



Escuela Técnica Superior de Ingenieros
Universidad de Sevilla



Control Borroso Industrial: Uso de la IEC 1131-7 para el control de plantas industriales

Autor: Rafael Pajarón Pérez
Tutor: Juan Manuel Escaño González
Cotutor: Luis Fernando Castaño Castaño

Proyecto Fin de Carrera
Ingeniería de Telecomunicación

Sevilla, 28 de septiembre de 2012

Este Proyecto se realizó en el



INGENIERÍA DE
SISTEMAS Y AUTOMÁTICA

Departamento de Ingeniería de Sistemas y Automática
<http://disa.us.es/disa/>

Grupo de investigación: Automática y Robótica Industrial

DEDICATORIA

A mis padres y mi novia Beatriz...

“La potencia intelectual de un hombre se mide por la dosis de humor que es capaz de utilizar.”

Friedrich Nietzsche

Agradecimientos

Me gustaría agradecer a Juan Manuel Escaño, mi tutor y amigo, por la oportunidad que me ha brindado para realizar este proyecto y aprender de él así como agradecer su infinita paciencia y ayuda. A mis padres y al resto de mi familia, mi hermana y mi abuela, mis tios, primos,... por vuestra confianza en mí. A mi novia, fuente inagotable de apoyo y cariño que me hicieron fuerte en los momentos de decline y cansancio. A todos mis compañeros y amigos de la universidad, porque sin todos vosotros, vuestros ánimos, vuestros consejos y vuestra ayuda no habría terminado nunca. A todos mis amigos de siempre, en especial a Pablo por estar siempre ahí. Y en definitiva agradecer a todas las personas que de una u otra manera me ayudaron en el desarrollo de mi Proyecto Fin de Carrera.

Prefacio

El presente proyecto se centra en la implementación de estrategias de Control Borroso en autómatas programables industriales, proporcionando a los lectores una guía práctica para iniciarse en el mundo de las plantas reales. Se ha realizado una conexión vía OPC para desarrollar una plataforma de interconexión entre el software Unity de Schneider Electric y Matlab, lo que proporcionará una herramienta muy útil para el desarrollo y sintonización de controladores antes de implantarlos en plantas reales.

Podemos dividir el proyecto en varias partes. Una primera parte en la que se exponen los objetivos marcados y se realiza una introducción a la teoría de la lógica borrosa, conceptos básicos sobre los que gira este proyecto. Posteriormente se expone el sistema que se pretende controlar para más tarde diseñar las estrategias de control, centrándonos en las técnicas de control borroso.

Por último se presentarán las conclusiones a las que se ha llegado y se expondrán nuevas líneas de trabajo en el proyecto realizado.

Índice general

Prefacio	I
Índice general	III
Índice de figuras	VII
Índice de tablas	VIII
1. Introducción	1
1.1. Objetivos del proyecto	2
1.2. Justificación	3
1.3. Estado del Control Borroso en la industria	3
2. Control Borroso	6
2.1. Control Borroso	6
2.1.1. Lógica borrosa	7
2.1.2. ¿Por qué utilizar lógica borrosa?	8
2.1.3. Conjuntos borrosos	9
2.1.3.1. Funciones de pertenencia	12
2.1.3.2. Variable lingüística	12
2.1.3.3. Operaciones borrosas	13
2.1.3.4. Inferencia borrosa	14
2.1.4. Sistemas de Control Borroso	15
2.1.4.1. Borrosificador (<i>Fuzzifier</i>)	16
2.1.4.2. Desborrosificador (<i>Defuzzifier</i>)	16
2.1.4.3. Diferentes controladores borrosos	17
2.2. Norma IEC 1131-7	18
2.2.1. IEC 1131 - Parte 7: Programación Control Borroso	22
2.2.1.1. Definiciones	24
2.2.1.2. Integración en autómatas programables	25
2.2.1.3. Lenguaje de control borroso FCL (<i>Fuzzy Control Lan- guage</i>)	26
2.2.1.4. Cumplimiento	27
2.3. Fuzzy Library	29
2.4. Fuzzy Logic Toolbox™	31

3. Caso de estudio	35
3.1. Planta piloto de laboratorio	35
3.1.1. Descripción de la planta	35
3.1.2. Reacción exotérmica	40
3.1.2.1. Modelado de la válvula	42
3.1.2.2. Modelo matemático	42
3.1.2.3. Identificación del sistema	44
3.2. Simulación de la planta piloto	45
3.2.1. Plataforma de interconexión Unity - Simulink	48
4. Control clásico aplicado a la planta	53
4.1. Teoría control clásico	53
4.1.1. Diseño del algoritmo	54
4.1.2. Sintonización de controladores PID	55
4.2. Simulink PID Controller Blocks	58
4.3. Diseño del controlador PID en Simulink para el modelo de la planta .	61
4.3.1. Simulaciones	65
4.4. Implementación del controlador PID en Unity	67
4.4.1. Introducción al software Unity Pro	67
4.4.1.1. Programación de la aplicación	69
4.4.2. Controlador PID	70
4.4.3. Simulaciones	74
4.4.3.1. Creación de un SCADA	74
4.4.4. Manual de uso de la plataforma Unity-Matlab	76
4.4.5. Resultados de las simulaciones	77
5. Control borroso directo aplicado a la planta	80
5.1. Control Borroso Incremental	81
5.2. Control borroso incremental en Simulink	84
5.2.1. Simulaciones	87
5.3. Implementación en Unity	90
5.4. Simulación	94
6. Conexión por OPC	99
6.1. OLE para Control de Procesos	99
6.1.1. Motivos para el desarrollo de la especificación OPC.	99
6.1.2. Origen de la especificación OPC	100
6.2. Configuración	101
7. Conclusiones	105
7.1. Conclusiones	105
7.2. Posibles ampliaciones para proyectos futuros	106
Anexos	107
Bibliografía	168

Índice de figuras

1.0.1.Aspecto de la planta piloto de laboratorio en la actualidad.	1
1.3.1.Algunos ejemplos de las numerosas aplicaciones que utilizan control borroso. (a) 1996 Saturn SL1 equipado con una transmisión automática controlada con lógica borrosa. (b) Helicóptero no tripulado. (c) Panasonic® olla arrocera donde la lógica borrosa controla el proceso de cocción. (d) ASML Water-Steppers.	3
2.1.1.Precisión y significado en el mundo real.	8
2.1.2.(a) Conjunto clásico. (b) Conjunto borroso. El conjunto de los días de la semana, sin duda, incluye los lunes, jueves y sábados. Es así como, sin duda, excluye la mantequilla, la libertad y las aletas dorsales, y así sucesivamente. Todo cae en cualquiera de los grupos, no hay una cosa que sea un día de la semana y no sea un día de la semana. Ahora, si consideramos el conjunto de días que comprende un fin de semana, la mayoría estaría de acuerdo en que los sábados y domingos pertenecen, pero ¿y el viernes? Se siente como una parte del fin de semana, pero de alguna manera parece que debería estar técnicamente excluido. Con los conjuntos clásicos no se tolera este tipo de clasificación. . . .	10
2.1.3.Ejemplo de un conjunto de personas donde se hace una distinción por altura.	10
2.1.4.Funciones de pertenencia para definir si una persona es alta o no. . . .	11
2.1.5.Términos principales de las funciones de pertenencia.	11
2.1.6.Formas comúnmente utilizadas de funciones de pertenencia.	12
2.1.7.Las principales fuentes y formas de formulación de reglas.	15
2.1.8.Funcionamiento del controlador Borroso.	16
2.2.1.Ejemplo de programación en LD.	20
2.2.2.Ejemplo de programación en FBD.	20
2.2.3.Ejemplo de programación IL.	21
2.2.4.Ejemplo de programación en ST.	21
2.2.5.Ejemplo de programación SFC.	22
2.2.6.Eschema contenido de la norma IEC 1131-7.	23
2.2.7.Ejemplo de un bloque de función de control borroso en representación FBD.	26
2.2.8.Intercambio de datos para programas en lenguaje de Control Borroso (FCL).	26
2.2.9.Ejemplo de bloque de función borroso.	27

2.2.10 Niveles de conformidad.	28
2.3.1. Ejemplos de bloques de funciones en representación FBD.	31
2.4.1. Diagrama del funcionamiento de la Fuzzy Logic Toolbox	32
2.4.2. Herramientas de la interfaz gráfica de usuario de la Fuzzy Logic Toolbox.	34
2.4.3. Funcionamiento de los tres editores.	34
3.1.1. Componentes de la planta piloto.	36
3.1.2. Fotografía del bastidor con el autómata programable M340 y sus módulos.	37
3.1.3. Sensor para el nivel del depósito.	38
3.1.4. Placa orificio de un caudalímetro eléctrico.	40
3.1.5. Esquemático de la planta piloto hecho con <i>Vijeo Citect</i>	40
3.1.6. Diagrama de la planta piloto con los cuatro elementos: reactor, intercambiador de calor, camisa de refrigeración y válvula de recirculación.	41
3.1.7. Característica Acción de control-Caudal de la válvula.	42
3.1.8. Reacción de la planta piloto a cambios en la entrada. Arriba la temperatura del tanque y abajo la apertura de la válvula.	45
3.2.1. Modelo de la planta desarrollado en Simulink.	46
3.2.2. Configuración de la condición inicial de un integrador.	46
3.2.3. Estructura interna del modelo de la planta en Simulink.	47
3.2.4. Subsistema que aproxima el caudal de entrada.	47
3.2.5. Estructura del subsistema que aproxima el intercambio de calor.	48
3.2.6. Respuesta de la temperatura a diferentes valores de apertura.	48
3.2.7. Esquema conexión Unity-Simulink por OPC.	49
3.2.8. Diagrama de flujo del programa de simulación del modelo.	49
3.2.9. Modelo de la planta desarrollado en Simulink preparado para la conexión por OPC.	50
3.2.10 Captura de pantalla del menú de simulación de Simulink.	50
3.2.1 Menú de configuración de parámetros de simulación de Simulink.	51
4.1.1. Diagrama de flujo de trabajo para el diseño de un PID.	55
4.1.2. Esquema controlador PID de una planta.	55
4.1.3. Curva de respuesta en forma de S.	57
4.1.4. Oscilación sostenida con periodo P_{cr}	57
4.2.1. Biblioteca de bloques de Simulink.	58
4.2.2. Parámetros del bloque de Simulink: Controlador PID	59
4.2.3. Parámetros avanzados del bloque PID	60
4.2.4. Linealización de la planta.	60
4.2.5. Ventana de sintonización del PID	61
4.3.1. Respuesta ante una entrada escalón unitario del sistema de primer orden.	62
4.3.2. Modelo de la planta con un controlador PID.	62
4.3.3. Pantalla de configuración de los parámetros del PID.	63
4.3.4. Resultado de utilizar la autosintonización de Simulink.	63
4.3.5. Pantalla para modificar el tiempo de respuesta.	64

4.3.6.Respuesta del sistema tras modificar el tiempo de respuesta.	64
4.3.7.Distintas respuestas variando el tiempo de respuesta dado por la autotuning. tosintonización.	65
4.3.8.Opciones para discretizar el controlador.	65
4.3.9.Resultados simulación 1.	66
4.3.10.Resultados simulación 2.	66
4.4.1.Pantalla de error al conectar el simulador de Unity Pro.	68
4.4.2.Panel de simulador del PLC cuando se produce el error.	68
4.4.3.Panel de simulador del PLC una vez cargado un proyecto.	68
4.4.4.Estructura de la aplicación en SFC.	69
4.4.5.Variables con dirección de memoria %MW fija.	70
4.4.6.Menú explorador de biblioteca de tipos de Unity.	70
4.4.7.Representación del bloque PIDFF en FBD.	71
4.4.8.Diagrama de estructura de controlador PIDFF.	73
4.4.9.Bloque PIDFF activado cada 2s.	74
4.4.10.Imagen del SCADA diseñado.	75
4.4.11.Creación de un SCADA en Unity.	75
4.4.12.Selección del modo de conexión: estándar o simulador.	76
4.4.13.Prueba 1. $K_p = -58,98$ y $T_i = 62,25s$	77
4.4.14.Prueba 2. $K_p = -58,98$ y $T_i = 15s$	78
4.4.15.Prueba 3. $K_p = -58,98$ y $T_i = 5s$	78
4.4.16.Respuesta del sistema en diferentes puntos de trabajo con: $K_p =$ $-58,98$, $T_i = 4ms$, $T_d = 16s$ y $T_s = 2s$	79
4.4.17.Respuesta del sistema en diferentes puntos de trabajo con: $K_p = -50$, $T_i = 4ms$ y $T_s = 2s$	79
5.1.1.Estructura general de un controlador borroso.	81
5.1.2.Estructura de un controlador borroso incremental.	81
5.1.3.Funciones de pertenencia de las entradas.	82
5.1.4.Funciones de pertenencia de la salida.	82
5.1.5.Respuesta ante entrada escalón y número de estado del proceso. . . .	83
5.2.1. <i>Fuzzy Logic Toolbox</i>	84
5.2.2.Controlador Borroso Incremental en Simulink.	84
5.2.3.Bloques que forman el controlador borroso.	85
5.2.4.Editor de Sistemas de Inferencia Borrosos.	85
5.2.5.Configuración para el control borroso incremental.	85
5.2.6.Ventana del Editor de Funciones de Pertenencia.	86
5.2.7.Ventana del Editor de Reglas.	86
5.2.8.Ventana del Visualizador de Reglas.	87
5.2.9.Comparación 1 entre el PI clásico y el PI Fuzzy.	88
5.2.10.Comparación 2 entre el PI clásico y el PI Fuzzy.	89
5.3.1.Explorador de librería de tipos.	90
5.3.2.Control borroso incremental implementado en Unity.	90
5.3.3.Representación borrosificador FUZ_ATERM en FBD.	91
5.3.4.Borrosificador en Unity.	92

5.3.5.Representación del bloque FUZ_MIN en FBD.	92
5.3.6.Conexión borrosificador - base de Reglas.	93
5.3.7.Representación desborrosificador DEFUZ en FBD.	93
5.3.8.Desborrosificador en Unity.	94
5.4.1.Selección del controlador borroso en el SCADA.	94
5.4.2.Simulación 1 con GE=0.1, GCE=0.5 y GCU=-117.96.	95
5.4.3.Simulación 2 con GE=0.1, GCE=6 y GCU=-9.83.	95
5.4.4.Simulación 3 con GE=0.1, GCE=25 y GCU=-0.79828.	96
5.4.5.Comparativa entre controladores clásico y borroso equivalentes.	97
6.1.1.Diagrama ejemplo aplicaciones sin y con OPC	100
6.2.1.Pantalla de configuración del OFS Configuration Tool de <i>Schneider Electric</i>	101
6.2.2.Pantalla de configuración IP del OFS Configuration Tools.	102
6.2.3.Código en Matlab para establecer la conexión por OPC.	103
6.2.4.Ventana del OPC Factory Server.	103

Índice de tablas

2.1.	Operaciones entre conjuntos borrosos.	13
2.2.	Funciones para aritmética INT y REAL en la biblioteca de control borroso.	30
3.1.	Listado de parámetros analógicos de la planta.	41
3.2.	Parámetros del modelo y variables constantes.	43
3.3.	Identificación de las variables de interés del modelo.	50
4.1.	Regla de sintonía de Ziegler-Nichols basada en la respuesta escalón de la planta (primer método).	57
4.2.	Regla de sintonía de Ziegler-Nichols basada en la ganancia crítica K_{cr} y periodo crítico P_{cr} (Segundo método).	58
4.3.	Parámetros Ziegler-Nichols PID.	62
4.4.	Parámetros Ziegler-Nichols PID.	63
4.5.	Parámetros del controlador PID en Simulink. Simulación 1.	66
4.6.	Parámetros del controlador PID en Simulink. Simulación 2.	66
4.7.	Descripción y valores del parámetro Para_PIDFF.	72
4.8.	Variables utilizadas en el bloque PIDFF.	74
5.1.	Números de estado y su correspondencia con las variables de entrada.	83
5.2.	Parámetros del controlador clásico y el borroso de la comparación 1.	88
5.3.	Parámetros del controlador clásico y el borroso de la comparación 2.	89