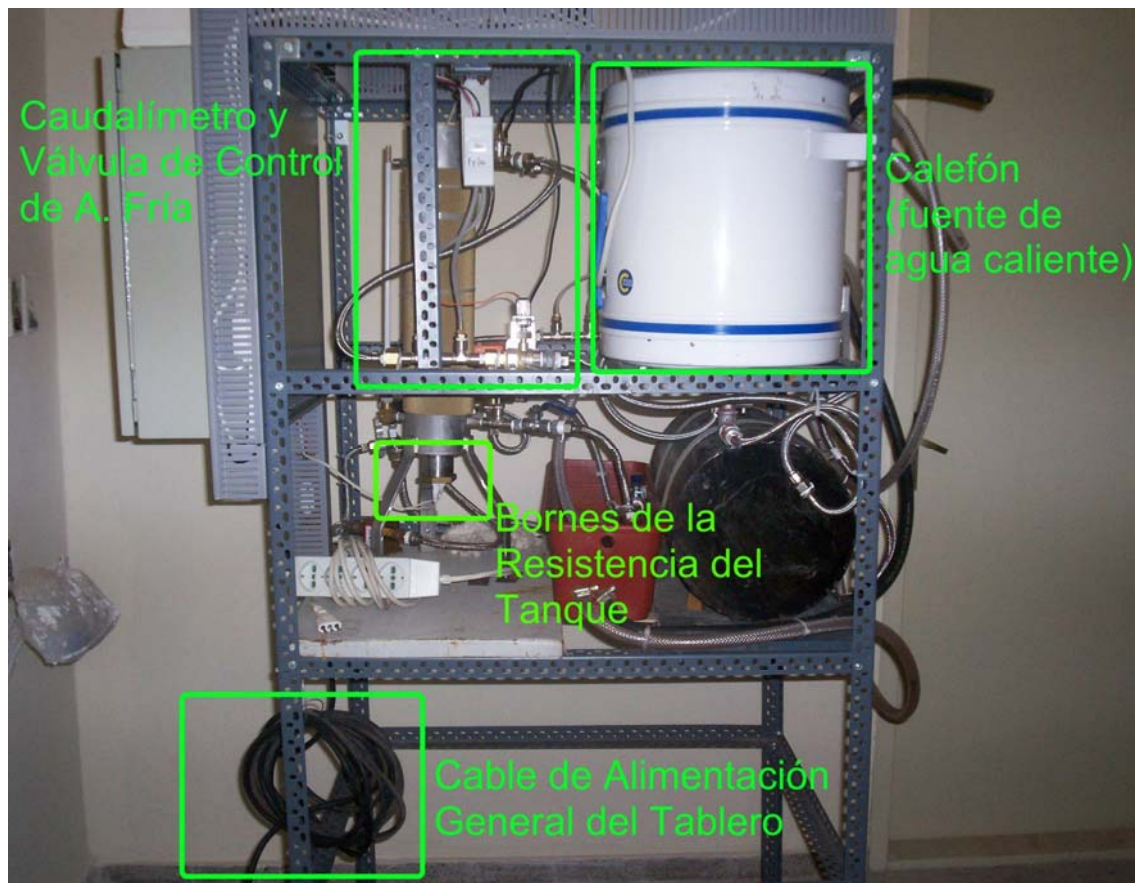
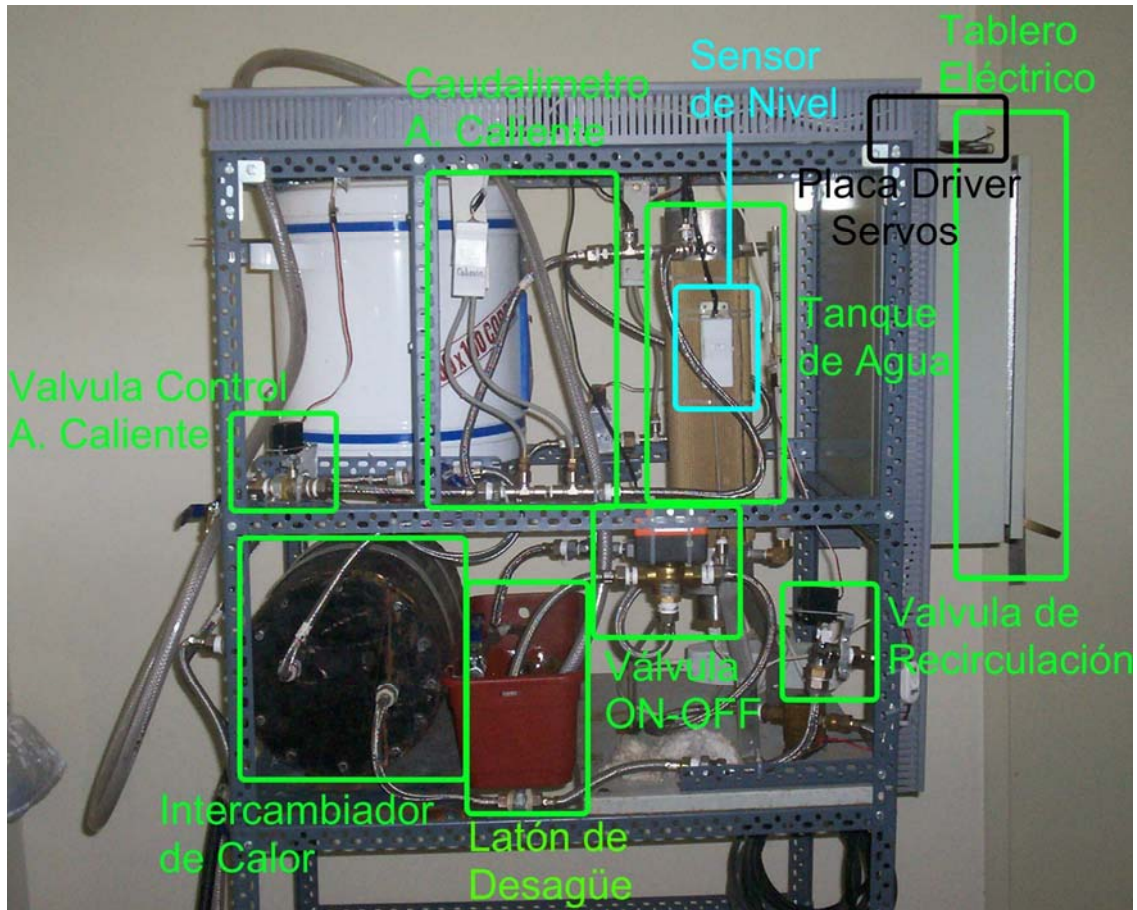
Las válvulas manuales Vo1 y Vo2 se utilizan para limitar los caudales máximos de agua caliente y agua fría respectivamente (éstas no son válvulas de control). El pasaje de caudal a través de la carcasa del intercambiador de calor se habilita mediante la apertura de la válvula Vo3 y finalmente la salida del tanque es manejada por una válvula (Vo4). Existe también una válvula manual general de entrada a la planta, la cual sirve para limitar el caudal máximo de entrada.

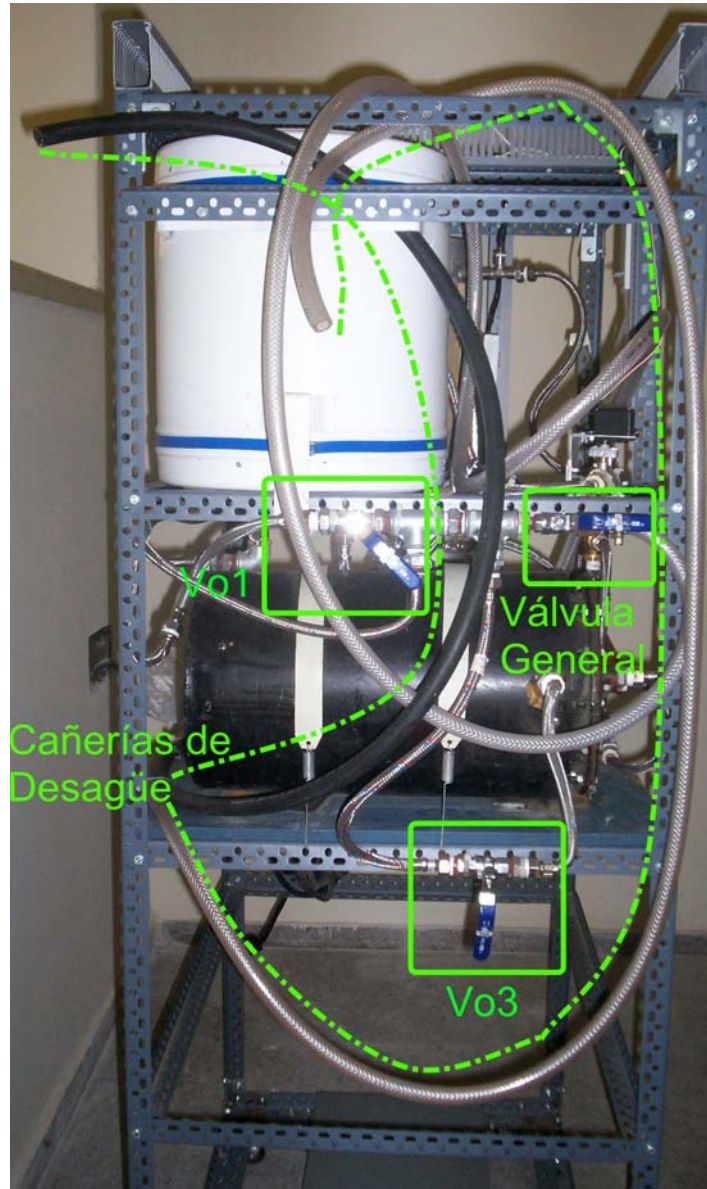
En lo que se refiere a los sensores dispuestos a lo largo del circuito de circulación de agua existen cuatro sensores de temperatura para medir las temperaturas de agua caliente (T1), agua fría (T2), agua enfriada por el intercambiador de calor (T4) y salida del tanque (T3). Además dos caudalímetros miden el caudal de agua caliente (FT1) y fría (FT2) que ingresan al tanque. Para medir el nivel de agua dentro del tanque hay un sensor de nivel (LT).

Los actuadores (servos) así como los sensores (RTD, caudalímetros, sensor de nivel) están conectados a un PLC, el cual se encarga de leer las señales, transferirlas al sistema de control y escribir en los actuadores el resultado del procesamiento.

El procesamiento de estas señales se realiza en Matlab y se utiliza el DCS 800xA de ABB como interfaz de comunicación entre PLC-Matlab, además de ser el encargado de proveer una interfaz amigable y almacenar los históricos de los experimentos realizados.

A continuación se presentan fotos con detalles de la mayoría de las componentes de la planta:





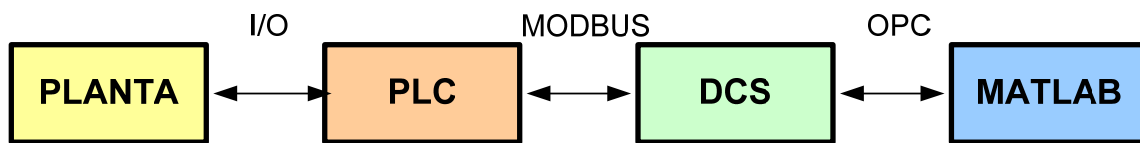
Bomba con habilitación manual



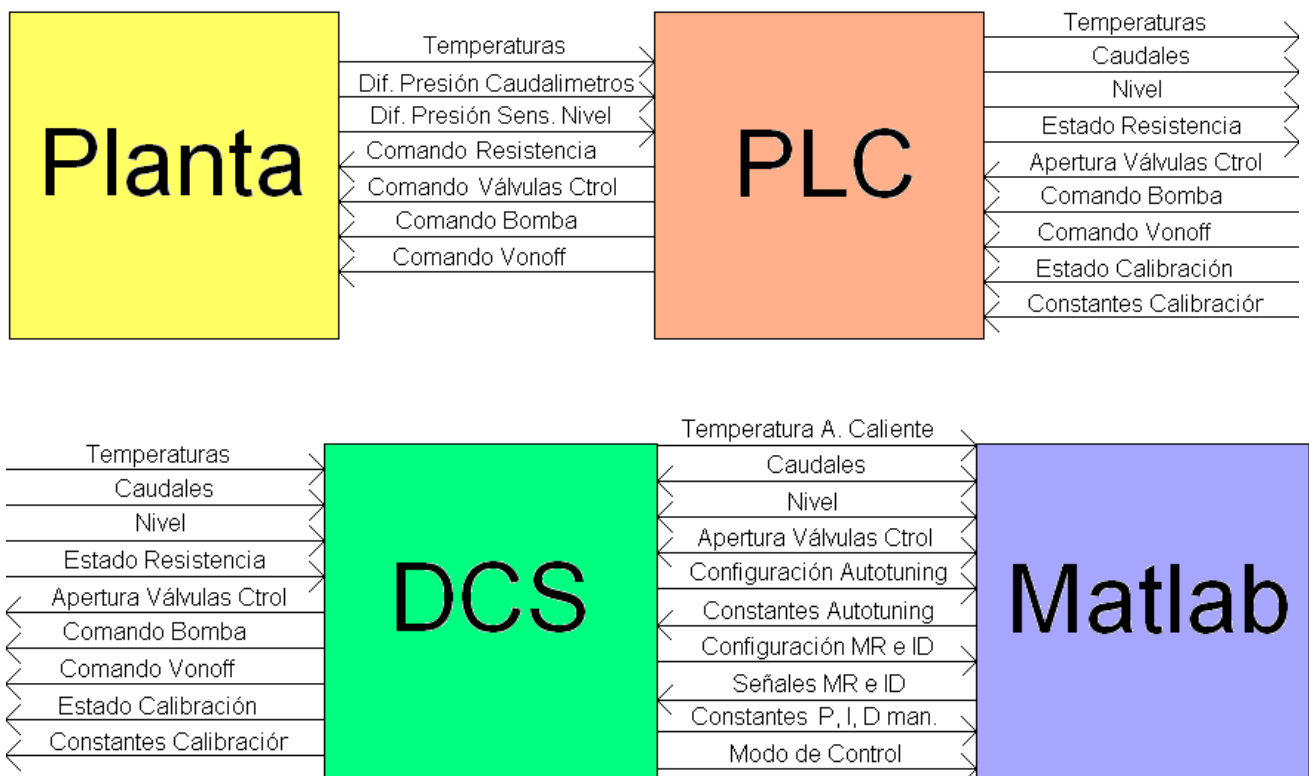
Detalle del extremo de la entrada general de agua

1.3 Comunicación de datos Planta-Matlab

El sistema consta de diferentes partes:



Entrando en mayor detalle, en cada interfaz se intercambia la información indicada en el siguiente esquema:



Planta: es la planta piloto diseñada y construida para simular un proceso industrial.

PLC KR 51: su principal función es adquirir la información de los instrumentos de medida y controlar los actuadores por medio de I/O. Como los instrumentos y actuadores no son industriales el PLC KR51 también debe adaptar las señales de estos

A grandes rasgos las funciones del PLC son:

- comunicar la información adquirida desde los sensores al DCS 800xA
- enviar comandos, que llegan desde el sistema, a los actuadores de la Planta.

Sistema 800xA: entre las funciones del sistema están:

- i. Actuar como SCADA
- ii. Despliegue de gráficos que muestren la planta.
- iii. Almacenamiento y despliegue de históricos en forma de tendencias.
- iv. Despliegue de datos en tiempo real
- v. Ser el nexo entre el PLC y Matlab.
- vi. Ser el soporte de la HMI diseñada

Matlab: Es la plataforma en la cual se hacen las aplicaciones de control avanzado ya que cuenta con una variedad de herramientas muy poderosas a la hora de crear, simular y testear sistemas tanto lineales como no lineales. Para este proyecto en particular se decidió trabajar con Simulink que es un entorno gráfico de programación propio de Matlab que tiene una gran cantidad de ventajas respecto a la programación clásica de Matlab (programación en archivos con extensión .m).

2 Guía de operación de la Planta

2.1 Pasos a seguir para iniciar la planta

- 1) Conectar la manguera de alimentación (la que tiene rosca) de agua a la red de OSE y colocar las dos mangueras (una negra y la otra transparente) de desagüe de manera tal que el agua desagote.
- 2) Abrir completamente la válvula Vo4.
- 3) Abra el tablero de la planta.
- 4) Conectar el alargue negro al tomacorrientes, cerciorarse de que estén todas las llaves de protección apagadas.
- 5) Realizar la conexión del PC con el PLC (cable de comunicación serial negro)
- 6) Conecte el PC a la corriente, mediante el alargue conectándolo al tomacorriente Reserva
- 7) Levantar la llave diferencial
- 8) Levantar la llave PLC
- 9) Levantar la llave Calefón (si se va a usar agua caliente)
- 10) Levantar la llave Reserva
- 11) No levante la llave “Válvulas y Sensores” hasta que se le indique (en este momento no se sabe que datos tienen cargados el PLC, por lo que se podrían dañar los actuadores)
- 12) Inicie el PC y loggése con nombre de usuario: ITC y password: itc
- 13) Levante el workplace (por detalles ver 2.2.1)
- 14) Espere que Matlab se inicialice.
- 15) Compruebe que los datos de calibración estén cargados y en caso contrario deberá realizarlo.
- 16) Abrir la canilla de alimentación.
- 17) Abra la válvula de entrada general de la planta.
- 18) Levante la llave “Válvulas y Sensores”
- 19) Realice la calibración de la planta de ser necesario.
- 20) Ahora se está en condiciones de empezar a trabajar con el sistema de control.

Alarmas y bloqueos configurados:

Para evitar accidentes indeseables se configuraron las siguientes alarmas en el sistema:

- Alto nivel: Cierra las válvulas Vc1 y Vc2, hasta que el nivel baje de 100%.
- Bajo nivel: Desconecta la alimentación de la resistencia del tanque.
- Válvula ON-OFF en posición desagote, impide el encendido de la bomba.
- Válvula Vc3 con apertura menor al 20%, impide el encendido de la bomba.

2.2 Guía rápida de utilización del Sistema

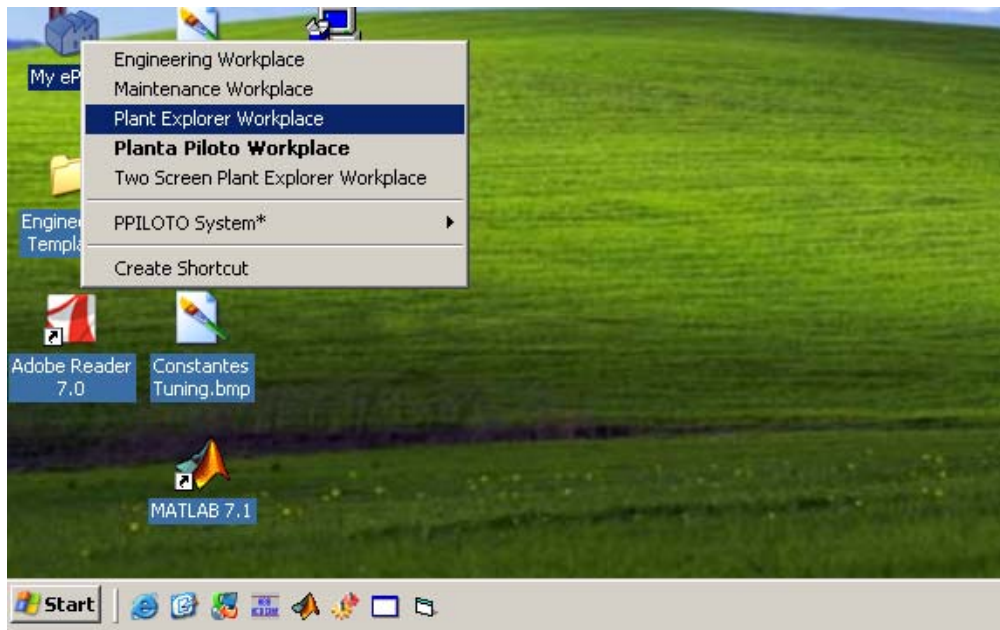
La PC tiene instalado el sistema operativo XP SP2, con el cual todos los usuarios deberían estar familiarizados ya que es uno de los sistemas operativos más usados por el común de las personas. Al prender la PC aparece la opción de loggearse con tres usuarios:

- 1.-engineer
 - a.-Contraseña: engineer
 - b.-Descripcion: Este usuario fue creado para correr los servicios del sistema DCS ABB y se recomienda fuertemente no hacer ningún tipo de cambios en el mismo.
- 2.-EngineerPP
 - a.-Contraseña: engineerpp
 - b.-Descripcion: Con este usuario se debe hacer la ingeniería del sistema, armar pantallas, hacer cambios en los objetos, configura Matlab, etc.
- 3.-ITC
 - a.-Contraseña: itc
 - b.-Con este usuario los estudiantes se loggearán para trabajar con el sistema de control, estando limitado en sus funciones de sólo operación.

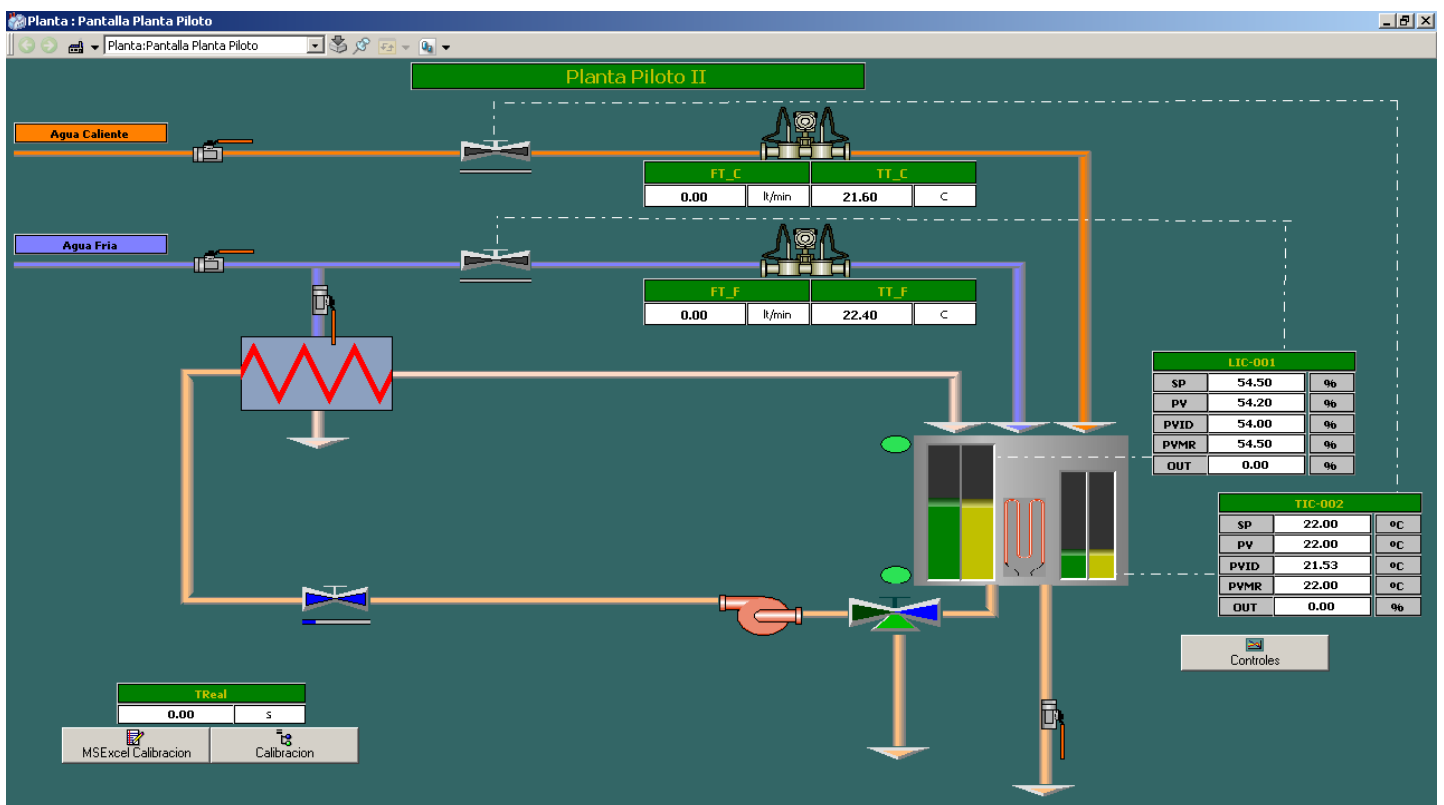
De aquí en más se va describir cómo usar el sistema para el usuario de ingeniería, ya que este requiere de más pasos que el usuario ITC. Para el usuario ITC el procedimiento es muy parecido e intuitivo.

2.2.1 Como ingresar al sistema de control

- ✓ Una vez prendida la PC el sistema comienza, solo, a cargar los servicios que necesita para funcionar, esto es así porque se seleccionó la opción de que comenzara a correr durante el inicio de la PC.
- ✓ El usuario se puede loggear como usuario ITC o EngineerPP. En el caso que se loggee como ITC, la pantalla principal se abrirá ocupando todo el escritorio y posibilitando el uso del sistema de control en modo operación, también el programa de Matlab comenzará a correr en forma automática. Si se selecciona el usuario EngineerPP, se debe presionar botón derecho sobre el icono con forma de fábrica y seleccionar la opción de Plant Explorer.



Luego de haber hecho lo anterior se abrirá la siguiente ventana:



La figura anterior es la pantalla principal, en la misma hay íconos animados, por ejemplo hay niveles en el tanque que cambian su altura de acuerdo a valores de proceso. Los íconos además de ser animados también son un acceso directo a los Faceplates de las distintas unidades de control, por ejemplo: válvula ON-OFF de tres vías, Bomba, controles de nivel y temperatura, válvula de realimentación.

En la parte inferior izquierda hay dos botones. Uno abre una planilla de Excel (MSExcel Calibración) que sirve para cargar automáticamente las constantes obtenidas de las calibraciones. El otro botón (Calibración) abre un faceplate especialmente creado para hacer dichas calibraciones.

2.2.2 Uso de los Controladores

A continuación se explica el funcionamiento de los controladores y sus faceplates, además se aprovechará la ocasión para explicar algún detalle de utilización del DCS.

2.2.2.1 Objeto Controlador (LIC o TIC)

Como el controlador de temperatura es prácticamente igual al de nivel (sólo se diferencian en que la ID y el MR del TIC son de primer orden mientras que en el LIC son de segundo orden) sólo se discutirá éste último.

2.2.2.2 Faceplates de los Controladores

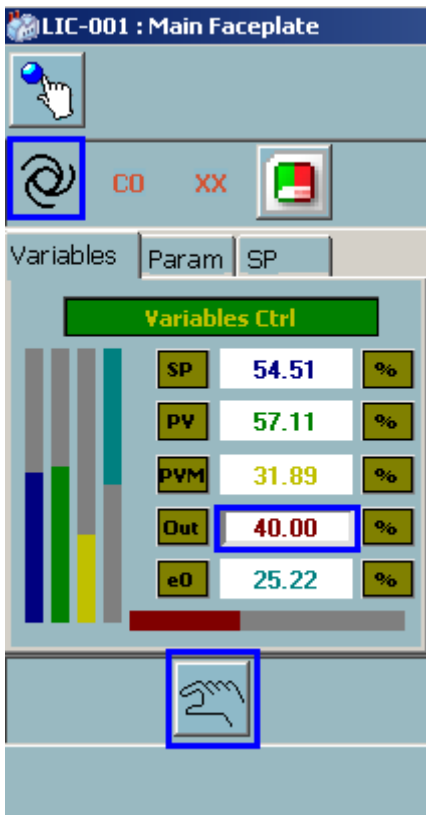
Para que la interfaz del usuario con el controlador diseñado sea amigable se creó el siguiente Faceplate:



Los controladores de nivel (LIC-001) y temperatura (TIC-002) tienen tres modos de funcionamiento: Manual, PID Automático y Modelo de Referencia, además de la posibilidad de habilitar la Identificación de la planta. Estos modos de funcionamiento son explicados, en detalle en el punto a continuación.

2.2.2.3 Modos de funcionamiento del Controlador

2.2.2.3.1 Modo Manual



En el modo manual se maneja directamente la salida del control, o sea que el usuario elije la apertura de la válvula a su juicio.

En parte superior izquierda de la figura de al lado se recuadró, en azul, un remolino. Este recuadro indica en qué modo se encuentra el control, en este caso es en modo automático.

En la parte inferior de la figura se recuadró una mano gris. Este botón habilita la opción de modo manual, el mismo aparece de ese color debido a que la opción de manual no está elegida y por lo tanto se puede elegir. Si el control estuviera en modo manual este botón tendría la mano dibujada en gris indicando que se está en modo manual y por lo tanto no se puede volver a seleccionar.

A la derecha hay cinco valores numéricos:

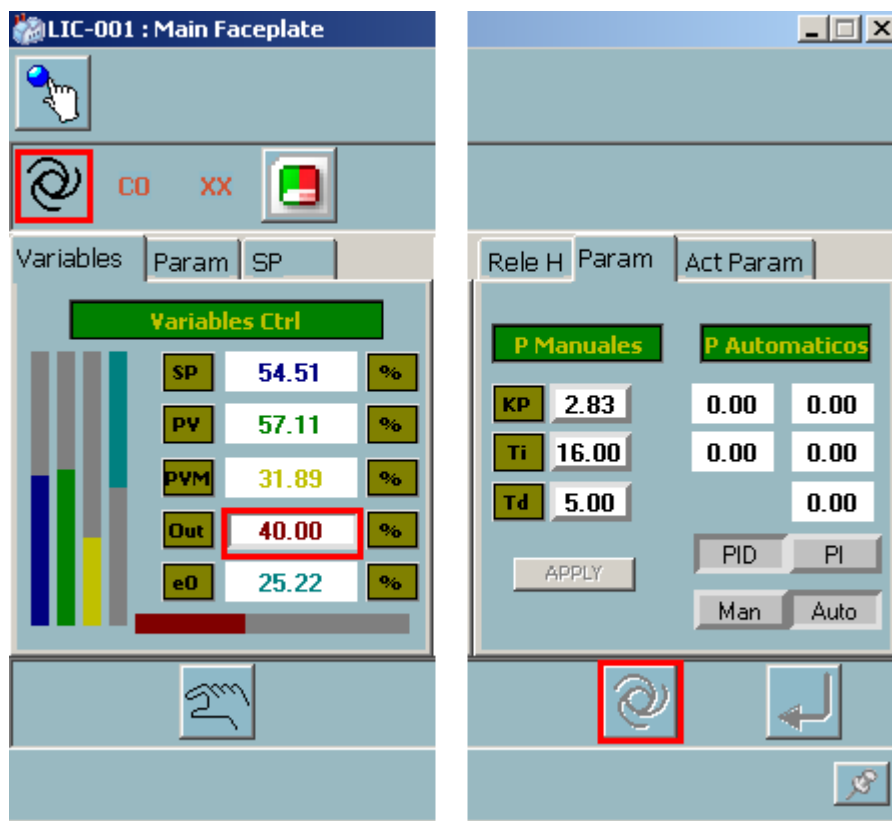
- 1- SP – Setpoint, este el valor de referencia que se desea como valor de proceso
- 2- PV – Valor de proceso
- 3- Out – Es el valor de salida del PID en forma automática, se debe notar que en modo manual la salida Out sigue a MOut (Bumpless)
- 4- MOut – Salida en modo manual, se debe notar que en modo automático no es posible cambiar manualmente este campo
- 5- E0 – Es el error a la entrada utilizado por el MR.

2.2.2.3.2 Modo PID Automático

El modo automático maneja la salida del control a través del algoritmo de PID programado, o sea que el usuario maneja el SP de proceso que desea y no se utiliza la salida de control.

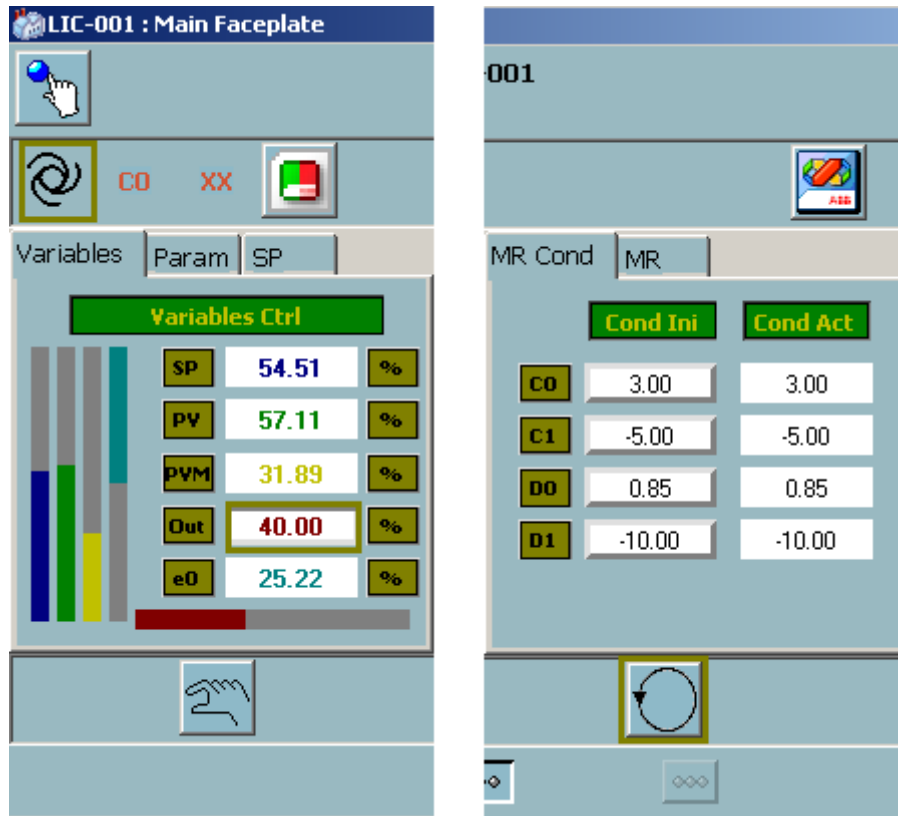
En la parte superior izquierda de la siguiente figura está encerrado, en un recuadro rojo, un dibujo de un remolino. El mismo indica que el control está en modo automático. En la parte de inferior izquierda de la figura se encuentra un remolino sin color, este botón habilita la opción de modo automático. El remolino aparece en ese color debido a que la opción de automático está elegida y por lo tanto no se puede volver a elegir. Si el control estuviera en otro modo, como por ejemplo manual, este botón tendría el remolino dibujado en color negro indicando que el control no está en modo automático y que se puede seleccionar.

En el centro de la figura aparece un número recuadrado en rojo, este número muestra el valor de la salida de control en cada instante del algoritmo.



2.2.2.3.3 Modo Control por MR

El modo MR maneja directamente la salida del control, o sea que el Modelo de Referencia maneja por su cuenta la apertura de la válvula.



En la parte superior izquierda de la figura de arriba, aparece un remolino recuadrado. Este recuadro indica en qué modo se encuentra el controlador, en este caso es en automático. En la parte de inferior izquierda de la figura se encuentra un círculo en negro, este es el botón que habilita la opción de modo Modelo de Referencia, la misma aparece en ese color ya que la opción de Modelo de referencia no se encuentra elegida y por lo tanto se puede elegir.

Cuando funciona el MR y se están adaptando los parámetros las indicaciones (en la parte superior del faceplate) 'CO' y 'XX' aparecen en verde, cuando 'CO' no se adapta aparece en rojo y cuando ningún parámetro se actualiza 'XX' también lo hace. En el centro de la figura se recuadró el valor de la salida del control.

Variables	Param	SP
Ctrl MR-ID		
g		0.0060
eMin		0.5000
g1		0.5000
Ctrl MR		
CO_Min		0.5000
Lambda		4.0000

En la figura de la izquierda se muestra donde se configuran los parámetros 'g', 'error mínimo' y 'g1'. Estos parámetros son compartidos por los algoritmos de Modelo de Referencia e Identificación.

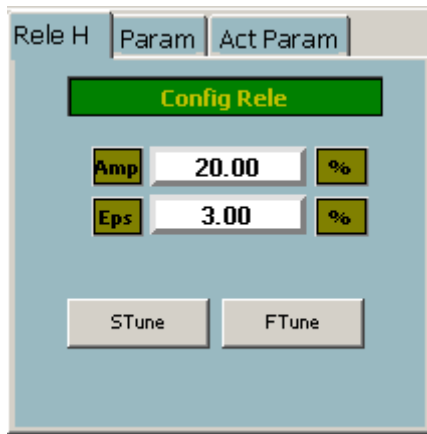
Debajo aparecen los parámetros 'C0 mínimo' y 'lambda', los cuales son de uso exclusivo del Modelo de Referencia. La explicación de estos parámetros se encuentra en el Capítulo 6 de Control Avanzado.

En las figuras siguientes se muestran las solapadas en donde se deben configurar las condiciones iniciales de los parámetros de control adaptativo, y el modelo de referencia a seguir.

MR Cond	MR
Modelo Referencia	
$\frac{aS + b}{S^2 + cS + d}$	
a	1.00
b	0.25
c	1.00
d	0.25

MR Cond	MR
Cond Ini	
C0	3.00
C1	-5.00
D0	0.85
D1	-10.00

2.2.2.4 Autotuning PID

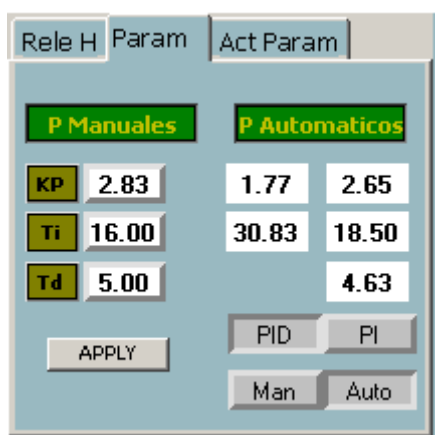


La sintonía del PID, mediante el algoritmo de relé implementado en este proyecto, se habilita y configura en la parte del faceplate del controlador mostrada en la figura a la izquierda.

La sintonía se habilita sin importar el modo de operación en el cual se encuentra el control. Una vez terminada la misma, el control vuelve al modo en el cual el controlador se encontraba antes de comenzar la sintonía.

Los parámetros que juegan en la sintonía son:

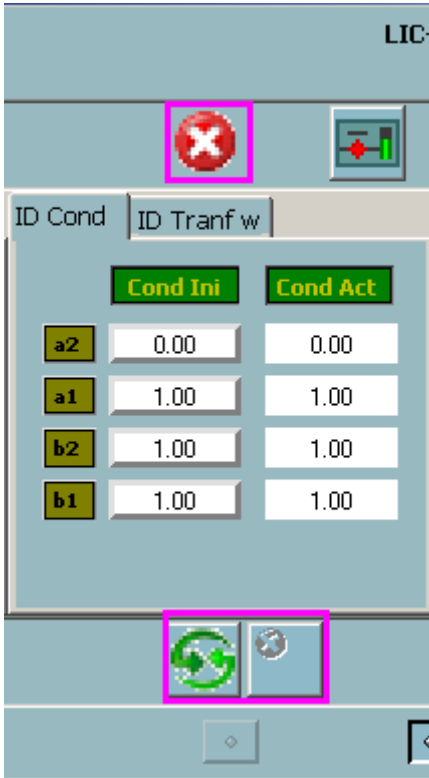
- 'Amp': amplitud pico a pico de la onda cuadrada del algoritmo.
- 'Eps': Épsilon pico a pico del algoritmo de relé
- 'STune': Se debe presionar este botón por más de un segundo para que la sintonía comience. Cuando la misma comienza aparece una indicación con forma de diagrama de Nyquist en el faceplate del Tuning.
- 'FTune': Se debe presionar este botón por más de un segundo para que la sintonía termine.



En la figura a la izquierda se muestra la parte del faceplate en donde se configuran las constantes del PID. Los botones 'PID' y 'PI' seleccionan el tipo de control a usar (solo se puede tener seleccionado uno pues el otro se deselecciona automáticamente). Los botones 'Man' y 'Auto' seleccionan de donde se toman las constantes para el algoritmo. Si se elige 'Auto' las constantes se toman del resultado del algoritmo de autosintonía de relé.

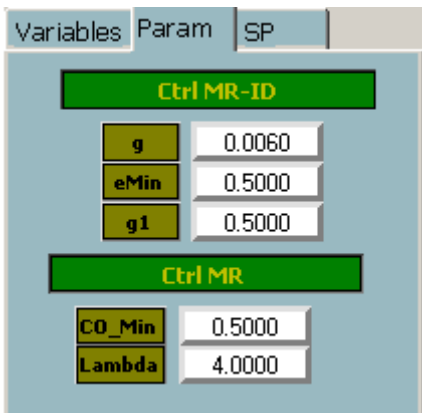
Finalmente para que la selección elegida se aplique se debe presionar el botón de 'Apply' por más de un segundo y se debe corroborar que las constantes elegidas aparezcan en la solapa 'Act Param'. El botón de 'Apply' se habilita solo si la opción elegida es correcta, si por ejemplo se selecciona control automático PID, sin antes hacer al menos una sintonía completa, el mismo permanece deshabilitado.

2.2.2.5 Identificación



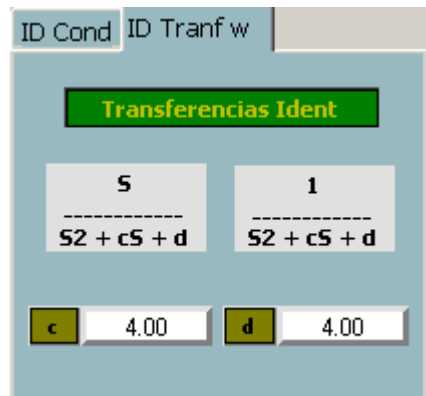
En la parte inferior de la figura de la izquierda se muestran los botones que habilitan o deshabilitan la identificación de la planta, el círculo rojo con una 'x' blanca en el interior deshabilita la identificación de la planta, mientras que las flechas verdes la habilitan. En el caso de que no se habilite la identificación, los parámetros de la misma no se actualizan.

Las condiciones iniciales del algoritmo de Identificación se encuentran en el centro de la figura a la izquierda, y a la derecha las condiciones actuales de los parámetros de la identificación.



Los parámetros 'g', 'error mínimo' y 'g1' se encuentran debajo del cartel Ctrl MR-ID. Estos son compartidos en el control por MR y en la identificación.

Las transferencias de la identificación (los identificadores, en este caso en particular la ID es de segundo orden) están en la figura de la derecha.



2.2.2.6 Botones Auxiliares

En la parte superior de los faceplates de los controladores (en las indicaciones) aparecen tres botones. El botón de la izquierda despliega las tendencias de las variables de control, el botón del centro despliega las tendencias relacionadas con el algoritmo de identificación y el botón de la derecha despliega las tendencias relacionadas del modelo de referencia.

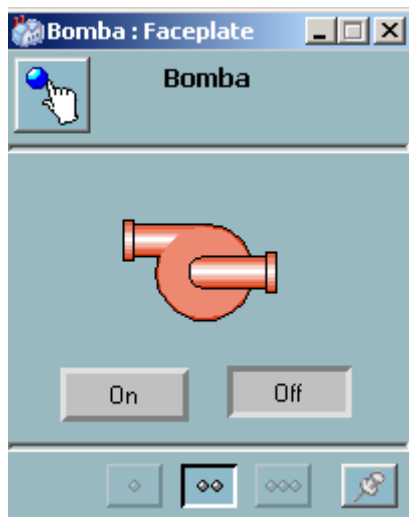


2.2.3 Otros objetos implementados

A continuación se explica el funcionamiento de los objetos Bomba, Resistencia y Válvula de Realimentación y sus respectivos Faceplates.

2.2.3.1 Bomba

El comando de la bomba es ON-OFF, o sea que la misma puede estar prendida o apagada. Para la misma se implementó un bloqueo de seguridad para que no prenda cuando por el lazo de realimentación no circula suficiente caudal, con esto se pretende que la misma no se sobre-exija .



La bomba se puede prender y apagar mediante los botones 'On' y 'Off' respectivamente. En la figura de la izquierda se muestra el faceplate de éste objeto. Cuando el botón 'Off' se encuentra presionado el dibujo de la bomba es de color rojo, esto indica la orden de parada de la misma y cuando el botón 'On' está presionado, el dibujo de la bomba es de color verde indicando que se dio la orden de marcha.

Cuando se produce algún bloqueo, automáticamente se "presiona" el botón 'Off' y se deshabilita el botón 'On' (la palabra On queda sin color). Además en la pantalla principal aparece un cartel que avisa que la bomba está bloqueada.

2.2.3.2 Resistencia

El comando de la resistencia es ON-OFF. A diferencia de la bomba ésta es controlada por el PLC y el usuario sólo tiene el control de configurar el 'duty cycle' del encendido y el período de trabajo.

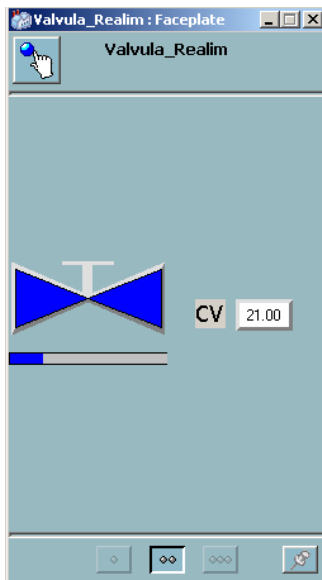
Como medida de seguridad el control de la resistencia, configurado en el PLC, comienza a funcionar cuando el nivel cubre totalmente la resistencia (nivel LT_0).



En la figura de la izquierda se muestra el faceplate de éste objeto. Cuando el botón 'On' se encuentra presionado, la resistencia se dibuja de color rojo indicando que está prendida, cuando el botón 'Off' está presionado, el dibujo de la misma es de color verde indicando que está apagada.

Los botones de este faceplate están solamente para mostrar el estado de la resistencia y no pueden ser operados.

2.2.3.3 Válvula de realimentación



Esta válvula es operada directamente por el usuario y se controla mediante el valor en el campo 'CV' que aparece en el faceplate de la izquierda. Este campo acepta valores entre 0 y 100, donde 0 significa válvula totalmente cerrada y 100 significa válvula totalmente abierta.

2.2.3.4 Indicaciones

Para presentar datos en pantalla de forma amigable se diseñaron objetos de indicación que presentan el valor de variables tales como caudales, temperaturas, niveles, setpoints, etc. Los mismos son muy fácilmente entendibles y no tiene sentido una explicación más detallada.

2.3 Exportación de Históricos a planillas de Excel

Abrir la planilla Excel: 'Trend_PPiloto.xls' ubicada en el escritorio:

2			Trends Planta Piloto					
3								
4				Start		End		
5			Month	8		8		
6			Day	16		16		
7			Year	2008		2008		
8			Hour	0		0		
9			Minute	5		12		
10			Seconds	0		1		
11				08/16/08 00:05:00		08/16/08 00:12:01	3200	
12	LIC-001							
13	SP		PV		PVM		OUT	
14	08-16-2008 00:05:00		70 08-16-2008 00:05:00		0 08-16-2008 00:05:00		69,68260193 08-16-2008 00:05:00	0

Llenar la planilla con los datos de fecha, hora de inicio y finalización del experimento, en los campos correspondientes. Luego presionar la tecla 'F9', y automáticamente se completan los datos en las columnas SP (SetPoint), PV (Process Value), PVM (Process Value del modelo, no se utiliza), OUT (salida del controlador, entrada a la planta).

Se debe generar una planilla para cada experimento realizado. Para no obtener datos innecesarios disminuir el tamaño de la misma. Se recomienda diferenciar cada experimento, por ej.: guardar los datos con un nombre diferente de archivo (fácil de distinguir) por cada experimento.

3 Prácticas

3.1 Práctica 0: Uso de Matlab y de la Planta Piloto

3.1.1 Objetivo

Creación de programas en Matlab para procesar los datos que se obtendrán en el laboratorio. Familiarización con la operación de la Planta

3.1.2 Ejercicios

Se recomienda la lectura del instructivo “Usando MATLAB para resolver problemas de Control” (HAKAS M. y FONSECA A, IIE, Publicado por el CEI en 1999), el mismo contiene explicaciones y ejemplos de las funciones de control de MATLAB más usadas.

Ejercicio 1: Análisis de respuesta al escalón (igual que la Práctica 0 de ITC, pero con una planta de primer orden)

- Considere una planta cuya función de transferencia es:

$$H(s) = \frac{2}{1.5s + 1}$$

- Obtenga su respuesta al escalón unitario
- Agréguele ruido blanco (asuma distribución normal con $\mu = 0$ y $\sigma^2 = 0.01$).
- Elabore un programa que a partir de la respuesta al escalón ruidosa identifique un modelo de 1er orden, extraiga los atributos de la respuesta al escalón (tiempo de respuesta, ganancia en régimen, etc.), y grafique la respuesta ruidosa junto con la del modelo identificado.

Ejercicio 2: Manejo de la Planta Piloto (Modo manual)

Estando la planta en modo manual ejecute las siguientes operaciones:

- Cierre las 3 válvulas de control.
- Cambie el valor de apertura de cada una de ellas (una a la vez), para poder apreciar su comportamiento e implementación.
- Abra la válvula de agua fría al 45% y establezca el nivel en un valor cercano al 50% del nivel del tanque, variando la apertura de la válvula manual de salida Vo4.
- Cambie de posición la válvula ON-OFF y mida con un cronómetro su tiempo de respuesta. Realice la operación en ambos sentidos, ¿el tiempo insumido es el mismo?
- Verifique si durante la realización de las operaciones anteriores, se dispara alguna de las alarmas implementadas e identifíquela.

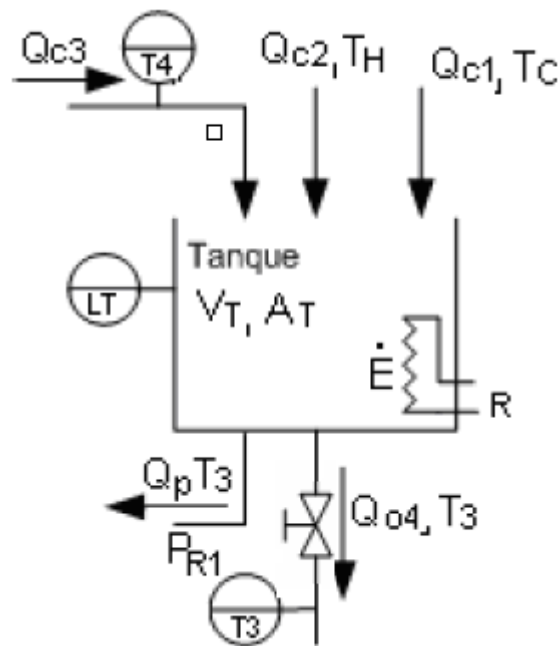
3.2 Practica 1: Modelado e identificación de parámetros

3.2.1 Objetivo

Construir un modelo lineal del sistema que define el nivel dentro del tanque y determinar sus parámetros por experimentación.

3.2.2 Modelado de la planta

Para el desarrollo se considera la siguiente hipótesis de trabajo: el flujo de agua a lo largo de las cañerías se considera de régimen turbulento.



¹ Esquema del Tanque

Planteando el balance de masa en el tanque:

$$\begin{aligned} Q_{c1} + Q_{c2} + Q_{c3} - Q_{o4} - Q_p &= \dot{V}_T \\ \left. \begin{aligned} Q_{c3} &= Q_p \\ \dot{V}_T &= \dot{L}_T A_T \end{aligned} \right\} \Rightarrow Q_{c1} + Q_{c2} - Q_{o4} &= \dot{L}_T A_T \\ Q_{o4} &= K_v \sqrt{L_T} \end{aligned}$$

L_T : Nivel de agua dentro del tanque en m .

A_T : Área transversal del tanque en m^2

V_T : Volumen de agua en el tanque en m^3

K_v : Constante de la válvula

¹ (Berenfus, Casas, & Rezk, 2006)

- Q_{c1} : Caudal de agua caliente
- Q_{c2} : Caudal de agua fría
- Q_{c3} : Caudal de agua de recirculación que entra al tanque
- Q_{o4} : Caudal de agua de salida del tanque
- Q_p : Caudal de agua de recirculación que sale del tanque


Calcule la función de transferencia que relaciona el caudal $Q_{c2}(s)$ con el nivel L_T , considerando que el caudal de agua caliente es constante.

3.2.3 Ensayos a realizar

Los valores que interesan obtenerse en estos ensayos son la entrada y la salida de la planta, estos son PV (entrada) y OUT (salida).


1. Utilizando una de las funcionalidades implementadas en la Planta Piloto (autosintonía por el método de relé de Aström) se obtendrá la respuesta de la misma a varias entradas escalón. Los datos obtenidos se relevarán en una planilla Excel a utilizar por los estudiantes. (Ver Modo Autosintonía PID).
2. Repita el experimento, trabajando con un nivel estable diferente.

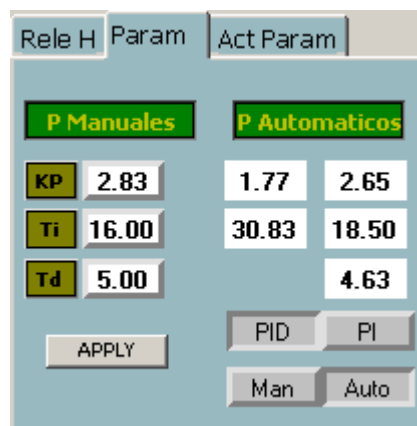
Pasos a seguir para realizar el ensayo:

- 1- El control de la planta debe estar en manual 
- 2- Las válvulas Vo3, Vc3 y Vc1 estarán cerradas. La bomba estará apagada.
- 3- La válvula ON-OFF estará en su posición normal (recirculación).
- 4- Abra la válvula Vo4 al 50%.
- 5- Abra la válvula Vc2 al 100% (escriba el valor 100 en Out) hasta que se supere el nivel deseado (setpoint)
- 6- Maneje la válvula Vc2 hasta estabilizar el nivel alrededor del 40%.
- 7- Una vez estabilizada la planta, presione el botón STune hasta que aparezca el

símbolo  en el faceplate.



- 8- Con el botón  se desplegarán la gráfica de las variables del proceso.
- 9- Cuando el símbolo desaparezca, se habrá terminado con la auto sintonía.
- 10- Exporte los datos obtenidos a la planilla Excel.
- 11- Anote los valores obtenidos por la auto sintonía (Parámetros Automáticos).



3.2.4 Identificación

En este momento se contará con la respuesta a escalones con los cuales se identificarán los parámetros de la función de transferencia lineal calculada.

Los datos relevados en las planillas Excel se utilizarán para obtener el modelo de la Planta, utilizando los programas realizados en la Práctica 0.

El método a utilizar para realizar la identificación, será el de respuesta escalón.

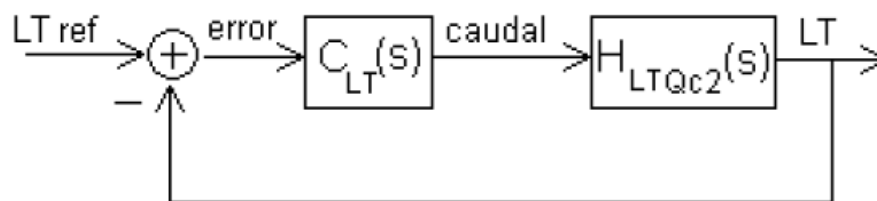
3.3 Práctica 2: Simulación y diseño de controladores asistido por computadora

3.3.1 Objetivo

Diseñar controladores para que el sistema realimentado cumpla con requerimientos predefinidos y simularlos para corroborar su desempeño

Los controladores a diseñar serán PI y PID. Las especificaciones de los controladores serán definidas por los docentes.

Diagrama del controlador implementado:



² Lazo de control del nivel de agua fría.

$$C_{LT}(s) = k_c \left(1 + \frac{1}{T_i s} + T_d s \right) \text{ (Controlador PID)}$$

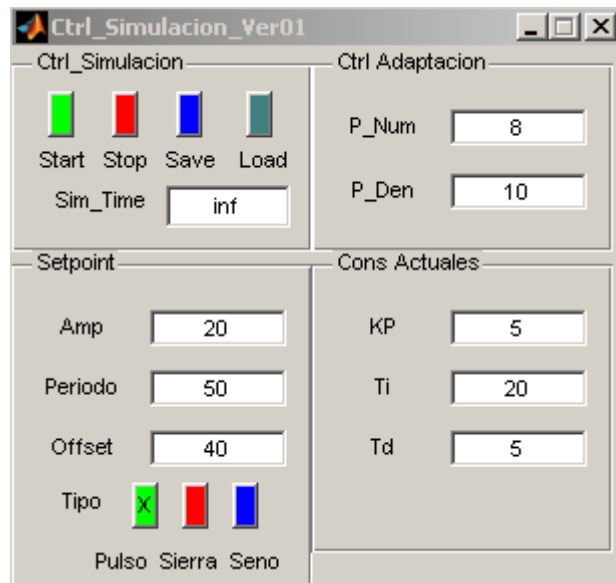
3.3.2 Simulación de los controladores

Se provee un programa en Matlab (Versión 7, SP3) con el cual se realizarán las simulaciones. Posee un generador de ondas, los parámetros del PID y un bloque donde se inserta el modelo de la planta identificado.

Para ejecutar el programa se debe iniciar Matlab, ir al directorio donde se han guardado los archivos y ejecutar Ctrl_Simulacion_Ver01.m.

Al ejecutarlo se abrirá la siguiente pantalla:

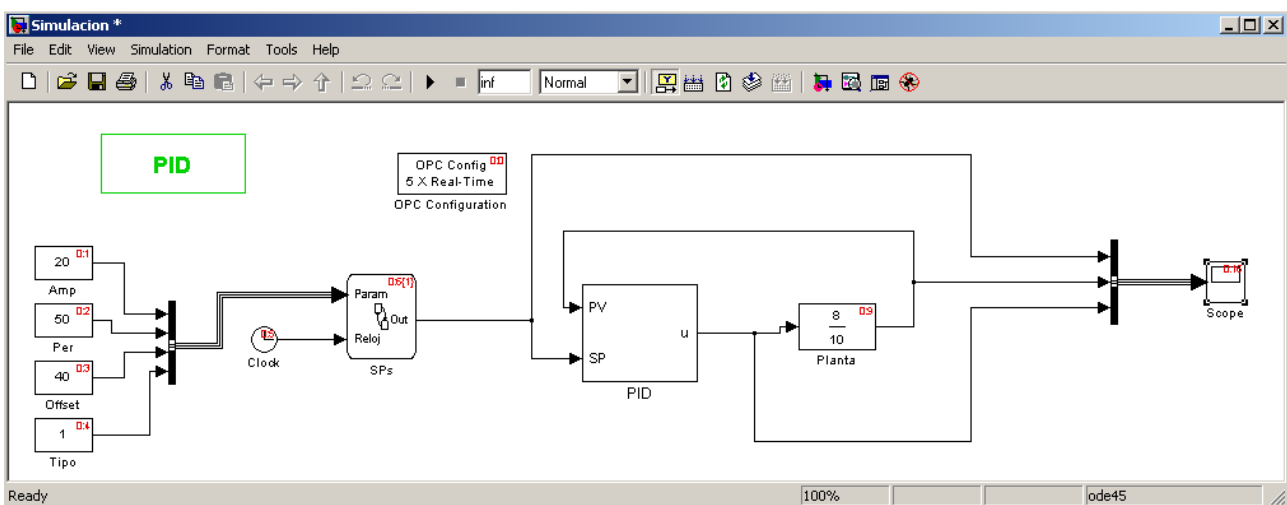
² (Berenfus, Casas, & Rezk, 2006)



Esta pantalla está dividida en varias secciones:

- “Ctrl_Simulación”: Es el control del programa, con ‘Load’ se cargan los datos, luego se pulsa ‘Save’ para guardarlos, a continuación se presiona ‘Stop’ y luego ‘Start’ para empezar a correr la simulación.
- “Ctrl Adaptación”: Se ingresan los datos de la planta, numerador (‘P_Num’) y denominador (‘P_Den’), con el formato [a b] = as+b.
- “Cons Actuales”: se ingresan los parámetros del controlador a simular (KP, Ti y Td), para diferenciar un controlador PID de uno PI, se debe de poner un cero en el parámetro Td. Si KP y/o Ti ≤ 0 , el programa da una alarma. Si Td < 0 , el programa da una alarma.
- “Setpoint”: es el generador de ondas de entrada a la planta. Amp, define la amplitud de la señal, Periodo el tiempo en segundos del período de la señal y Offset, la diferencia con el cero del valor mínimo de la onda. En Tipo, se pulsa el cuadrado (en el cual aparece una X para confirmar la selección) para seleccionar un pulso, un diente de sierra o una onda sinusoidal.

Además de la pantalla anterior, se despliega la siguiente ventana:

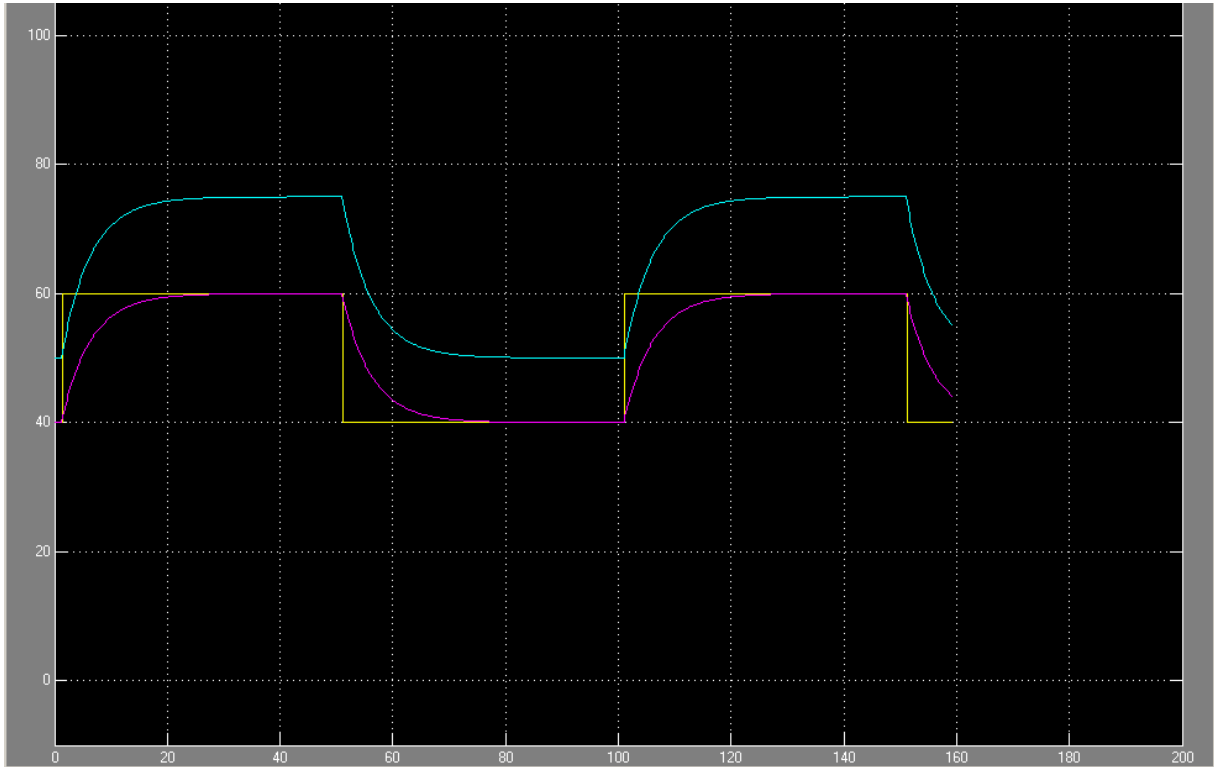


Esquema del PID implementado en Matlab

En esta imagen se ve el diagrama completo del controlador y haciendo doble-click sobre el



símbolo: se despliega una ventana que contiene la gráfica con las variables de control del sistema:



Gráfica con las variables del sistema

- En amarillo el SetPoint configurado.
- En violeta el Process Value (salida de la planta).
- En celeste la salida del controlador (entrada a la planta).

Con el programa el estudiante podrá visualizar el comportamiento de sus controladores y confirmar si se cumplen las especificaciones de los mismos.

3.4 Práctica 3: Implementación y evaluación de controladores.

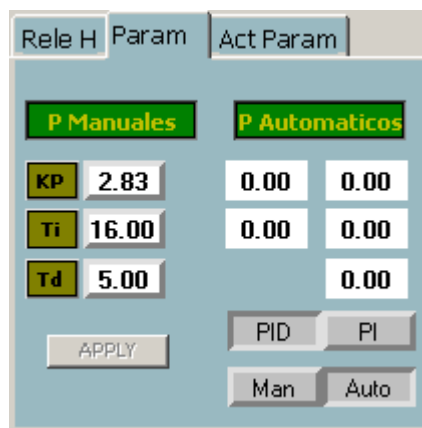
3.4.1 Objetivo

Implementar los controladores diseñados para la Planta y realizar una comparación de los resultados obtenidos, en la experiencia, con los simulados.

3.4.2 Implementación de Controladores


La implementación de los controladores estará realizada en Matlab. Los estudiantes deberán configurar los diferentes controladores con los parámetros calculados previamente.

Para implementar los controladores, los estudiantes cargarán los parámetros obtenidos teóricamente en el faceplate correspondiente.



Se ingresarán los parámetros de los controladores diseñados en 'P Manuales', se elegirá la opción PI o PID según corresponda, se seleccionará la opción manual y, para que los datos sean cargados en el sistema, se presionará por más de 1 segundo el botón 'APPLY'. Para comprobar que la carga fue satisfactoria, se verificará que los valores se desplieguen en la solapa 'Act Param'.

3.4.3 Ensayo

- Se llevará la planta al punto de equilibrio en el cual se trabajó en la Práctica 1, para luego presionar el botón  que dará comienzo al control, en modo PID o PI (según corresponda), de la variable de proceso de planta con el controlador configurado.
- Se verificará el funcionamiento de los controladores observando la respuesta a escalones (subida y bajada) generados manualmente, variando el SP Manual.

Variables Param SP

Setpoint

Manual Pulso Sierra Seno

SP Manual	54.50	%
Amplitud	20.00	%
Periodo	360.00	s
Offset	40.00	%

- Realice una evaluación aproximada del desempeño de los controladores y en caso de no satisfacer las especificaciones, verifique los datos ingresados al sistema y que el punto de operación sea el utilizado en la Práctica 1.
- Si aun no se logró cumplir con las especificaciones, varíe las constantes de los parámetros utilizando algún criterio.
- Una vez alcanzado los requerimientos, se exportarán todos los datos relevados a la planilla Excel, para comprobar que los controladores diseñados cumplieron las especificaciones previstas.
- Deberá guardar los valores de las constantes con las cuales se logró cumplir las especificaciones pedidas.

Bibliografía

Hakas M. y Fonseca A., "*Usando MatLab para resolver problemas de Control*", CEI, 1999.

Canetti R., Fonseca A., Hakas M., Rromán J., et al., "*Prácticas de Laboratorio de Introducción a la Teoría de Control*", CEI, 2005.

Berenfus Viviana, Casas Natalia y Rezk Gabriela, "*Planta Piloto para Control de Procesos*", Facultad de Ingeniería del Uruguay, 2006.

F. Ferreira, S. Rodríguez y P. Rodríguez, "*Planta Piloto para Control de Procesos v2*", Facultad de Ingeniería del Uruguay, 2008.