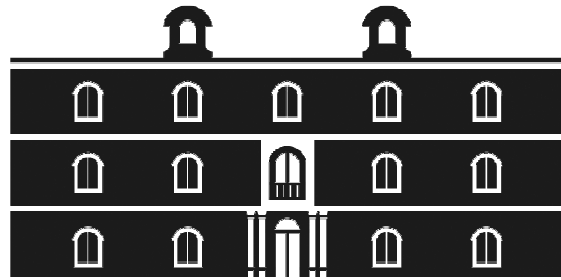


Universidad
Politécnica
de Cartagena



industriales
etsii UPCT

EMULADOR HIDRÁULICO DE SISTEMA DE CONTROL CON AUTÓMATA PROGRAMABLE

Titulación: Ingeniero en Automática y Electrónica Industrial
Alumno: D. Juan García Cuevas
Director: D. Sergio Gallardo Vázquez

Cartagena, 15 de Junio de 2010

Resumen

Este emulador ha sido desarrollado con componentes industriales de la firma Festo y con el presente proyecto se pretende realizar un documento de base para utilizar con fines didácticos, de forma que sirva como ejemplo de aplicación para simular situaciones reales de control, con componentes reales utilizados en la industria. El sistema está compuesto por un banco de trabajo móvil que representa un proceso hidráulico formado por varios depósitos abiertos y cerrados, que permiten la simulación de procesos industriales de regulación y control, así como la monitorización de los parámetros de temperatura, caudal, nivel y presión mediante el uso de un regulador industrial PID integrado en un PLC, utilizando como núcleo del sistema un autómata programable Simatic S7-300.

Este sistema ha sido desarrollado por la forma comercial Festo y debe utilizarse exclusivamente con fines de formación universitaria y profesional, en el campo de la automatización y las comunicaciones. Los usuarios y/o los instructores deben asegurar que los alumnos cumplen las medidas de seguridad adecuadas.

Índice de contenidos

1.- Características.....	7
<i>Datos técnicos básicos</i>	<i>7</i>
<i>Contenido para la formación</i>	<i>7</i>
<i>Estructura y aplicaciones.....</i>	<i>9</i>
<i>Técnicas de control</i>	<i>11</i>
2.- Análisis teórico, simulación y funciones de control	12
<i>Descripción del sistema de regulación de nivel</i>	<i>12</i>
<i>Régimen de funcionamiento.....</i>	<i>13</i>
<i>Resistencia hidráulica</i>	<i>14</i>
<i>Capacitancia hidráulica.....</i>	<i>15</i>
<i>Diagrama de bloques.....</i>	<i>16</i>
<i>Ecuaciones que definen el sistema de control de nivel</i>	<i>17</i>
<i>Determinación experimental de la función de transferencia</i>	<i>18</i>
<i>Ajuste del sistema en lazo cerrado</i>	<i>21</i>
<i>Precisión estática</i>	<i>22</i>
<i>Ajuste del regulador PI.....</i>	<i>26</i>
3.- El PLC S7 300	32
<i>Tipos actuales de PLC de Simatic.....</i>	<i>32</i>
<i>Características del lenguaje de programación Simatic [9]</i>	<i>33</i>
<i>Características del sistema modulas de la serie S7 300</i>	<i>36</i>
<i>Características de las CPU S7 300</i>	<i>37</i>
<i>Requisitos del Software Simatic S7</i>	<i>37</i>
<i>El software S7</i>	<i>38</i>
<i>Filosofía del uso de proyectos en el Administrador Simatic</i>	<i>39</i>
<i>La organización del programa en bloques</i>	<i>40</i>
<i>Clasificación de los bloques de organización (OB).....</i>	<i>42</i>
<i>Tratamiento de errores con bloques de organización (OB).....</i>	<i>42</i>
<i>Programas de una CPU S7-300 [8]</i>	<i>43</i>
<i>Pasos para la creación de un proyecto en Simatic S7</i>	<i>45</i>
4.- Calibrado de sensores	61

<i>Señales Normalizadas</i>	61
<i>Sensor de Nivel</i>	62
<i>Procedimiento de ajuste</i>	64
<i>Electro-Válvula Proporcional</i>	67
5.- Control PID con S7	70
<i>Introducción</i>	70
<i>Funciones de la librería S7</i>	70
<i>Función para regulación PID continua (FB 41)</i>	71
Rearranque completo / Rearranque	75
Parámetros de ENTRADA de la función FB 41	76
Parámetros de SALIDA de la función FB 41	84
<i>Características de los parámetros en los controladores PID</i>	85
<i>Sintonía de los parámetros del PID con S7</i>	86
<i>PID Self-Tuner TUNING_C</i>	86
Parámetros de entrada TUNING_C	88
Parámetros de salida TUNING_C	89
Parámetros de entrada/salida	90
Modos de Funcionamiento	90
Modo “Initial Controller Tuning”	91
Fases del proceso de auto-ajuste	91
Modo “Controller Adaptation to an Identified Process”	93
Fases del proceso	94
Controlador manual	94
Modificación de los parámetros del controlador	95
<i>Ajuste del tiempo de muestreo</i>	95
<i>Ejemplo de sintonización de un PID en un proceso continuo</i>	95
<i>PID Self-Tuner “FB_TUN_EC”</i>	97
<i>Requisitos del proceso</i>	98
<i>Descripción de los modos de funcionamiento FB “TUN_EC”</i>	99
6.- Bibliografía Utilizada	102

Índice de figuras:

<i>Figura 1: Diagrama PI del sistema completo.</i>	<i>10</i>
<i>Figura 2: Diagrama del sistema de control de nivel en B102.....</i>	<i>13</i>
<i>Figura 3: Diagrama de bloques del sistema de regulación de nivel.....</i>	<i>16</i>
<i>Figura 4: Respuesta real del sistema de control de altura del depósito. 19</i>	
<i>Figura 5: Esquema de control en lazo abierto para simulación con MatLab.....</i>	<i>20</i>
<i>Figura 6: Respuesta temporal del sistema en bucle abierto.</i>	<i>21</i>
<i>Figura 7: Diagrama de bloques del sistema modelado.....</i>	<i>21</i>
<i>Figura 8: Diagrama de ganancias estáticas.....</i>	<i>23</i>
<i>Figura 9: Esquema de control en lazo cerrado para simulación con MatLab.....</i>	<i>24</i>
<i>Figura 10: Error de precisión estática en bucle cerrado.....</i>	<i>24</i>
<i>Figura 11: Diagrama de bloques con representación de respuesta temporal.....</i>	<i>25</i>
<i>Figura 12: Esquema de control en lazo cerrado con regulador PI.</i>	<i>28</i>
<i>Figura 13: Esquema interno del regulador PI.</i>	<i>29</i>
<i>Figura 14: Respuesta temporal del sistema con regulador PI.</i>	<i>29</i>
<i>Figura 15: Esquema de control en lazo cerrado con regulador PI y control de saturación.....</i>	<i>30</i>
<i>Figura 16: Respuesta temporal del sistema con regulador PI y control de saturación.</i>	<i>30</i>
<i>Figura 17: Comparativa de respuesta temporal del sistema regulado con control PI con y sin control de saturación.</i>	<i>31</i>
<i>Figura 18: Gama de autómatas Simatic S7.</i>	<i>33</i>
<i>Figura 19: Árbol de directorios de un proyecto en el Administrador Simatic.</i>	<i>40</i>
<i>Figura 20: Proceso de creación de un proyecto.</i>	<i>45</i>
<i>Figura 21: Parametrización de un proyecto.</i>	<i>50</i>
<i>Figura 22: Ventana de parametrización de un proyecto.</i>	<i>51</i>
<i>Figura 23: Aplicación NetPro para configurar las redes de comunicación.</i>	<i>54</i>
<i>Figura 24: Contenido de un bloque de datos DB global.....</i>	<i>58</i>
<i>Figura 25: Características del sensor de nivel.</i>	<i>62</i>
<i>Figura 26: Ángulo máximo de detección del sensor de nivel.</i>	<i>63</i>
<i>Figura 27: Ajuste del sensor de nivel.....</i>	<i>63</i>

<i>Figura 28: Ajuste de escala para entrada analógica.....</i>	<i>65</i>
<i>Figura 29: Válvula de control proporcional.....</i>	<i>67</i>
<i>Figura 30: componentes de ajuste de válvula.....</i>	<i>67</i>
<i>Figura 31: Diagrama de bloques del regulador PID (FB41).....</i>	<i>72</i>
<i>Figura 32: Regulador PID, rama de valor real.</i>	<i>73</i>
<i>Figura 33: Regulador PID, Error de regulación.....</i>	<i>74</i>
<i>Figura 34: Regulador PID, Algoritmo.</i>	<i>74</i>
<i>Figura 35: Regulador PID, procesado de valores manipulados.....</i>	<i>75</i>
<i>Figura 36: Ejemplo de utilización de la función Tuning_C.</i>	<i>96</i>
<i>Figura 37: ejemplo de proceso de control.....</i>	<i>96</i>

1.- Características

Datos técnicos básicos

- Tensión de alimentación externa: 240 V AC.
- Tensión de distribución interna: 24 V DC.
- Control: Autómata programable Simatic S7-300.

Sensores:

- Sensor de ultrasonidos (salida de corriente, 4-20mA).
- Caudalímetro de paletas (salida de frecuencia, 0-1000Hz).
- Transductor de presión (salida de tensión, 0-10V).
- Sonda de temperatura (Pt100).

Actuadores:

- Bomba con velocidad regulable mediante PWM (0-10V).
- Calefactor con mando a relé.
- Válvula proporcional (0-10V).
- Válvula electro-neumática 2/2.

Dimensiones:

El sistema está montado sobre un panel de aluminio de 700x700x32mm, con ranuras de fijación cada 50mm., sujeto sobre una mesa metálica rodante de 350x700x750mm., con cuatro ruedas orientables, dos de ellas con freno.

Contenido para la formación

Mecánica:

- Ensamblado y ajuste de elementos.
- Instalación de los componentes (sensores, actuadores, etc....).

Electricidad:

- Cableado de componentes.
- Señales de control y medida.
- Estandarización de esquemas eléctricos y de instalación.
- Aplicaciones de los diferentes tipos de sensores.
- Calibrado y ajuste de sensores.
- Localización sistemática de averías en un sistema de control.

Regulación:

- Tipos de regulación industrial.
- Control en lazo abierto y lazo cerrado.
- Principios de la regulación P, I, D.
- Reguladores P, PI, PD, PID.
- Ajuste de reguladores industriales.
- Autotuning, control adaptativo.
- Alarmas de proceso.

Componentes de la estación:

- Módulo de interfase de E/S.
- Sensor de caudal.
- Sensor de temperatura.
- Sensor analógico ultrasónico de distancia.
- Sensor analógico de presión.
- Electro-válvula proporcional.
- Bomba centrífuga.
- Variador de velocidad del motor de la bomba.
- Calefactor de resistencia.
- Autómata programable.
- Depósitos graduados abiertos de 15 litros, 300 mm.
- Depósito de presión.

Estructura y aplicaciones

La estructura del sistema es modular, se mantienen separados, como elementos independientes:

- La máquina: Aquí se encuentran los elementos mecánicos, sensores y actuadores que componen la estación.
- El control: El cuadro eléctrico con la alimentación y el sistema de control (PLC), los adaptadores de señal y la fuente de alimentación.
- El mando: El Panel de Mando, con los pulsadores y pilotos indicadores.

Las señales de la estación están centralizadas en un bornero de conexión, que permiten la conexión mediante un sistema estandarizado de cableado, permitiendo el cambio de autómatas o controladores, con una sencilla operación.

La estación combina cuatro lazos de control con sensores y actuadores analógicos y digitales. Mediante el controlador adecuado será posible utilizarlos de forma individual o combinada con control en cascada.

- Control de nivel
- Control de caudal
- Control de presión

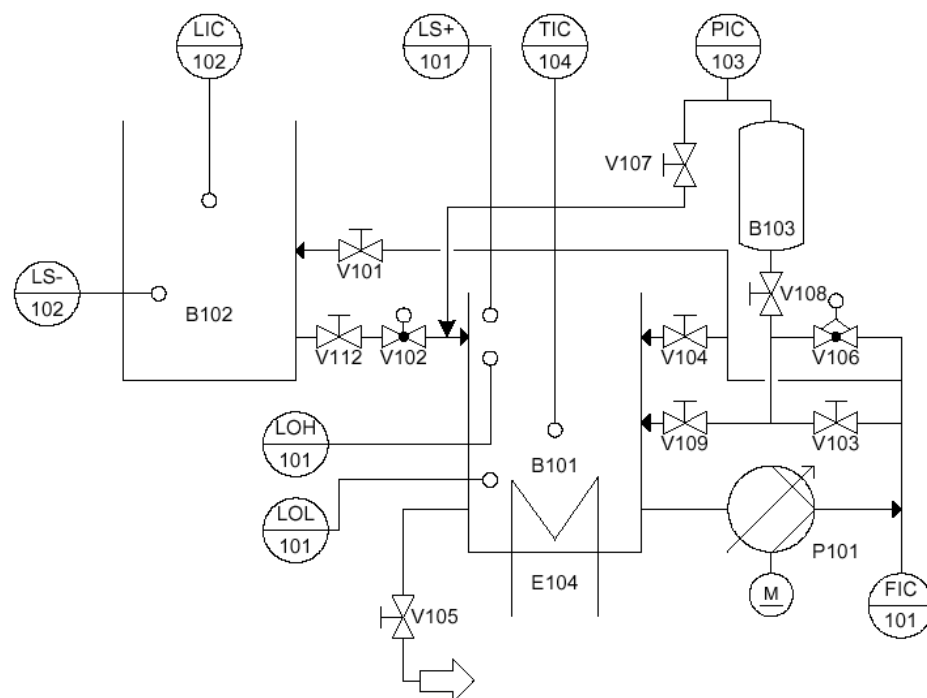


Figura 1: Diagrama PI del sistema completo.

Según la configuración adoptada en la estación, será posible realizar:

- Control continuo de nivel

Magnitud controlada: Nivel del depósito 2.

Sensor: de nivel analógico por ultrasonido.

Actuador controlado: Motor-bomba con variador de frecuencia para control de velocidad.

- Control continuo de caudal

Magnitud controlada: Caudal circulante entre depósitos 1 y 2.

Sensor: de caudal por pulsos (frecuencia) o (convertor de frecuencia a tensión).

Actuador: Motor-bomba con variador de frecuencia para control de velocidad.

Actuador: electro-válvula proporcional.

- Control continuo de presión

Magnitud controlada: Presión en el depósito cerrado 3.

Sensor: de presión analógico.

Actuador: electro-válvula proporcional.

Técnicas de control

Los lazos de control disponibles, permiten realizar numerosas técnicas de control, comenzaremos aplicando al control de nivel, la regulación "PI" por reubicación de polos, técnicas de auto-sintonía de parámetros, incluso otras técnicas avanzadas de control, que se podrían utilizar en un futuro como continuación de este proyecto, tales como el control robusto y el control adaptativo.

2.- Análisis teórico, simulación y funciones de control

La regulación automática de una magnitud física consiste en mantener dicha magnitud dentro de un estado de equilibrio previamente definido, así se dice que un proceso es autorregulado cuando, partiendo de un estado de equilibrio se alcanza un nuevo estado de equilibrio después de una variación de la carga.

Antes de la simulación abordaremos el análisis teórico de la autorregulación de nivel de líquido en un depósito abierto, para conseguir que el sistema de control mantenga constante el nivel, cuando aparezcan variaciones en la salida o entrada de caudal al depósito.

Descripción del sistema de regulación de nivel

Los elementos básicos del sistema regulador de nivel son:

- Los dos depósitos (B101 y B102)
- El motor-bomba (P101)
- El sensor analógico de nivel por ultrasonidos (LIC/102)

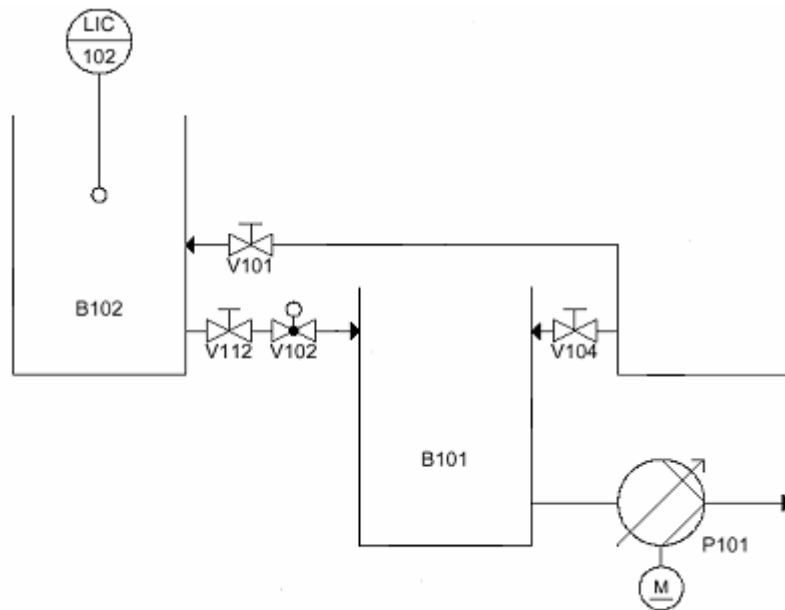


Figura 2: Diagrama del sistema de control de nivel en B102.

Además de estos elementos básicos para identificar el lazo de regulación, se encuentran entre otros las válvulas para regular los caudales de entrada y salida de los depósitos, (V101, V102, V104 y V112), así como algunos sensores digitales utilizados para detectar los niveles máximos y mínimos de los depósitos con el fin de evitar el deterioro de la bomba al trabajar en vacío, el desbordamiento del depósito o la lectura errónea del sensor de nivel cuando el depósito tiene un nivel mínimo de líquido.

Régimen de funcionamiento

Al analizar los sistemas que implican flujo de líquidos a través de tuberías, resulta necesario diferenciar el comportamiento según el régimen de funcionamiento, entre régimen laminar y turbulento, de acuerdo con la magnitud del número de Reynolds. Se ha demostrado experimentalmente que el régimen es laminar para un número de Reynolds inferior a 3000, y turbulento para un número superior a 4400. y cuando el número de Reynolds está entre 3000 y 4400 se dice que el régimen es de transición entre laminar y turbulento.

El número de Reynolds es adimensional $R = \frac{\rho \cdot D \cdot v}{\eta}$

Donde: **D** es el diámetro del tubo (mm), **ρ** la densidad del fluido (kg/m^3), **η** la viscosidad del fluido, y **v** su velocidad.

Resistencia hidráulica

Supongamos un tubo corto que une los dos depósitos, la resistencia para el flujo del líquido en el tubo se define como el cambio en la diferencia de nivel de los depósitos necesaria para producir un cambio de una unidad en la velocidad del flujo:

$$R = \frac{\text{cambio en la diferencia de nivel (m)}}{\text{cambio en la velocidad del flujo (m}^3/\text{s)}}$$

Debido a que la relación entre la diferencia de nivel y la velocidad del flujo son diferentes, para el flujo laminar y el turbulento.

Para el caso de **flujo laminar** en régimen estable, la variación del cambio de velocidad de flujo es proporcional a la diferencia de nivel.

$$Q = K.H$$

Siendo: **Q** la velocidad del flujo en estado estable (m^3/s), **K**, coeficiente de proporcionalidad (m^2/s), **H**, diferencia de altura de los depósitos.

Esta ecuación es análoga a la resistencia eléctrica definida por la Ley de Ohm, que dice que, en un elemento resistivo la diferencia de potencial es directamente proporcional a la corriente eléctrica que lo atraviesa.

Por tanto, para el flujo laminar, la resistencia hidráulica **R** es:

$$R = \frac{dH}{dQ} = \frac{H}{Q}$$

Para el caso de **flujo turbulento** en régimen estable, la variación del cambio de velocidad de flujo es proporcional a la raíz cuadrada de la diferencia de nivel.

$$Q = K \cdot \sqrt{H}$$

Podemos linealizar la ecuación obtenida en torno al punto de funcionamiento (Q', H') .

$$dQ = \frac{K}{2\sqrt{H}} dH$$
$$\left. \frac{dH}{dQ} = \frac{2\sqrt{H}}{K} \right|_{(H', Q')} = \frac{2\sqrt{H'} \sqrt{H'}}{Q} = \frac{2H'}{Q}$$

Por tanto, para el flujo turbulento en régimen estable, la resistencia hidráulica **R** es:

$$R = \frac{2H}{Q}$$

La relación obtenida en este caso sólo será válida para pequeñas variaciones en torno al punto de funcionamiento.

Capacitancia hidráulica

La capacidad volumétrica de un depósito hidráulico es la máxima cantidad de volumen de fluido que puede almacenar.

La capacitancia hidráulica, **C**, se define como el cambio en la cantidad de líquido almacenado para producir un cambio en una unidad de altura (potencia).

$$C = \frac{\text{cambio en el líquido almacenado (m}^3\text{)}}{\text{cambio en la altura del líquido (m)}}$$

El potencial es la cantidad que indica el nivel de energía acumulado del sistema, en el caso de depósitos de sección constante, la capacitancia es el volumen partido de la altura, lo que equivale al área del de la sección transversal.

Diagrama de bloques

El diagrama simplificado de los elementos básicos que intervienen en el regulación de nivel del depósito B102 es el siguiente:

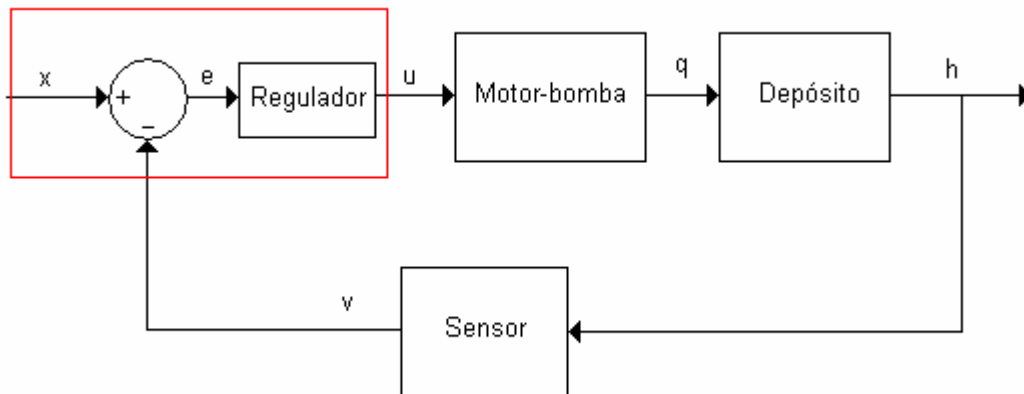


Figura 3: Diagrama de bloques del sistema de regulación de nivel.

En este diagrama se pueden identificar las siguientes señales:

- x: Consigna de nivel deseado.
- h: Nivel real del depósito B102.
- q: Caudal de líquido introducido en el depósito.
- v: Señal de nivel del depósito.
- e: Señal de error, diferencia entre consigna y valor real.
- u: Señal de control del motor para el grupo motor-bomba.

Ecuaciones que definen el sistema de control de nivel

Para calcular las ecuaciones que definen el funcionamiento dinámico del sistema, se ha supuesto que los bloques que lo forman son lineales o están linealizados en torno al punto de funcionamiento.

El depósito B102, es el elemento en el que queremos controlar la altura (variable controlada), y, suponiendo que las variaciones de flujo de entrada y salida son lentas:

$$\Delta q = \frac{\Delta V}{\Delta t} \quad q_i - q_o = \frac{C.dH}{dt}$$

Suponiendo régimen laminar: $q_o = \frac{H}{R}$

$$R.q_i = \frac{R.C.dH}{dt} + H$$

Esta ecuación diferencial lineal puede resolverse aplicando la teoría clásica de control. Suponiendo las condiciones iniciales nulas, y tomando como entrada (q_i) y como salida (h), la función de transferencia que modeliza el bloque “depósito” es:

$$\frac{H(s)}{Q_i(s)} = \frac{R}{R.C.s + 1}$$

Función de transferencia que representa un sistema de primer orden con constante de tiempo ($\tau = R.C$), en esta ecuación el parámetro R, varará en función del estado de las válvulas de salida del depósito, circunstancia que habrá que tener en cuenta en cada caso.

El Motor-bomba es el elemento sobre el que actuarán las señales de control para aportar el caudal necesario al depósito, su función de transferencia la supondremos lineal y constante:

$$\frac{Q_i(s)}{U(s)} = K_M$$

Constante por determinar de los datos experimentales obtenidos.

El Sensor de ultrasonidos es el elemento que capta la altura actual del depósito (variable controlada) y obtiene una señal equivalente de corriente normalizada en el rango 4 – 20 mA., su función de transferencia la supondremos lineal y constante:

$$\frac{V(s)}{H(s)} = K_S$$

Constante por ajustar en función del rango de altura que se pretenda controlar.

Determinación experimental de la función de transferencia

La función de transferencia del conjunto motor, bomba, depósito, que llamaremos sistema, es de 1º orden y tras realizar algunas comprobaciones para modelar las constantes R y C se ha obtenido la siguiente tabla de datos experimentales del sistema:

Para una entrada escalón de la bomba pasando de 0 a 10 V la lectura de tiempo y altura del depósito es la siguiente:

t(s)	0	16	44	64	81	94	115	140	171	225	273
h(mm)	15	35	65	85	100	110	125	140	155	175	185

La gráfica obtenida es:

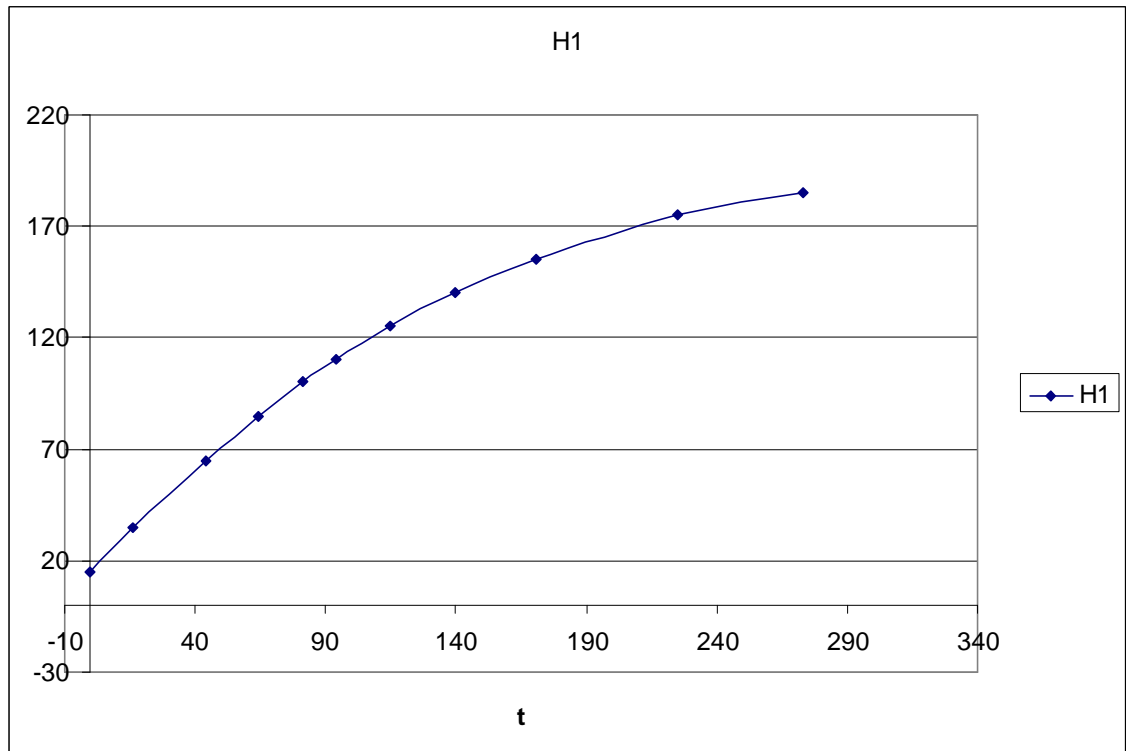


Figura 4: Respuesta real del sistema de control de altura del depósito.

Que corresponde aproximadamente a la respuesta de un sistema de 1^o orden para una entrada escalón. La función de transferencia buscada se caracteriza por dos parámetros básicos, que son la ganancia K y la constante de tiempo τ .

Para obtener la función de transferencia de la forma:

$$\frac{H(s)}{U(s)} = \frac{K}{\tau \cdot s + 1}$$

Obtendremos el valor de K, ganancia estática, como el cociente entre:

$$K = \frac{190 - 15}{100} = \frac{175}{100} = 1.75$$

El valor de τ se obtiene calculando el tiempo en el que la señal de salida del sistema tarda en alcanzar el 63,2% del valor final. Es este caso corresponde al valor:

$$(190-15)*0.632 + 15 = 125.6 \text{ mm}$$

$$T(125.6 \text{ mm}) = 115 \text{ s}$$

$$\tau = 115$$

La función de transferencia es aproximadamente:

$$\frac{H(s)}{U(s)} = \frac{1.75}{115s + 1}$$

Utilizando esta función de transferencia en MatLab para simular el comportamiento del sistema obtendremos el siguiente diagrama:

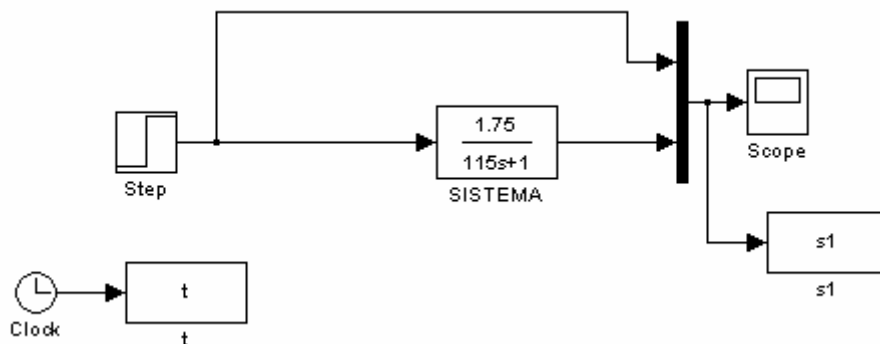


Figura 5: Esquema de control en lazo abierto para simulación con MatLab.

La respuesta temporal de este modelo en MatLab se obtiene con las siguientes instrucciones:

```
>> hold on  
>> plot(t,s1(:,1),'b')  
>> plot(t,s1(:,2),'r')
```

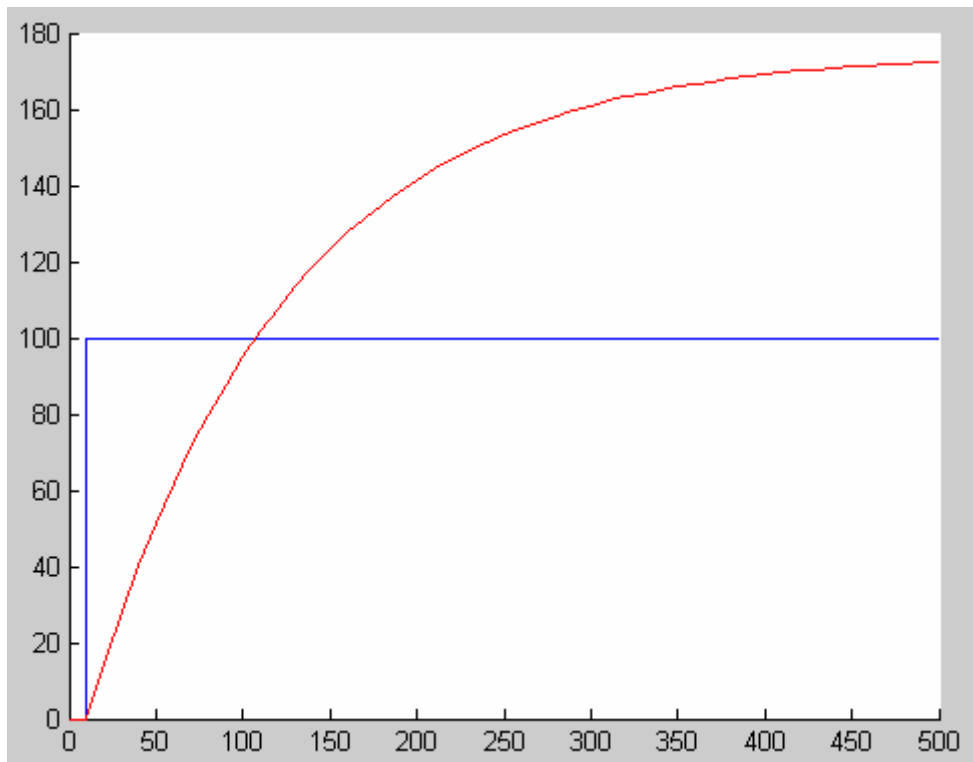


Figura 6: Respuesta temporal del sistema en bucle abierto.

Ajuste del sistema en lazo cerrado

El diagrama de referencia que utilizaremos es el siguiente, en el se han incluido en un solo bloque el grupo motor bomba y depósito, cuya función de transferencia es la que acabamos de obtener:

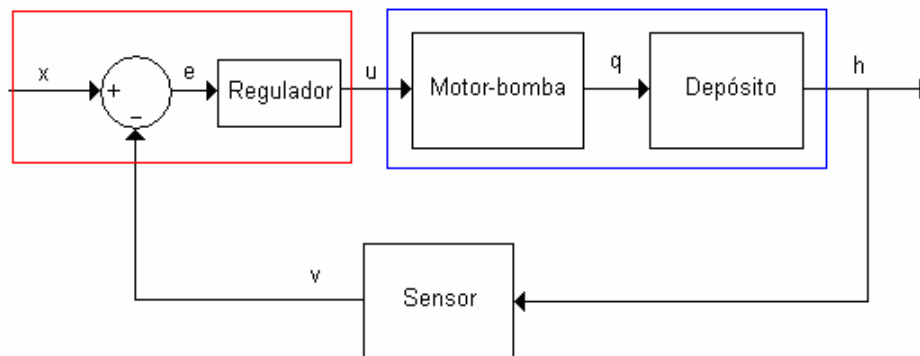


Figura 7: Diagrama de bloques del sistema modelado.

El sensor estará modelizado en MatLab como un bloque proporcional cuya salida nos dará la lectura directa en milímetro de la altura de líquido actual del depósito, por tanto su función de transferencia será una constante unitaria

$$\frac{V(s)}{H(s)} = 1$$

Para ajustar las unidades del sistema formado por el motor, la bomba y depósito, de forma que la variable de entrada sea mm de altura introduciremos una constante de proporcionalidad, como el cociente entre el valor máximo de trabajo de la bomba, 100%, y la altura máxima que se puede alcanzar, 295 mm.

Así el conjunto del sistema está modelizado por la función de transferencia:

$$\frac{H(s)}{U(s)} = \frac{\frac{100}{295} \cdot 1.75}{115s+1} = \frac{0.59}{115s+1}$$

Precisión estática

Si a un proceso se le aplica una cierta entrada, por ejemplo un escalón, y su respuesta o salida alcanza después de un cierto tiempo un régimen de equilibrio, el proceso será intrínsecamente estable y es denominado “auto-regulable”. Si por el contrario el proceso no alcanza un régimen estable será denominado proceso “no auto-regulable”.

Ambos procesos pueden ser estabilizados cerrando el lazo de realimentación, aunque la mayoría de los procesos industriales que nos ocupan en este estudio son procesos auto-regulados, como sucede con nuestro sistema de control de altura.

La precisión de un sistema es la capacidad que este tiene para que la variable controlada siga fielmente a la consigna. Así pues un sistema de

control será muy preciso cuando la diferencia entre la variable manipulada y la consigna sea mínima o nula.

Esta diferencia también se llama error de regulación, e , y se obtiene a la salida del comparador, $e = x - v$.

Así pues se define en régimen de equilibrio la **precisión estática** como la relación entre el error de regulación y la consigna, y se expresa en tanto por ciento.

$$pe = \frac{e(t)}{x(t)} \times 100$$

Simbolizando las ganancias estáticas de cada uno de los bloques que componen el sistema, el diagrama es el siguiente:

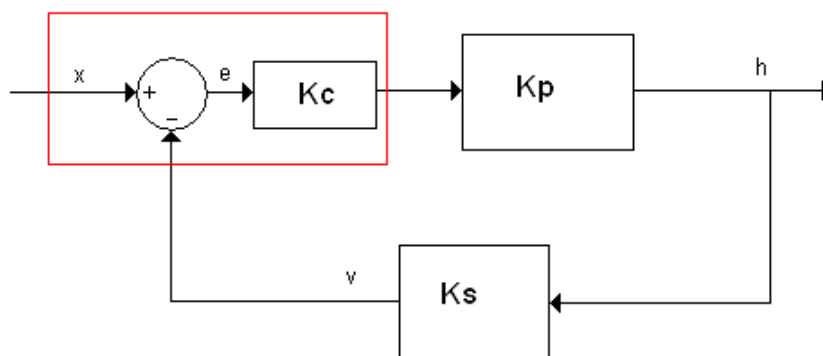


Figura 8: Diagrama de ganancias estáticas.

En este caso se ha simplificado el sistema en tres bloques, controlador, **Kc**, proceso, **Kp**, y sensor, **Ks**.

$$e = x - v$$

$$e = x - e(Kc * Kp * Ks)$$

$$e(1 + Kc * Kp * Ks) = x$$

$$pe = \frac{e}{x}$$

$$pe = \frac{1}{1 + Kc * Kp * Ks} * 100$$

Al cerrar el lazo de control el conjunto quedará modelizado por el siguiente diagrama de bloques:

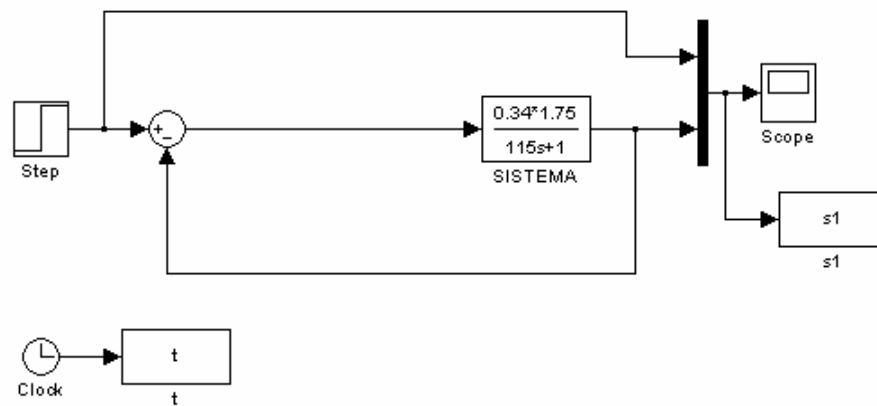


Figura 9: Esquema de control en lazo cerrado para simulación con MatLab.

En la gráfica de respuesta, para una consigna de 100 mm, puede observarse una reducción de la ganancia estática del sistema, con el consiguiente aumento del error de precisión, debido precisamente al hecho de cerrar el lazo de realimentación:

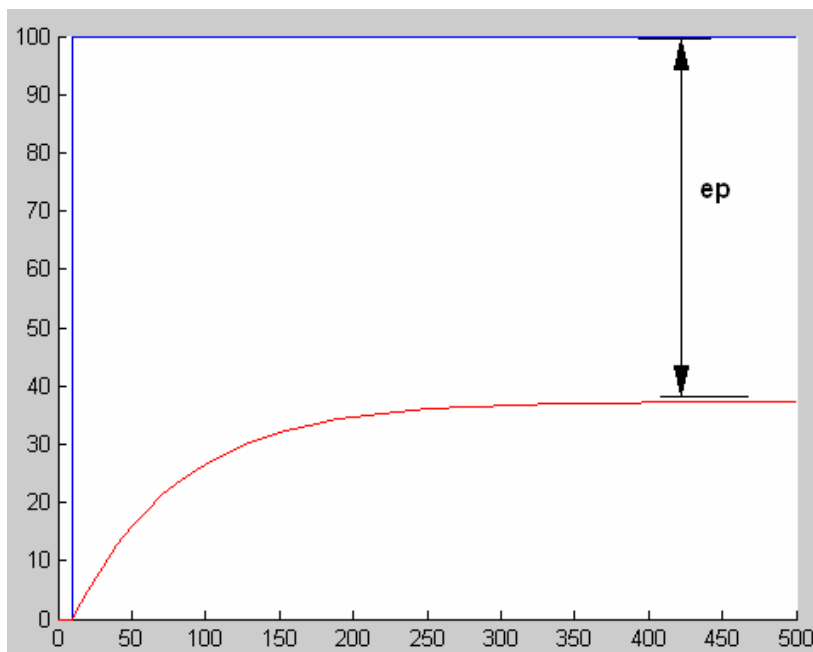


Figura 10: Error de precisión estática en bucle cerrado.

La precisión estática del sistema en bucle cerrado puede calcularse como en cociente entre la señal de error y la consigna, siendo la señal de error la diferencia entre la consigna y la señal medida, si representamos cada uno de los bloques por su ganancia estática

$$pe = \frac{1}{1 + Kc * Kp * Ks} * 100$$

$$pe = \frac{1}{1 + 0.34 * 1.75} * 100$$

$$pe = 63\%$$

Lo que supone un error considerable del **63 %**, este efecto se ha producido al cerrar el lazo de realimentación, como efecto negativo ha aumentado el error de regulación y como efecto positivo se ha reducido el tiempo de establecimiento a casi la mitad.

Un diagrama mas intuitivo consiste es sustituir los bloques que representan una función de transferencia por bloques que representa la respuesta temporal, para nuestro sistema, el esquema será:

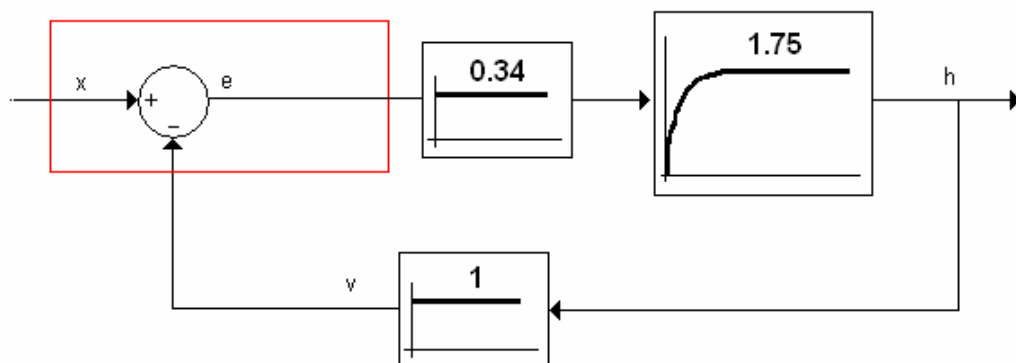


Figura 11: Diagrama de bloques con representación de respuesta temporal.

Ajuste del regulador PI

Existen numerosas técnicas para el ajuste de los parámetros de un regulador PID, desde las puramente intuitivas basadas en la observación de la respuesta en lazo cerrado, las basadas en la observación de la respuesta en lazo abierto y las puramente empíricas basadas en la definición de las especificaciones de tiempo del proceso controlado, utilizaremos ahora el método empírico basado en las especificaciones temporales de la respuesta del proceso controlado ya que existe una relación estrecha entre el tiempo de respuesta deseado y la posición de los polos del sistema realimentado.

La elección del tipo de controlador y el ajuste de sus parámetros es determinante para que el proceso controlado cumpla adecuadamente las características particulares de funcionamiento, como son el grado de precisión, la sensibilidad, la estabilidad y la rapidez de respuesta. Por ejemplo la elevada ganancia de un controlador proporcional permite una gran precisión estática pero una pérdida de estabilidad.

El tiempo de respuesta de un proceso controlado, es el tiempo necesario hasta alcanzar el equilibrio en su salida cuando se aplica una consigna escalón. Este tiempo puede asemejarse a la constante de tiempo, **T**, de un sistema de 1º orden o al tiempo de estabilización de un sistema de 2º orden.

Por otra parte, el parámetro **T**, está íntimamente relacionado con los polos de la ecuación característica, cuyo valor para un sistema de 1º orden es:

$$1 + Ts = 0 \quad \text{polo en} \quad s = -\frac{1}{T}$$

T, es el tiempo en el que se alcanza el 63% del valor final de la respuesta, si se define, **T₉₈**, como el tiempo necesario para alcanzar el

98 % del valor final, se puede demostrar que $T_{98} = 4 * T$, y por tanto su

polo equivale a $s = -\frac{4}{T_{98}}$

En un sistema de segundo orden o superior siempre suele existir un tiempo de respuesta o polo dominante. Dicho de otro modo los polos que están próximos a cero dominan la respuesta.

En resumen definir el tiempo de respuesta deseado equivale a fijar la posición de los polos del sistema realimentado.

Para nuestro sistema incluido el regulador **PI** la función de transferencia completa será:

$$G = \frac{G_R * G_P}{1 + G_R * G_P}$$

$$G_R = K_p \left(1 + \frac{1}{T_I s} \right) \quad G_P = \frac{K}{1 + T s}$$

$$G = \frac{K_p * K (1 + T_I s)}{K_p * K + (T_I + T_I * K_p * K) s + (T_I * T) s^2}$$

$$G = \frac{1 + T_I s}{1 + \frac{T_I + T_I * K_p * K}{K_p * K} s + \frac{T_I * T}{K_p * K} s^2}$$

La ecuación característica que define la posición de los polos es:

$$K_p * K + (T_I + T_I * K_p * K) s + (T_I * T) s^2 = 0$$

Si se desea que el sistema realimentado tenga una respuesta temporal sin oscilaciones, la parte imaginaria de los polos debe ser nula, y si se especifica un tiempo de establecimiento, T_s , entonces la parte real de los

polos debe ser igual a $-\frac{4}{T_s}$, el sistema de dos ecuaciones con dos

incógnitas es el siguiente:

$$\left. \begin{aligned} (T_I + T_I * K_P * K)^2 - 4(T_I * T * K_P * K) &= 0 \\ -\frac{T_I + T_I * K_P * K}{2T_I T} &= -\frac{4}{T_s} \end{aligned} \right\} \text{siendo las incógnitas } T_I \text{ y } K_P$$

$$K_P = \frac{8T - T_s}{K * T_s}$$

$$T_I = \frac{8T * T_s - T_s^2}{16T}$$

Sustituyendo valores tenemos:

$$\left\{ \begin{aligned} K_P &= \frac{8 * 115 - 50}{0.6 * 50} = 29 \\ T_I &= \frac{8 * 115 * 50 - 50^2}{16 * 115} = 23.6 \end{aligned} \right.$$

El sistema regulado quedara formado por los siguientes bloques:

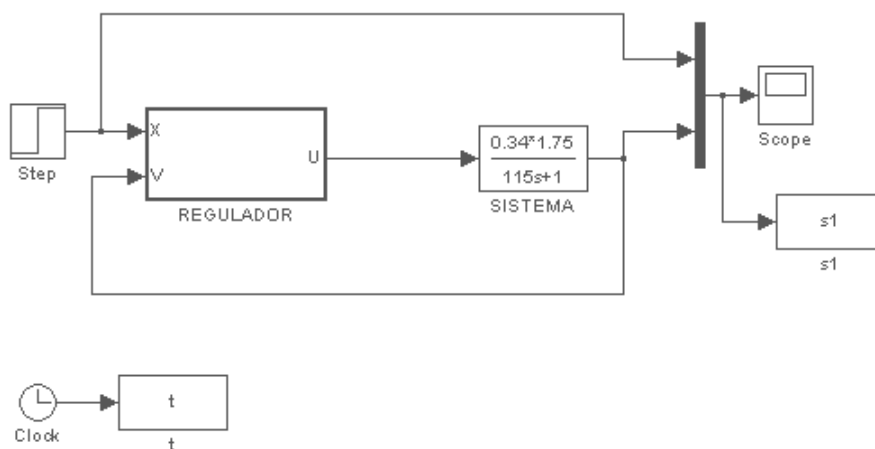


Figura 12: Esquema de control en lazo cerrado con regulador PI.

El regulador estará formado por:

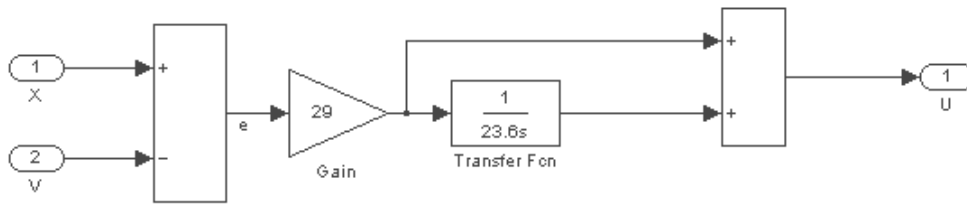


Figura 13: Esquema interno del regulador PI.

Y la respuesta temporal del sistema ante un escalón es ahora:

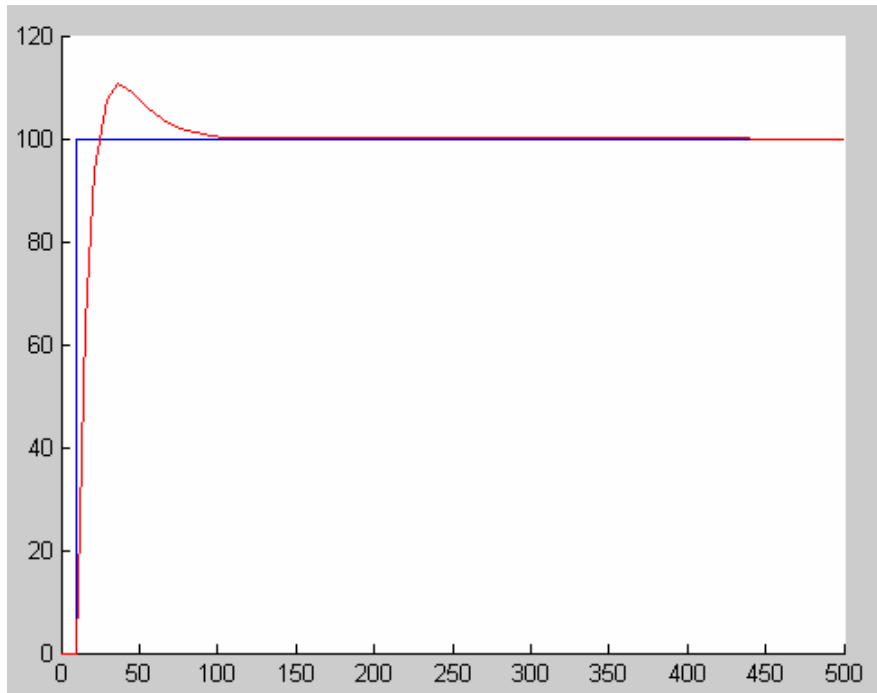


Figura 14: Respuesta temporal del sistema con regulador PI.

Donde puede apreciarse que se ha conseguido una regulación muy precisa, en teoría, pues se trata de una simulación. Hay que tener en cuenta que en el modelo se han introducido numerosas aproximaciones y sobre todo no se ha tenido en cuenta la saturación de las señales de control, así pues si introducimos un bloque que limite la señal de control a los valores límites la respuesta se aproximaría algo mas a la realidad.

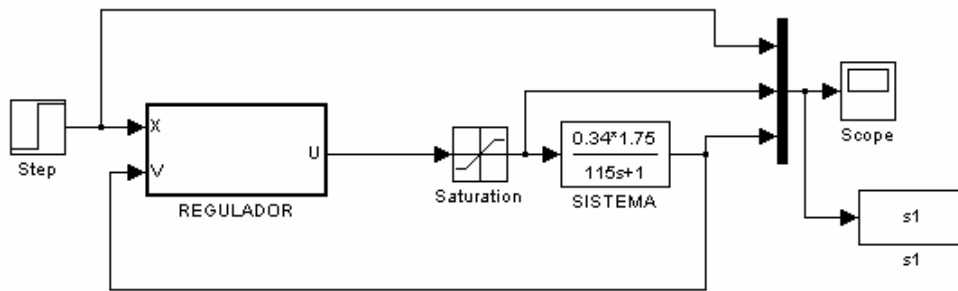


Figura 15: Esquema de control en lazo cerrado con regulador PI y control de saturación.

Al introducir las condiciones de saturación máximas de la señal de control sobre la bomba se modifica sustancialmente la respuesta temporal del sistema, en la siguiente gráfica puede verse esta respuesta incluida la señal de control introducida en la bomba.

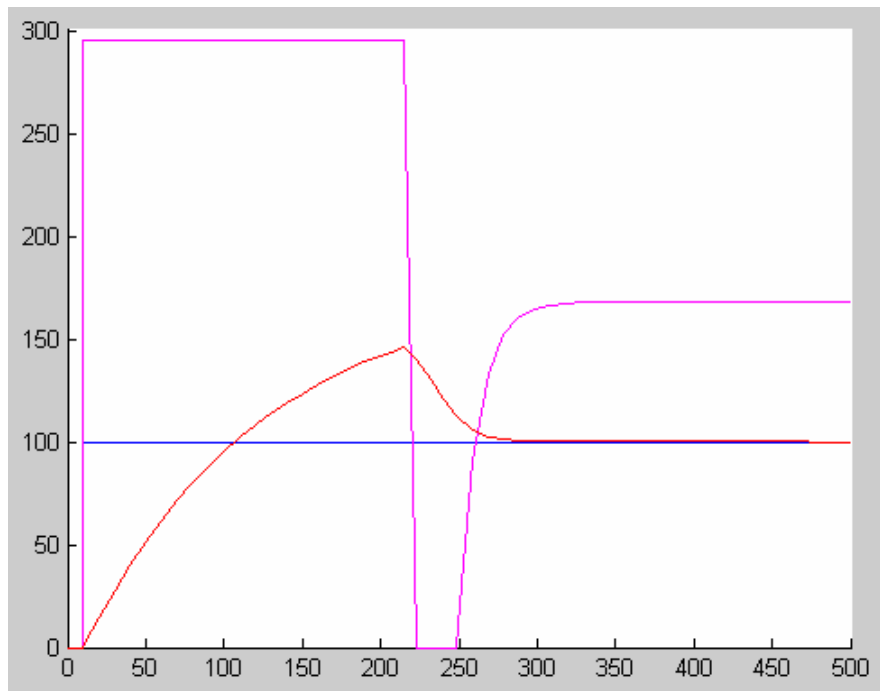


Figura 16: Respuesta temporal del sistema con regulador PI y control de saturación.

Si comparamos ambas respuesta con y sin control de la saturación la grafica será:

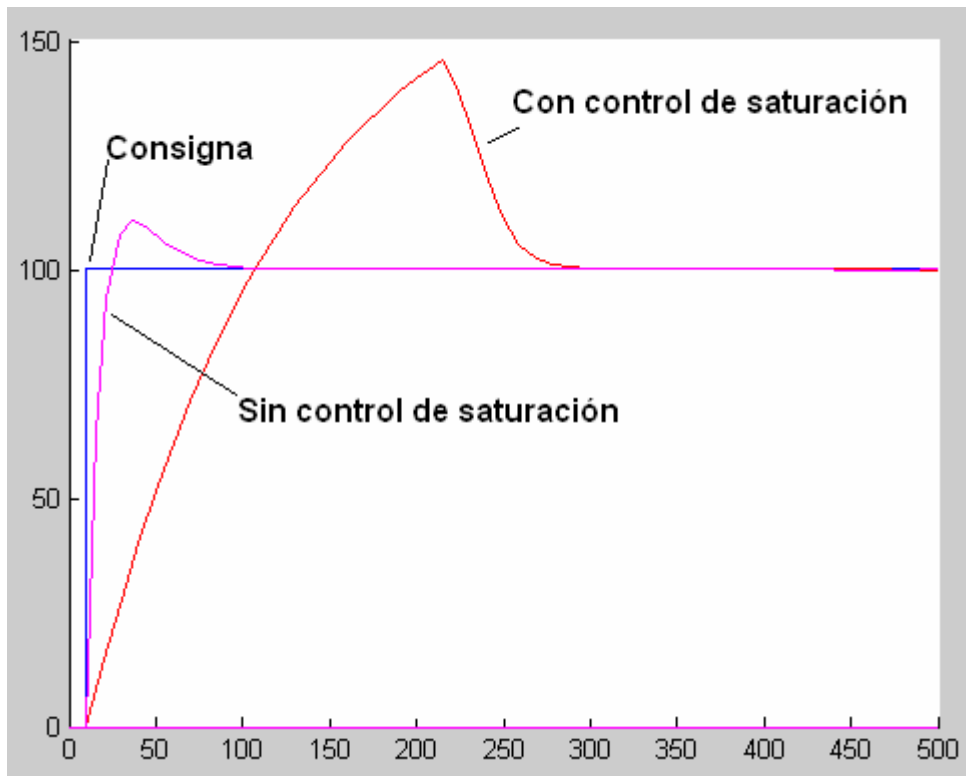


Figura 17: Comparativa de respuesta temporal del sistema regulado con control PI con y sin control de saturación.

La diferencia es considerable, por lo que como conclusión se deduce que la situación permite avanzar muchísimo en el diseño de sistemas de control pero sin olvidar los parámetros y las restricciones que deben limitar nuestro modelo para que reproduzca lo más fielmente posible taran el sistema como el entorno.

3.- EI PLC S7 300

Los equipos Simatic S7 son sin lugar a dudas sistema de los más avanzado en el campo de la automatización de procesos mediante PLC. En el mercado existen numerosos modelos PLC, pero básicamente podemos clasificarlos en dos grandes tipo, los de mapa de memoria única y los de múltiples mapas de memoria. Los primeros de mapa de memoria única son equipos sencillos de programar para aplicaciones sencillas y todas sus zonas de memoria se encuentran contiguas en un único mapa de memoria, distribuido en bloques lógicos, entre los que se encuentran los bloques de área de programa, área de datos, área de dispositivos, etc... Por otro lado se encuentran los PLC más complejos, basados en múltiples mapas de memoria, uno para albergar programa, datos, entradas, salidas, dispositivos, etc... Un ejemplo de los primeros son los autómatas Omron y un ejemplo de los segundos son los Simatic.

Tipos actuales de PLC de Simatic [5].

En la actualidad SIMATIC S7 se compone de tres gamas de sistemas de automatización que se distinguen por sus prestaciones.

EI SIMATIC S7-200	Es un Micro PLC compacto para la gama inferior de prestaciones. Este autómata dispone de un paquete de software propio ya que las características del sistema S7-200 no permiten realizar una conversión mediante software.
EI SIMATIC S7-300	Es el mini-autómata modular para la gama media.
EI SIMATIC S7-400	Cubre la gama alta y superior.

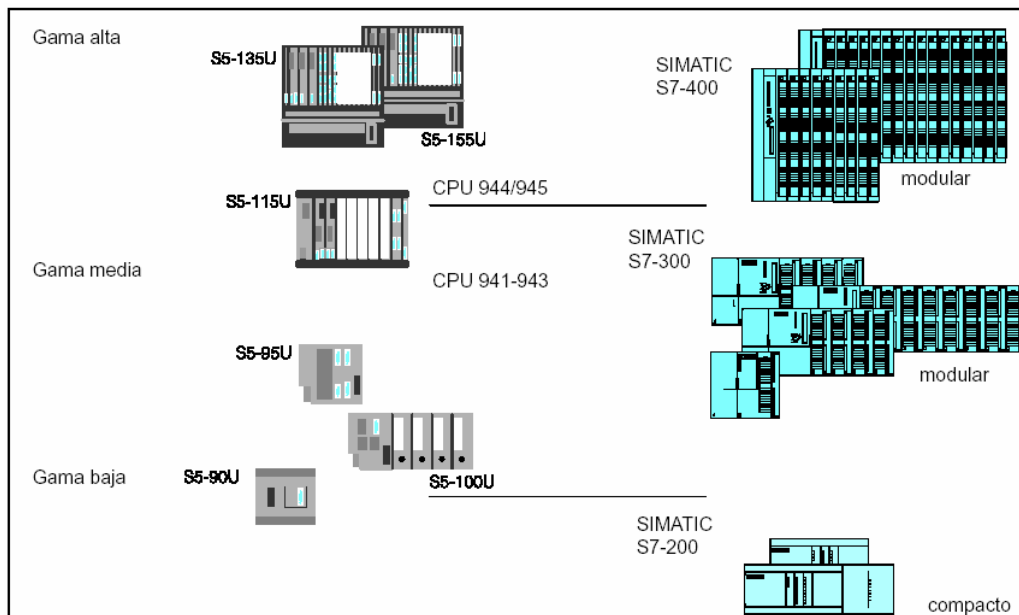


Figura 18: Gama de autómatas Simatic S7.

Características del lenguaje de programación Simatic [9]

Básicamente existen tres lenguajes denominados Lista de instrucciones “AWL”, Diagrama de escalera “KOP”, y por último diagrama lógico “FUP”.

Hay numerosas discusiones al respecto del mejor lenguaje para programa, los defensores del AWL, a menudo procedentes de lenguajes de programación estructurados, Los defensores del KOP, normalmente estructuran bien el programa, pero suelen terminar necesitando una “caja” para cada nueva función que se les plantea, proceden del aprendizaje de diseños cableados con contactores. Por último, los programadores en FUP tienen el inconveniente de que Step 7 aún no está orientado a este tipo de programación.

La programación de este PLC no obliga a utilizar un lenguaje común para todo el proyecto, siendo nuestra propia limitación a la hora de cambiar de idioma lo que nos obliga a obcecarnos en uno en particular. La decisión no es pues **qué** lenguaje de programación utilizar, sino **cuando** utilizar cada uno. Para conocer dicha respuesta, deberemos aprender primero las ventajas e inconvenientes de cada uno de ellos.

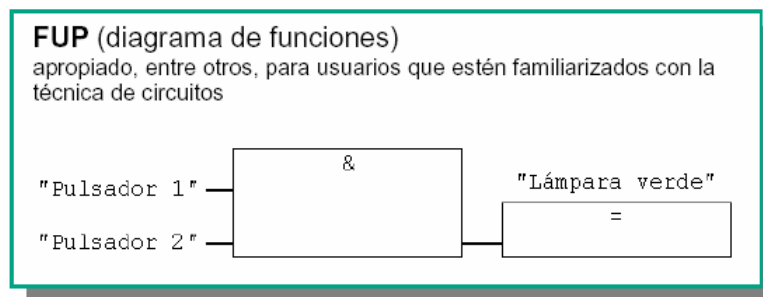
Veamos ahora las **ventajas del KOP**:

- Es muy sencillo ver en él las condiciones que no cumplen, y seguir el proceso.
- Totalmente indicado para programadores más cercanos al mundo eléctrico que al informático en tareas de tratamiento digital (bobinas, set, reset...)

Y los **inconvenientes del KOP**:

- Las series y paralelos requieren tanto espacio en pantalla que se salen de la misma, por lo que obliga a desplazar a menudo. La solución a dicho problema pasa por utilizar marcas.
- El realizar comparaciones de salto analógicas es misión casi imposible, a poco que se complique el tema.
- Y el principal problema: las cajas de KOP necesitan una sistemática de proceso por parte del Step 7 que hace que no se optimice el código de las mismas, por lo que el programa haciendo lo mismo va más lento.

Ventajas e inconvenientes del lenguaje FUL



Por último las **ventajas del FUP** son:

- Permite realizar gran cantidad de series y paralelos en la misma pantalla, con lo cual se acerca a la ventaja del AWL, pero con mayor claridad en el diagnóstico.
- Es el indicado para los programadores electrónicos, acostumbrados a la lógica digital cableada.

Y los inconvenientes del **FUP** son:

- No es útil, al igual que le pasaba al KOP, para tratar valores analógicos ni condiciones de salto.
- Sufre el mismo problema de optimización de código en el tratamiento que realiza del mismo el Step 7.
- Los programadores de KOP suelen no identificar de una manera rápida las combinaciones *and* y *or* en un status de programa.

Características del sistema modulas de la serie S7 300

La gama de módulos del sistema S7 sigue en la línea del ya conocido y acreditado SIMATIC S5. Esta gama se ha ampliando progresivamente.

Para el S7 se dispone de los tipos de módulos siguientes:

- módulos centrales (CPU),
- puentes de alimentación (PS),
- módulos interfase (IM),
- módulos de comunicación CP; (p. ej.: para PROFIBUS), módulos de función FM; (p. ej.: contaje, posicionamiento y regulación),
- los módulos digitales y analógicos se denominan de señal (SM).

Algunas de las características de los módulos STEP 7 se distinguen por sus nuevas prestaciones:

- Los módulos ya no disponen de puentes e interruptores.
- Todos los módulos funcionan sin ventilador y tienen el grado de protección IP 20.
- Existen módulos parametrizables y módulos con funciones de diagnóstico.
- La ocupación de los slots del S7 es más flexible.
- Los aparatos de ampliación y los sistemas de periferia descentralizada ET 200 pueden disparar alarmas.

Características de las CPU S7 300

Característica	312 IFM	313	314	314 IFM	315	315-2 DP
Memoria de trabajo integrada	6 kbytes	12 kbytes	24 kbytes	24 kbytes	48 kbytes	
Memoria de carga integrada	20 kbytes RAM; 20 kbytes EEPROM	20 kbytes RAM	40 kbytes RAM	40 kbytes RAM; 40 kbytes EEPROM	80 kbytes RAM	
• ampliable con Memory Card	~	hasta 512 kbytes	hasta 512 kbytes	~	hasta 512 kbytes (en la CPU hasta 256 kbytes programables)	
Tamaño de la imagen del proceso; E/S, respectivamente	32 bytes + 4 on-board	128 bytes	128 bytes	124 bytes + 4 on-board	128 bytes	
Área de direccionamiento de la periferia	Entradas: 128	128	512	Entradas: 496	1024	
• E/S digitales	+ 10 on-board Salidas: 128 + 6 on-board			+ 20 on-board Salidas: 496 + 16 on-board		
• E/S analógicas	32		64	Entradas: 64 + 4 on-board Salidas: 64 + 1 on-board	128	
Marcas	1024	2048				
Contadores	32	64				
Temporizadores	64	128				
Suma máxima de los datos remanentes	72 bytes		4736 bytes	144 bytes	4736 bytes	
Datos locales	512 bytes en total; 256 bytes por prioridad	1536 bytes en total; 256 bytes por prioridad				
Bloques:						
OBs	3	13	13	13	13	14
FBs	32	128	128	128	128	128
FCs	32	128	128	128	128	128
DBs	63	127	127	127	127	127
SFCs	25	44	48	48	48	53
SFBs	2	7	7	14	7	7

Requisitos del Software Simatic S7

El software de configuración y programación de SIMATIC S7 está diseñado según los criterios ergonómicos y es ampliamente auto-explicativo. Los requisitos básicos del sistema son SO Windows, PC con un procesador Pentium y una capacidad de memoria RAM de 128 MB

como mínimo, aunque se recomienda 256 MB, puerto RS232, monitor VGA, un teclado un ratón.

Memoria disponible en el disco duro:

- El paquete básico ocupa 105 MB en caso de instalarlo en un solo idioma, el requerimiento de memoria dependerá del tipo de instalación que se elija.
- STEP 7 debería disponer de unos 128 MB de la memoria principal para crear archivos de intercambio STEP 7.
- Se habrán de reservar aproximadamente 50 MB para los datos de usuario.
- Como mínimo se requiere 10 MB libre en la unidad C: para el programa de instalación Setup (los archivos del setup se borran al terminar la instalación).

El software S7

STEP 7 contiene un programa de instalación llamado Setup que ejecuta la instalación automáticamente. Las instrucciones que aparecen en pantalla le guiarán paso a paso a través de todo el proceso de instalación.

Una vez instalado el paquete y las licencias de uso correspondientes, la forma más rápida de arrancar STEP 7 es haciendo un doble clic en el icono "**Administrador SIMATIC**". Seguidamente se abre la ventana del Administrador SIMATIC, desde donde se accede a todas las funciones que se hayan instalado, es decir, tanto a las del paquete básico como a las del software opcional.

El Administrador SIMATIC es el interface de acceso a las funciones de configuración y programación, estas permiten:

- crear proyectos
- configurar y parametrizar el hardware

- configurar enlaces de comunicación
- crear programas
- comprobar los programas creados y ponerlos en servicio

El acceso a las funciones, que se basa en la programación orientada a objetos, es intuitivo y fácil de aprender. El Administrador SIMATIC permite operar offline, es decir, sin tener conectado el autómatas programable y online, es decir, con el autómatas programable conectado.

Filosofía del uso de proyectos en el Administrador Simatic

El procedimiento fundamental consiste en la filosofía de uso de proyecto. Los proyectos representan la totalidad de datos y programas de una solución de automatización. Sirven para almacenar de forma ordenada todos los datos y programas requeridos para elaborar una solución de automatización determinada.

Un proyecto de STEP 7 abarca toda la gestión de programas y datos de una solución de automatización, independientemente del número de módulos centrales utilizados y de cómo estén interconectados. Por consiguiente, el proyecto no se limita solamente a un programa de usuario destinado a un módulo programable, sino que puede englobar varios programas de usuario para varios módulos programables que se encuentren bajo un mismo nombre de proyecto.

Por supuesto, también es posible crear un programa de usuario sencillo para un solo módulo programable. En este caso el proyecto se limitará a un módulo central.

El árbol de directorios que ofrece STEP 7 para los programas de usuario y los datos en su proyecto es el siguiente:

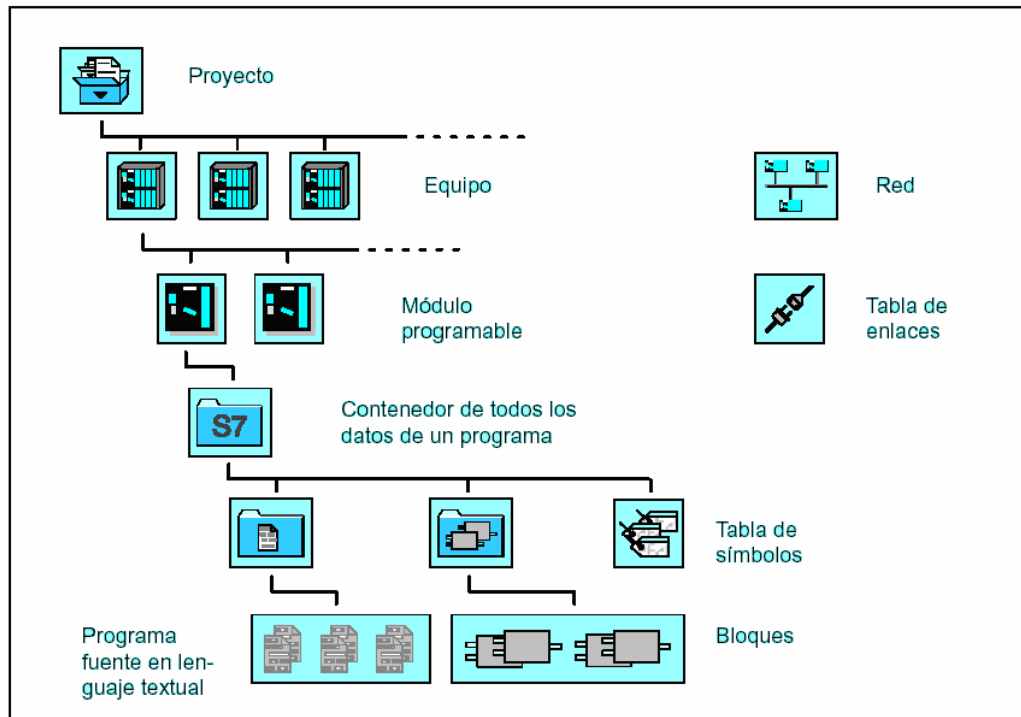


Figura 19: Árbol de directorios de un proyecto en el Administrador Simatic.

La organización del programa en bloques

La organización del código fuente de programa se estructura en bloques funcionales de varios tipos:

Bloques de organización (OB)	Los bloques de organización (OBs) constituyen el interface entre el sistema operativo y el programa de usuario. Los distintos bloques de organización se reparten las diferentes tareas.
Función (FC)	Las funciones (FC) son bloques lógicos “ sin memoria ”. Los parámetros de salida contienen los valores que se obtienen tras ejecutar la FC. El uso posterior y la memorización de los parámetros actuales después de llamar a una FC está en manos del usuario.
Bloque de función (FB)	Un bloque de función (FB) es un bloque lógico “ con memoria ”. Como memoria se utiliza un bloque de datos de instancia que sirve para almacenar los parámetros actuales y los datos estáticos de bloques de función. Los bloques de función se aplican, p. ej., en la programación de reguladores.

<p>Funciones de sistema (SFC) Bloques de función de sistema (SFB)</p>	<p>Las funciones de sistema son los bloques de organización especiales que se pueden llamar desde el programa de usuario. No todas las funciones tienen que ser programadas por el usuario. Existen bloques preconfeccionados que residen en el sistema operativo de los módulos centrales, por ejemplo para programar funciones de comunicación. Las funciones de sistema (SFC) tienen las características de una función (FC) y los bloques de función de sistema (SFB) con las características de un bloque de función (FB).</p>
<p>Bloque de datos (DB)</p>	<p>Los bloques de datos en el S7-300 pueden tener una longitud de hasta 8 kbytes, en el S7-400 de hasta 64 kbytes). Los bloques de datos guardan los datos del programa de usuario. Cabe distinguir dos tipos de bloques de datos diferentes:</p> <ul style="list-style-type: none"> • Los bloques de datos globales no están asignados a ningún bloque. • Los bloques de datos de instancia están asignados a un bloque de función. <p>Todo bloque de datos puede ser un bloque de datos global o un bloque de datos de instancia.</p>
<p>Bloques de datos de sistema (SDB) (Parametrización de la CPU)</p>	<p>Además de los programas o los datos del programa de usuario, existen otros que contienen los ajustes, como pueden ser los parámetros de módulos o también direcciones. Estos bloques se denominan bloques de datos de sistema (SDB). Los bloques de datos de sistema se crean con herramientas especiales de STEP 7, p. ej., al introducir los datos de configuración del hardware o al elaborar tablas de enlaces.</p>

Clasificación de los bloques de organización (OB)

Función	Descripción	Bloque
Programa principal	Ciclo libre	OB 1
Alarmas	Alarma de retardo	OB 20 a OB 23
	Alarma horaria	OB 10 a OB 17
	Interrupciones de hardware	OB 40 a OB 47
	Alarmas de proceso	interrupciones
	Alarmas cíclicas	OB 30 a OB 38
	Alarma de multiprocesamiento	OB60
Arranque	Nuevo arranque manual	OB 100
	Rearranque manual	OB 101
	Rearranque automático	OB 101
Error	Error	OB 121, OB 122, OB 80 a OB 87
Otras	Tarea no prioritaria	OB90

Tratamiento de errores con bloques de organización (OB)

Función	Bloque
Llamada de un bloque no cargado	OB 121
Retardo en acuse de recibo al acceder directamente a los módulos de la periferia	OB 122
Retardo en acuse al actualizar la imagen del proceso y las marcas de acoplamiento	OB 122
Errores de direccionamiento	OB 122
Tiempo de ciclo excedido	OB80
Retardo en acuse de recibo en el byte de entrada EB 0	OB85
Código de operación no permitido	STOP
Retardo en acuse al acceder directamente a la periferia en el área de direccionamiento ampliada	OB 122
Error de paridad o retardo en acuse de recibo al acceder a la memoria de usuario	OB 122
Error de transferencia de un bloque de datos	OB 121
Colisión de alarmas temporizadas	OB80
Error al crear un bloque de datos	aviso de la SFC
Error de interface	OB84

Programas de una CPU S7-300 [8]

En la CPU se ejecutan principalmente dos programas diferentes: el sistema operativo y el programa de usuario.

Sistema operativo

El sistema operativo, que está integrado en la CPU, organiza todas las funciones y procesos de la CPU que no están ligados a una tarea de control específica. Sus funciones son:

- gestionar el arranque en caliente y el arranque normal
- actualizar la imagen de proceso de las entradas y emitir la imagen de proceso de las salidas
- llamar a los programas de usuario
- detectar las alarmas y llamar los OBs de tratamiento de alarmas
- detectar y tratar los errores
- administrar las áreas de memoria
- comunicar con unidades, dispositivos de programación y otras estaciones de comunicación

Modificando los parámetros del sistema operativo (preajustes) se puede controlar el comportamiento de la CPU en tareas determinadas.

Programa de usuario

El programa de usuario primero se ha de crear y luego se ha de cargar en la CPU. Contiene todas las funciones requeridas para procesar la tarea específica de automatización. Las tareas del programa de usuario son:

- definir las condiciones del arranque en caliente y del arranque normal

- Preestablecer un valor determinado para las señales.
- tratar datos del proceso, ej. efectuar combinaciones lógicas de señales binarias
- leer y evaluar valores analógicos, definir señales binarias de salida, emitir valores analógicos
- reaccionar ante las alarmas
- tratamiento de perturbaciones en el desarrollo normal del programa.

Bloques del programa de usuario

El software de programación STEP 7 permite estructurar el programa de usuario, es decir, subdividirlo en distintas partes. Esto aporta las siguientes ventajas:

- los programas de gran tamaño se pueden estructurar y programar de forma clara y legible
- se pueden estandarizar determinadas partes del programa
- se simplifica la organización del programa
- las modificaciones del programa pueden realizarse con mayor facilidad
- se simplifica el test del programa, ya que puede ejecutarse por partes
- se simplifica la puesta en marcha.

Pasos para la creación de un proyecto en Simatic S7

Para crear un proyecto con step 7 el procedimiento es el siguiente:

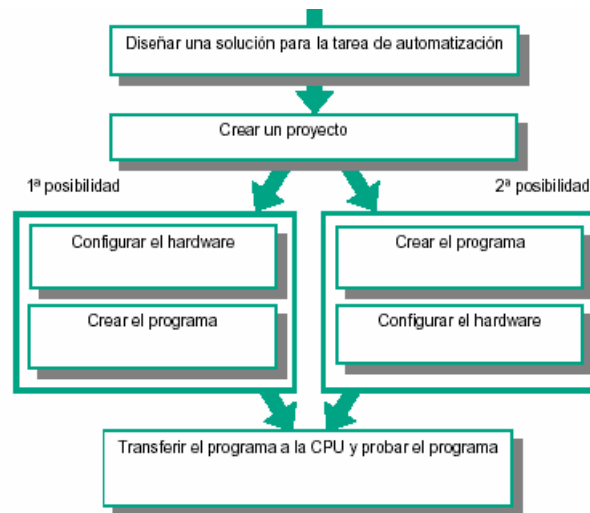


Figura 20: Proceso de creación de un proyecto.

1º Diseño de una solución de automatización: supongamos que la primera parte de nuestro proyecto consiste realizar el ajuste y la monitorización de todas las variables analógicas y los actuadores del “control de nivel en bucle cerrado”.

Inicialmente definimos las variables de entrada y le asignamos una posición de memoria en el autómata, que para mayor comodidad posterior para la monitorización, lo haremos en un bloque de datos global, denominado **DB10**, Los bloques de datos en el S7-300 pueden tener una longitud de hasta 8 kbytes, y sirven para guardar los datos del programa de usuario.

Descripción de la variable	Denominación	Valor inicial
Consigna de nivel	NIVEL_CONSIGNA	150
Lectura de la entrada analógica del sensor ultrasónico de nivel	NIVEL_PEW	
Modo de trabajo de la bomba (100% o regulable)	BOMBA_MODO	TRUE
Regulación del grupo motor bomba	BOMBA_REG	100
Electro-válvula para descarga	EV_102	FALSE

Las entradas y salidas del sistema son las siguientes:

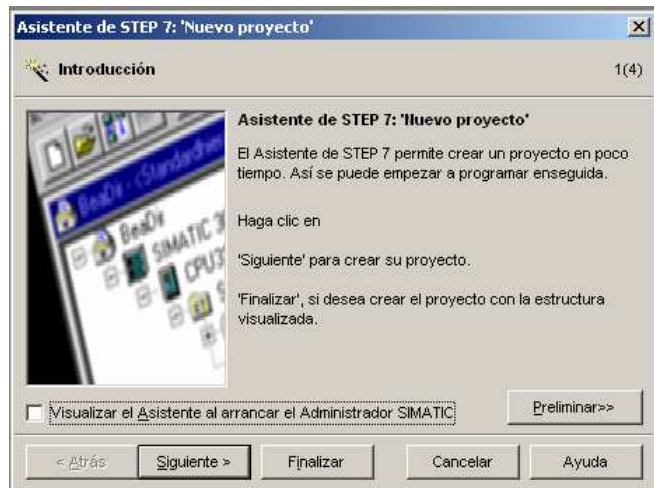
Descripción de la variable	Posición	Valor inicial
Entrada analógica para la lectura del sensor ultrasónico de nivel.	PEW752	
Salida analógica para la regulación del grupo motor bomba.	PAW752	100
Salida Digital para el control de modo de funcionamiento del grupo motor bomba.	A124.0	TRUE
Salida Digital para el control de la electroválvula EV102 de descarga rápida del depósito.	A124.1	FALSE

Descripción de proceso: El proceso consiste en establecer una consigna en la variable CONSIGNA_NIVEL, y que el sistema sea capaz de ajustarse con un control en bucle cerrado exclusivamente proporcional.

2º Creación del proyecto en S7: Abrimos el administrador Simatic con el icono que se encuentra en el escritorio tras la instalación del paquete de software.



Seleccionamos en el menú superior “Asistente -> Nuevo Proyecto” y arrancamos el asistente que nos guiará en la creación del nuevo proyecto.



El Asistente de STEP 7 crea un proyecto STEP

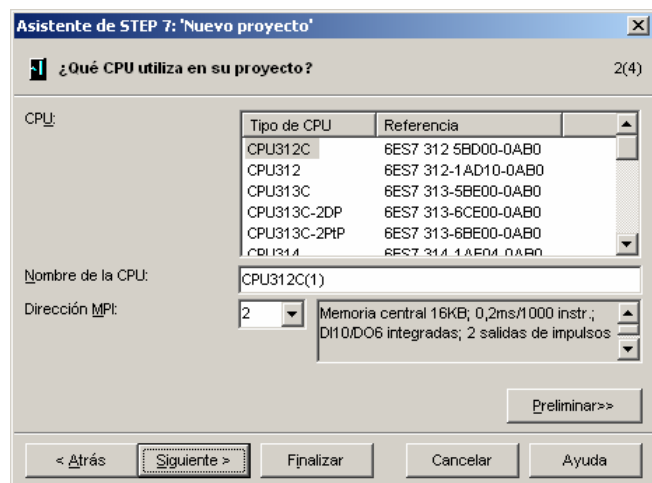
7 de forma rápida para poder empezar enseguida con la programación.

Con ayuda del Asistente de STEP 7 se selecciona qué (CPU) desea utilizar, qué bloques de organización debe contener su programa para esta CPU y cómo se debe llamar el nuevo proyecto.

En la presentación preliminar compruebe la estructura de su proyecto y corrija la selección de la CPU, la de los bloques de organización y la del nombre del proyecto, si es necesario.

Visualizar el Asistente al arrancar el Administrador SIMATIC: Mientras esta opción esté activada, cada vez que se arranque el Administrador SIMATIC también se arrancará el Asistente. Para activar y desactivar dicha opción, haga clic en la casilla de verificación.

3º Configuración del hardware: tipo de CPU: En la lista se visualizada la CPU que se pueden utilizar en la solución de automatización. Para ello seleccionamos la CPU 313C.

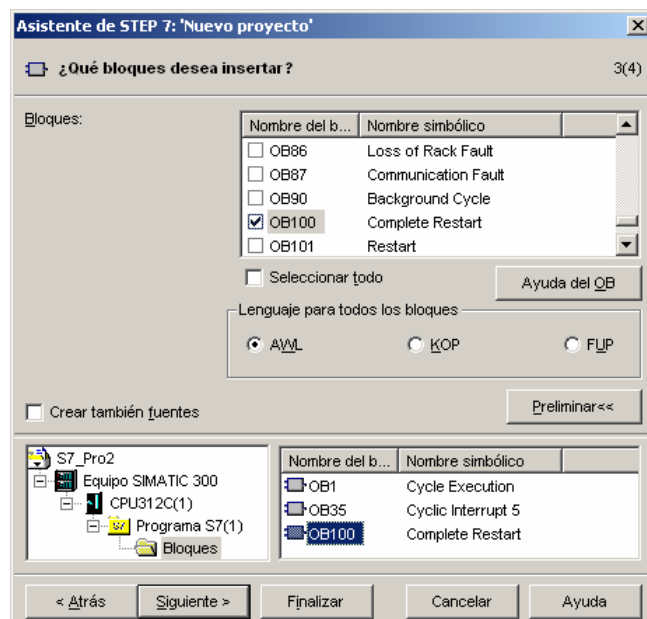


A continuación se visualiza una breve descripción del módulo seleccionado. Esto facilita la selección, puesto que - por razones de espacio - en la lista sólo se incluyen los tipos de CPU y los números de fábrica correspondientes.

Nombre de la CPU: Con este nombre se representa la CPU en STEP 7. El tipo de CPU seleccionado será el nombre preajustado. Si es necesario, se puede reemplazar este nombre por otro que se ajuste más a su significado.

Dirección MPI: Normalmente no se necesitará modificar la dirección preajustada. Si se conectan varias CPUs a una red vía el bus MPI, cada una de ellas necesitará una dirección MPI unívoca. Por ello, si desea conectar CPUs a una red, puede ajustar aquí la dirección del módulo de la CPU en el bus MPI.

Bloques a insertar: Los bloques son partes del programa de usuario que se diferencian por su función, estructura o finalidad de aplicación. Los bloques especiales, bloques de organización (OBs), forman el interface entre el sistema operativo de la CPU S7 y el programa de usuario.



En la lista visualizada se seleccionan los bloques que el Asistente de STEP 7 deba integrar en su programa de usuario. Para ello se hace clic en la casilla de verificación situada delante del nombre del bloque.

Seleccionar todo: Al seleccionar esta opción, se seleccionan automáticamente todos los bloques contenidos en la lista para integrarlos en el programa de usuario. Si vuelve a hacer clic en la casilla de verificación, se cancelará la selección de los bloques.

Lenguaje para los bloques seleccionados: Elija aquí el lenguaje de programación para los bloques seleccionados:

- KOP: esquema de contactos

- FUP: diagrama de funciones

- AWL: lista de instrucciones

En nuestro caso seleccionaremos únicamente: OB1, OB100, OB35, que corresponde a:

OB100 Bloque que se ejecutará sólo una vez después de un reset.

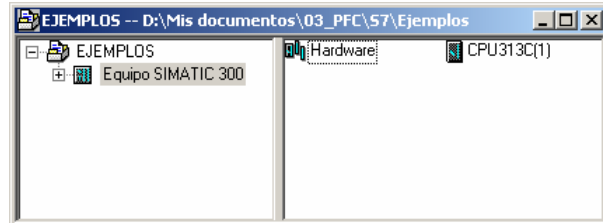
OB1 Bloque de ejecución cíclica por cada ciclo de scan.

OB35 Bloque de ejecución por interrupción cíclica.

Una vez finalizado la creación del proyecto este se encuentra localizado por defecto en el directorio configurado para guardar los proyectos, que es: "C:\Archivos de programa\Siemens\Step7\s7proj\"

El hardware básico, queda configurado con el asistente, pero normalmente hay que ajustarlo, el la versión de la CPU, la fuente de alimentación y otras tarjetas opcionales que pueden estar montadas en la aplicación.

El administrador Simatic mostrará dentro de la ventana del proyecto que hemos creado con el asistente, un equipo que genéricamente se llama "Equipo SIMATIC 300" y que está formado por dos objetos: "Hardware" desde el que se accede a la aplicación de configuración de las opciones hardware y el objeto "CPU313C(1)" desde el que se accederá a los bloques funcionales de programa.



Las modificaciones del hardware se hacen con la aplicación HW-Config incluida en el administrador Simatic, en general el Procedimiento para configurar y parametrizar es el siguiente, se tiene que haber abierto o creado un proyecto en el Administrador SIMATIC.

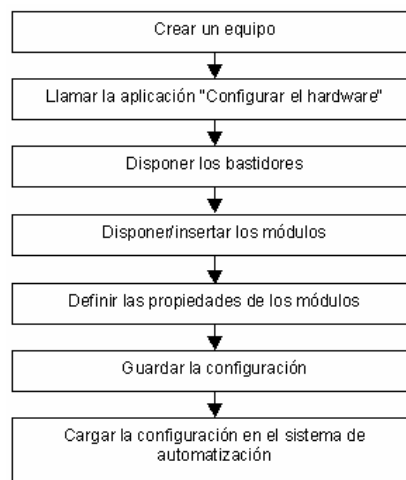


Figura 21: Parametrización de un proyecto.

Ventana de configuración: Para configurar un sistema de automatización se dispone de dos ventanas:

- la **ventana del equipo** en la que se emplazan los bastidores.
- la ventana "**Catálogo de hardware**" de la que se seleccionan los componentes de hardware requeridos, p. ej. bastidor, módulos y módulos.

Ventana del equipo:

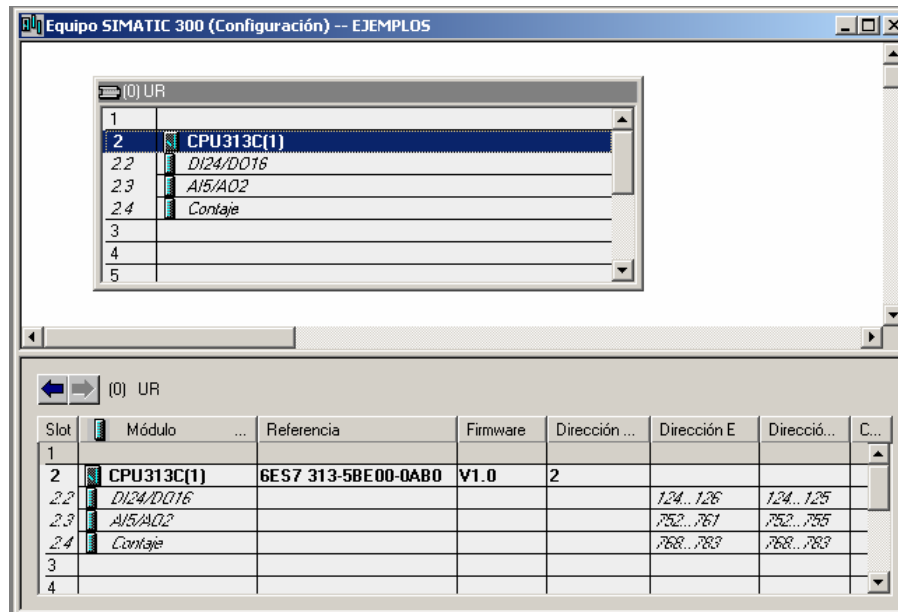
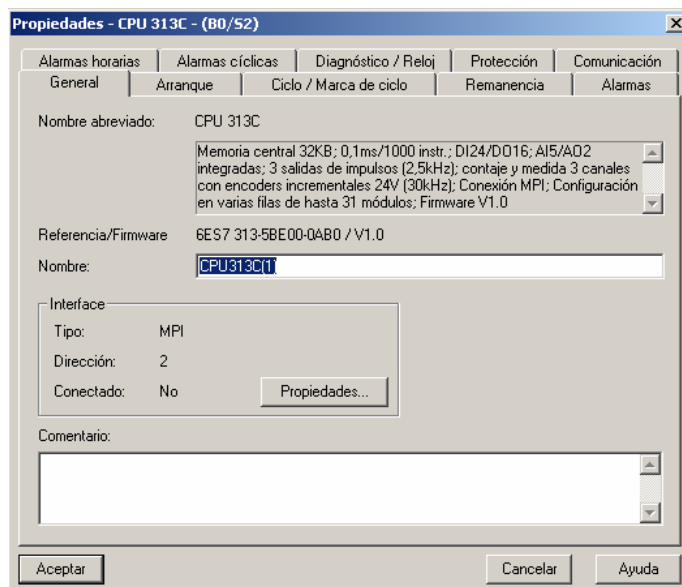


Figura 22: Ventana de parametrización de un proyecto.

Para ver y modificar las propiedades de cualquiera de los elementos basta con pulsar con el ratón sobre el elemento en cuestión montado en nuestro bastidor, en la imagen de la derecha puede verse la configuración y las propiedades de la CPU seleccionada.



Configurar: Por "configurar" se entiende en STEP 7 la disposición de los bastidores, de los módulos, de los aparatos de la periferia descentralizada y de los módulos adicionales. Los bastidores se representan en una tabla de configuración que puede acoger un número determinado de módulos adicionales, del mismo modo que los bastidores "reales".

En la tabla de configuración, STEP 7 asigna automáticamente una dirección a cada módulo. Si la CPU se puede direccionar libremente, es posible modificar las direcciones de los módulos de un equipo.

La configuración se puede copiar cuantas veces se desee a otros proyectos de STEP 7. Si es necesario, también se puede modificar y cargar en una o varias instalaciones existentes. Durante el arranque del sistema de automatización, la CPU compara la configuración teórica creada en STEP7 con la configuración física (real) de la instalación. Así es posible detectar e indicar inmediatamente los posibles errores.

Parametrizar: Por "parametrizar" se entiende en STEP 7:

- ajustar las propiedades de los módulos parametrizables para la configuración centralizada y para una red. Ejemplo: una CPU es un módulo parametrizable. El tiempo de vigilancia de ciclo es un parámetro ajustable;
- ajustar los parámetros de bus, así como los del maestro y de los esclavos para un sistema maestro (PROFIBUS) u otros ajustes para el intercambio de datos entre componentes.

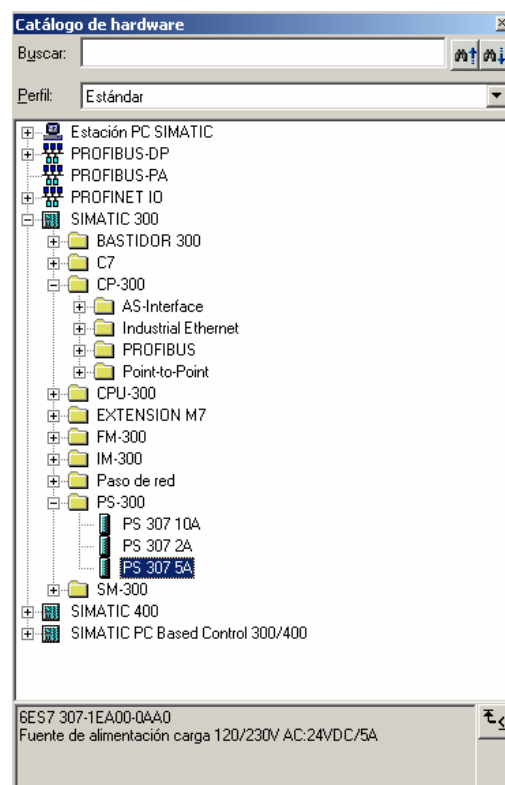
Los parámetros se cargan en la CPU, la cual los transfiere en el arranque a los módulos en cuestión. Los módulos se pueden sustituir muy fácilmente, puesto que los parámetros creados en STEP7 se cargan automáticamente en el nuevo módulo durante el arranque.

Las propiedades de los sistemas de automatización S7 y de los módulos están preajustadas de tal forma que normalmente el usuario no necesita configurar. Es indispensable configurar:

- para cambiar los parámetros predeterminados de un módulo (p. ej., habilitar la alarma de proceso en un módulo)
- para configurar enlaces de comunicación
- en el caso de utilizar equipos con periferia descentralizada (PROFIBUSDP o PROFINET IO),
- en el caso de utilizar equipos S7-400 con varias CPUs (multiprocesamiento) o bastidores de ampliación.
- en el caso de utilizar sistemas de automatización de alta disponibilidad.

En nuestro caso será necesario añadir dos elementos a nuestro bastidor estos podemos encontrarlos en el “Catálogo de Hardware”:

- Una fuente de alimentación con denominada: “PS-307-5A”
- Un módulo Ethernet de comunicaciones denominado “CP 343-1 Advanced”.



Visualizar el catálogo de hardware: Si no se visualiza la ventana "Catálogo de hardware", en la parte derecha se elija el comando **Ver > Catálogo**.

El contenido del catálogo puede actualizarse con los nuevos componentes.

A nivel de proyecto es necesario configurar la red para poder acceder al PLC para programarlo, monitorizar variables del proceso etc... Esto se hace con la utilidad NetPro a la que se puede acceder desde el administrador simatic a través del menú “Herramientas -> Configurar red”. En nuestro caso añadimos dos subredes “MPI” y “Ethernet”, para configurar el acceso a cada una de ellas pulsaremos sobre el dispositivo de conexión localizado en la CPU o sobre el enlace que lo une a la red.

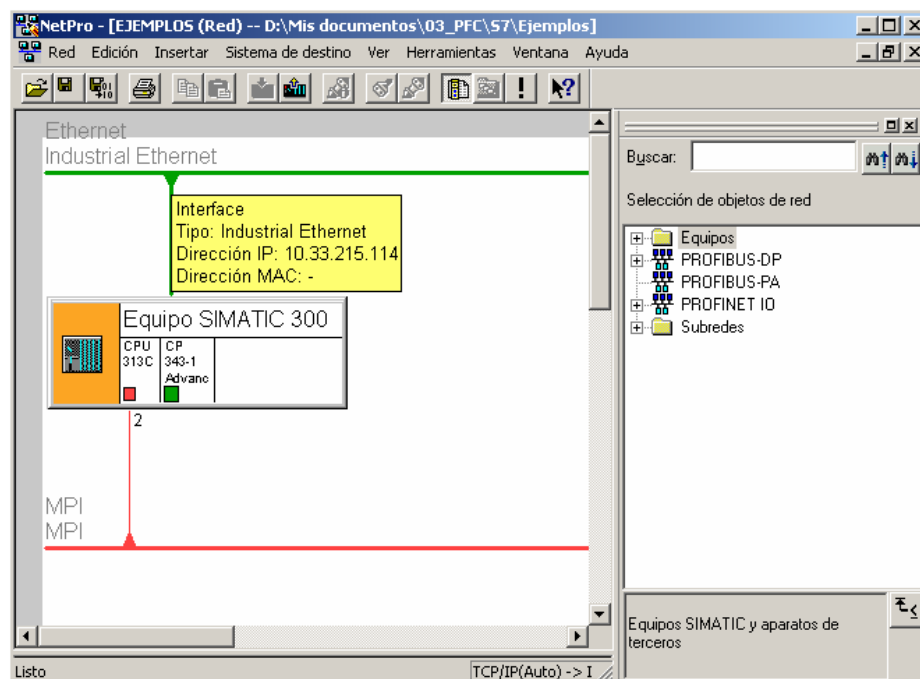
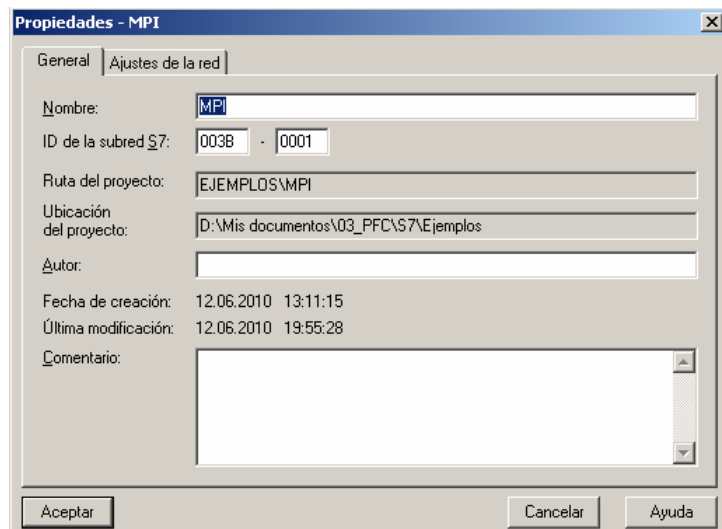


Figura 23: Aplicación NetPro para configurar las redes de comunicación.

Subred MPI: En esta ficha se puede cambiar la dirección de la estación y elegir la subred MPI a la que está conectada. Las direcciones MPI

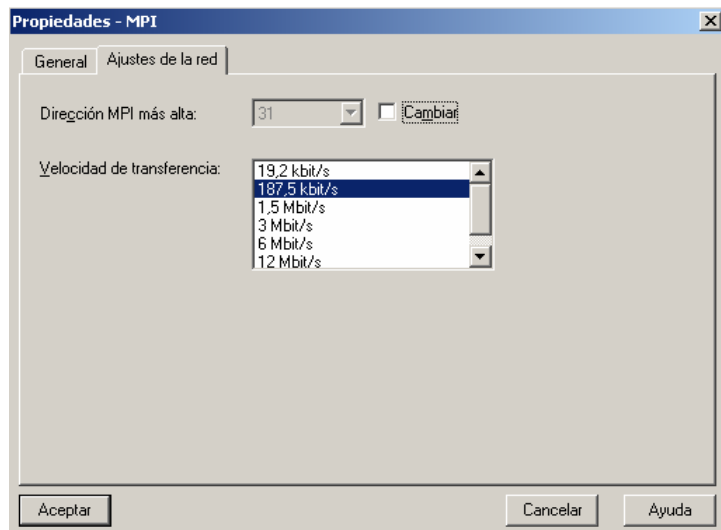


no deben ser asignadas más de una vez. Además de visualiza la velocidad de transferencia ajustada para la subred a la que está conectado el módulo. Si está seleccionada la opción "no conectado" se indica la velocidad de transferencia preajustada para la estación. Se pueden tener varias subredes MPI y elegir la subred a la que desea conectar el interface MPI de la estación.

El ID de la subred S7 se compone de dos números separados por un guión, un número para el proyecto y un número para la subred, En caso de que quiera pasar al modo online con una PG sin disponer de un proyecto coherente, tiene que conocer el ID de la subred S7.

También es posible configurar otras opciones de la subred MPI como la ruta y la ubicación del proyecto, el autor, las fechas de creación y la última modificación, así como un campo de comentarios para explicar brevemente la tarea que tiene que llevar a cabo la subred.

En STEP 7 se ajustan automáticamente las propiedades de la subred como por ejemplo la velocidad de transferencia para todas las estaciones de una subred. Si se modifican las



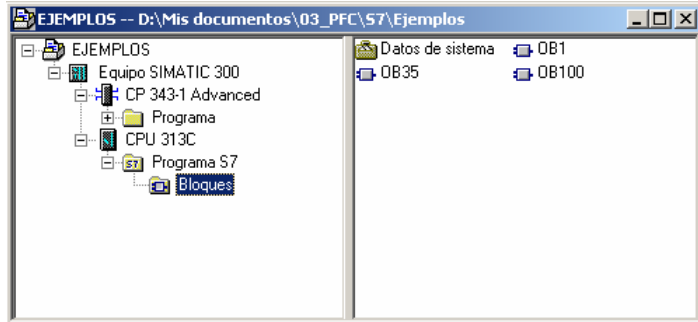
propiedades de la subred con STEP 7 hay que asegurarse de que los ajustes tengan efecto en cada estación de la subred que forman el proyecto.

La "Dirección MPI más alta", permite optimizar la subred MPI. El ajuste de la velocidad de transferencia depende de las propiedades de las

estaciones MPI utilizadas. Dicha velocidad no puede ser superior a la de la estación más lenta.

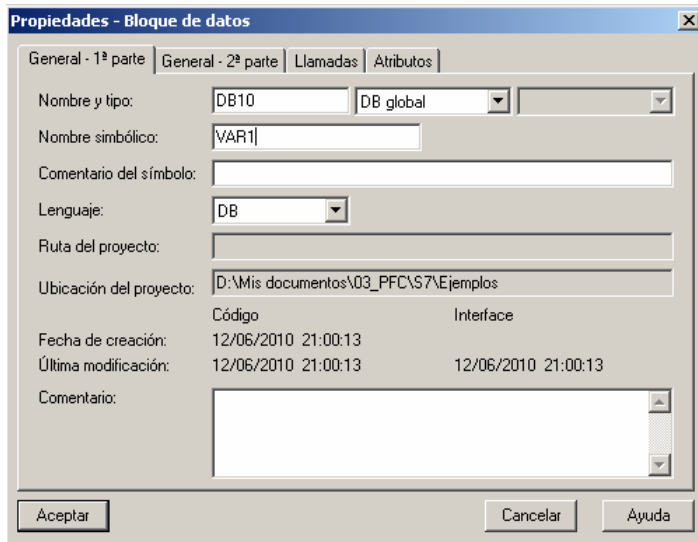
4º Creación del programa:

Una vez configurado el hardware y las subredes de comunicación el proyecto tiene un aspecto similar al de la imagen.



En el objeto CPU 313C hay una carpeta denominada Programa S7 / Bloques, donde se encuentran los tres bloques que insertamos en la creación del proyecto, además será necesario incluir algún bloque de datos DB.

Comencemos por la creación del bloque de datos DB10, pulsando con el botón derecho sobre la carpeta de bloques “Insertar nuevo objeto -> bloque de datos”



Esta ficha permite ver o introducir las propiedades generales del bloque a crear, nombre, tipo, nombre simbólico, comentario, lenguaje, etc..

Nombre del bloque, consistente en el tipo de bloque y su número, **DB10**.

Los bloque de datos podrá ser "DB global", "DB de instancia". Si la carpeta de bloques contiene FBs o SFBs, podrá seleccionar de esta lista desplegable el tipo "DB de instalcia" y asignarle un FB o SFB. Al generar un bloque de datos de instancia se toman las propiedades "Autor" y "Familia" del FB o del SFB asignado.

Además existen otras pestañas en la ficha de propiedades del bloque denominadas "General – 1º parte", "General – 2º parte", "Llamadas" y "Atributos", donde podemos configurar entre otras opciones: autor, familia, versión, tamaño,

El DB está protegido contra escritura en el PLC: para los bloques de datos es posible establecer protección contra escritura, de tal forma que durante la ejecución del programa no sea posible sobrescribir éstos.

Protección KNOW HOW: un bloque compilado con esta opción no permite examinar la parte de las instrucciones. La interface del bloque puede visualizarse, pero no puede modificarse.

Bloque estándar: el bloque actual es un bloque estándar de SIEMENS. Dotado de protección KNOW HOW, los campos de entrada para Nombre, Familia, Versión y Autor aparecen en gris y no pueden ser editados.

Unlinked: un bloque de datos con la propiedad "UNLINKED" sólo se guarda en la memoria de carga. El bloque no ocupa lugar en la memoria de trabajo y no se integra en el programa. A este tipo de bloques no puede accederse con comandos MC7. Dependiendo de la CPU en cuestión, los contenidos de un DB de este tipo sólo pueden transferirse a la memoria de trabajo con SFC 20 BLKMOV o SFC 83 READ_DBL.

Los datos que introducimos en el DB10 son los siguientes:

Dirección	Nombre	Tipo	Valor inicial	Comentario
0.0		STRUCT		
+0.0	NIVEL_CONSIGNA	REAL	1.500000e+002	VALOR INTRODUCIDO COMO CONSIGNA DE NIVEL (mm)
+4.0	NIVEL_VALOR	REAL	0.000000e+000	NIVEL DEPOSITO SALIDA DE LA FUNCIÓN ESCALA (mm)
+8.0	NIVEL_MAX	REAL	2.950000e+002	LECTURA MAXIMA
+12.0	NIVEL_MIN	REAL	5.000000e+001	LECTURA MINIMA
+16.0	NIVEL_ERR	WORD	W#16#0	ERROR LECTURA NIVEL
+18.0	NIVEL_PEW	WORD	W#16#0	LECTURA PERIFERIA
+20.0	BOMBA_MODO	BOOL	TRUE	MODO 1=100% 0=REGULABLE A124.2
+22.0	BOMBA_REG	REAL	8.000000e+001	REGULACION BOMBA 0..100
+26.0	EV_102	BOOL	TRUE	ELECTROVALVULA DESCARGA RÁPIDA DEPÓSITO A124.0
=28.0		END_STRUCT		

Figura 24: Contenido de un bloque de datos DB global.

El código de programa que realiza el lazo de control lo introduciremos en el bloque de interrupción cíclica OB35, ajustándolo a un tiempo de 1000 ms, el código estará formado por dos segmentos, el primero encargado de leer el valor de la entrada analógica del sensor ultrasónico y escalar su valor en milímetros.

Segm. 1: SENSOR DE NIVEL

LEER SENSOR DE NIVEL Y ESCALAR

```

CALL "ESCALA_PEW"
IN      :=PEW752
HI_LIM :="DATOS".NIVEL_MAX
LO_LIM :="DATOS".NIVEL_MIN
BIPOLAR:=FALSE
RET_VAL:="DATOS".NIVEL_ERR
OUT     :="DATOS".NIVEL_VALOR

L      PEW 752
T      "DATOS".NIVEL_PEW

L      "DATOS".NIVEL_ERR
L      0
==I
SPB   ET01
L      0
T      "DATOS".NIVEL_VALOR
S      "NIVEL_ALARMA"
ET01: NOP 0

```

El segundo segmento es el encargado de leer el valor de consigna y compararlo con el valor del sensor, actuando sobre el regulador. Todo este proceso lo realiza una función de la librería "CONT_C" (FB41) que tiene asociada un bloque de datos de instancia "PID" (DB41). Mas adelante se dedicará un apartado exclusivamente a explicar el funcionamiento de las funciones de control PID y auto-sintonización,

ahora simplemente la usaremos utilizando los parámetros mínimos necesarios, que son:

NIVEL_CONSIGNA: donde se encuentra el valor de nivel líquido deseado por el usuario.

NIVEL_VALOR: es el valor actual de nivel de líquido escalado en mm que está obteniendo el sensor ultrasonico.

PAW752: es la salida directa de la periferia analógica que actúa sobre en controlador PWM del motor, regulando su potencia.

RESET_PID: para rearrancar el PID, estará siempre a cero para evitar su re arranque.

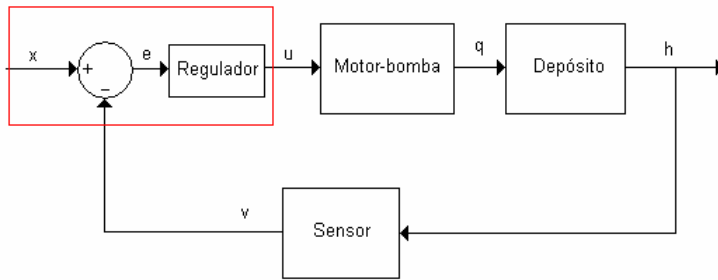
MAN_PID: permite poner el regulador en modo manual, estará siempre a cero para simplificar el programa.

Segm. 2: REGULADOR

REGULACION PID SOBRE MOTOR-BOMBA PARA REGULAR NIVEL DEL DEPOSITO SEGUN CONSIGNA ESTABLECIDA

```
CALL "CONT_C" , "PID"  
COM_RST := "RESET_PID"  
MAN_ON := "MAN_PID"  
PVPER_ON:=  
P_SEL :=  
I_SEL :=FALSE  
INT_HOLD:=  
I_ITL_ON:=  
D_SEL :=  
CYCLE :=  
SP_INT := "DATOS".NIVEL_CONSIGNA  
PV_IN := "DATOS".NIVEL_VALOR  
PV_PER :=  
MAN :=  
GAIN :=  
TI :=  
TD :=  
TH_LAG :=  
DEADE_W :=  
LMN_HLM :=  
LMN_LLM :=  
PV_FAC :=  
PV_OFF :=  
LMN_FAC :=  
LMN_OFF :=  
I_ITLVAL:=  
DISV :=  
LMN :=  
LMN_PER :=PAW752  
QLMN_HLM:=  
QLMN_LLM:=  
LMN_P :=  
LMN_I :=  
LMN_D :=  
PV :=  
ER :=
```

El diagrama de bloques del sistema es el siguiente:



Para monitorizar el funcionamiento del programa se ha incluido un bloque “tabla de variables” denominado “VAT_1” que permite monitorizar en tiempo real el contenido de las variables definidas:

	Operando	Símbolo	Forma	Valor de estado	Valor de forzado
1	DB10.DBD 0	"DATOS".NIVEL_CONSIGNA	REAL	150.0	
2	DB10.DBW 18	"DATOS".NIVEL_PEW	DEC	10480	
3	DB10.DBD 4	"DATOS".NIVEL_VALOR	REAL	142.8675	
4	M 20.0	"RESET_PID"	BOOL	false	
5	M 20.1	"MAN_PID"	BOOL	false	
6	DB41.DBD 72	"PID".LMN	REAL	14.26505	
7	DB41.DBX 0.3	"PID".P_SEL	BOOL	true	
8	DB41.DBX 0.4	"PID".I_SEL	BOOL	false	
9					

4.- Calibrado de sensores

Todos los dispositivos analógicos de uso general, actuadores o sensores están diseñados para trabajar en un rango amplio de la magnitud a medir y requieren de un ajuste previo que regulen los valores de offset y fondo de escala, para adaptar el dispositivo al entorno de trabajo. Que debe ser una magnitud incluida en el rango de medida del sensor. Normalmente las entradas o salidas de los sensores o actuadores suele ser una señal normalizada de tensión o corriente.

Señales Normalizadas

Las señales eléctricas analógicas, que se utilizan en los sistemas de control están normalizadas para conseguir los siguientes objetivos:

1. Estandarizar el conexionado de dispositivos.
2. Amplificación de las señales de los sensores para separar el ruido sobre las líneas eléctricas industriales.
3. Según criterios de diseño para la reducción de la fuente de error según la circuitería adoptada.

Los acondicionadores obtienen la señal de los transductores de magnitud, que son los encargados de convertirla en una señal manipulable, normalmente eléctrica, para la visualización de la magnitud o para la realización de operaciones de control y supervisión.

La señal más frecuentemente utilizada es la eléctrica con la finalidad de ser usada en dispositivos electrónicos.

- - Señales de tensión (CC):
 - 0-5, 0-10 V, +/-10 V, 0-24 V ó 0-32 V;
- - Señales de corriente (CC):
 - 0-50 mA, 0-20 mA ó 4-20 mA;
- - Pulsos (On-Off CC):
 - 5 V, 0-10 V ó 0-24 V.

Sensor de Nivel

El sensor de nivel situado en la parte superior del depósito B101, es un sensor analógico de ultrasonidos, modelo Berosonar 3RG6232-3A-3LS-M18 de la marca Siemens, con un rango de medida mínimo de 50 mm y máximo ajustable hasta 300 mm.

Dispone de una señal de sincronización, Mientras no está aplicada la señal de desbloqueo (XI), permanece memorizado el último estado de conmutación. Cuando se aplica nuevamente la señal de desbloqueo, la salida se actualiza. Uniendo los pins "XI" de hasta un máximo de 10 sensores BERO, se puede sincronizar la lectura.

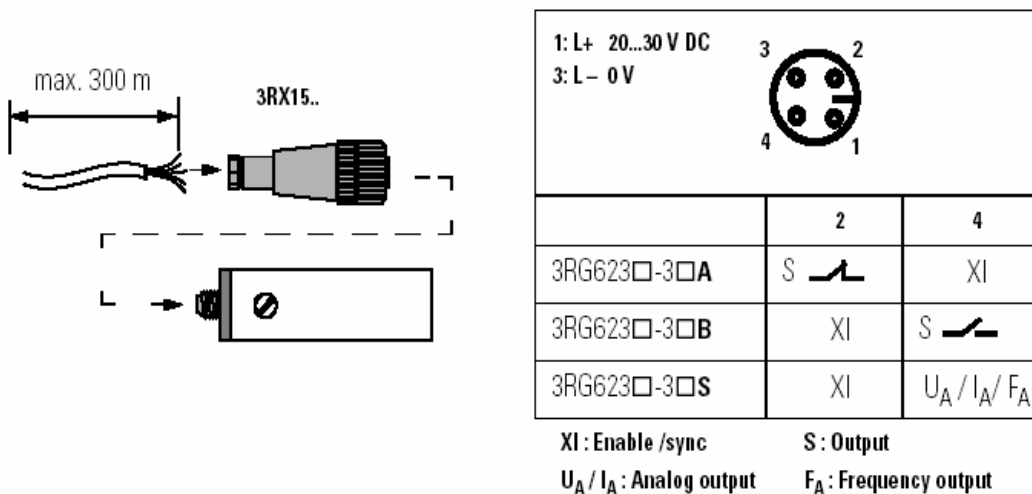


Figura 25: Características del sensor de nivel.

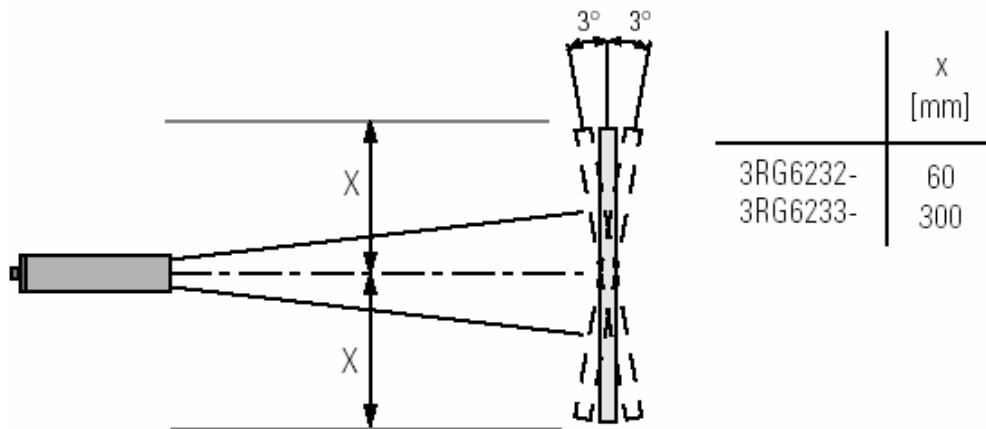


Figura 26: Ángulo máximo de detección del sensor de nivel.

El ajuste del nivel máximo del sensor se realiza con el potenciómetro "P", situado en la parte posterior. Según se puede ver en la siguiente figura.

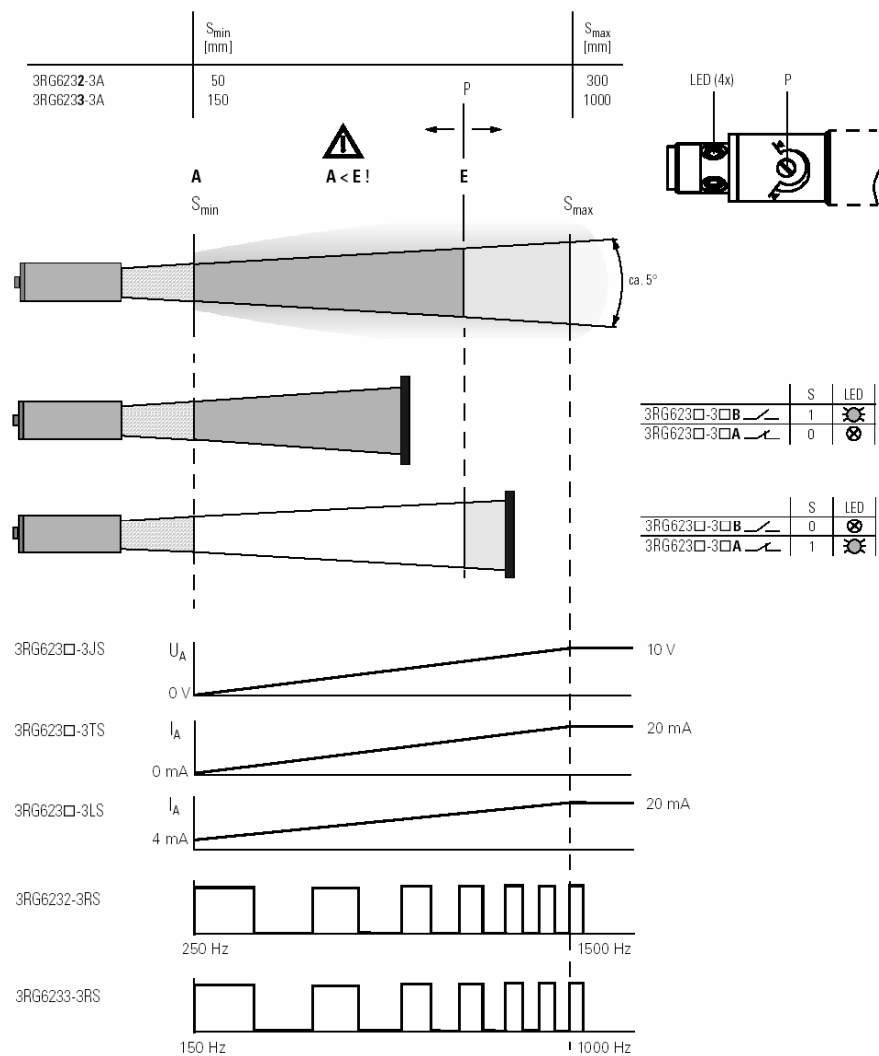


Figura 27: Ajuste del sensor de nivel.

Procedimiento de ajuste

Ajustaremos el sensor para operar entre los valores de 300 y 0 mm para obtener una señal de corriente entre 4 y 20 mA. Que posteriormente con un acondicionador de señal externo transformaremos en una señal de tensión variable entre 0 y 10 voltios y finalmente disponer en la posición de memoria del PCL etiquetada como "Nivel_Valor" con la lectura graduada en milímetros serigrafiada en la pared del tanque, de 300 a 0mm.

1. Llenamos el tanque hasta el nivel máximo indicado en la pared del tanque, 300 mm.
2. En la salida del acondicionador medimos la señal de tensión con ayuda del polímetro.
3. Vaciamos el tanque hasta el valor mínimo de 10 mm.
4. En la salida del acondicionador medimos la señal de tensión con ayuda del polímetro y ajustamos la señal para obtener una tensión de:

$$V = 10(mm) \cdot \frac{10(V)}{300(mm)} = 0,33V$$

5. Comprobamos la linealidad del sensor, según la escala ajustada de 0,33 Voltios por centímetro.

Internamente en el autómatas esta señal analógica leída por el canal 0 (PEW 752) es convertida a formato digital de 27648 puntos. Para tener un valor fácilmente interpretable, se utiliza la función de escalado, FC105 incluida en la librería "Converting Block" del software del autómatas.

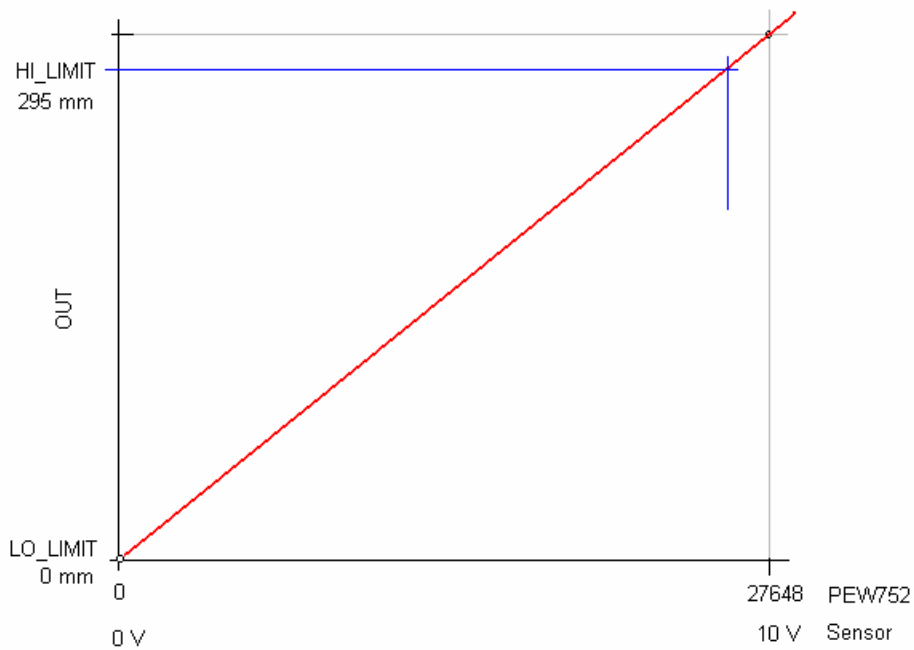


Figura 28: Ajuste de escala para entrada analógica.

Utilizaremos las variables simbólicas siguientes:

Simbólico	Variable	Valor inicial
"ANALOG".NIVEL_VALOR	DB1.DBD0	0
"ANALOG".NIVEL_MAX	DB1.DBD4	298
"ANALOG".NIVEL_MIN	DB1.DBD8	0
"ANALOG".NIVEL_ERR	DB1.DBD12	0
"ALRMA_NIVEL"	M20.0	0

Bloque: OB1 "Main Program Sweep (Cycle)"

Segm.: 1 Escalado señales analógicas

```
CALL "SCALE"          FC105          -- Scaling Values
IN      :=PEW752
HI_LIM :="ANALOG".NIVEL_MAX  DB1.DBD4  -- LECTURA MAXIMA
LO_LIM :="ANALOG".NIVEL_MIN  DB1.DBD8  -- LECTURA MINIMA
BIPOLAR:=FALSE
RET_VAL:="ANALOG".NIVEL_ERR  DB1.DBW12  -- ERROR LECTURA NIVEL
OUT     :="ANALOG".NIVEL_VALOR DB1.DBD0  -- NIVEL DEPOSITO SALIDA DE LA FUNCIÓN ESCALA
```

Segm.: 2 Gestión de errores escalado

```
L      "ANALOG".NIVEL_ERR  DB1.DBW12  -- ERROR LECTURA NIVEL
L      0
==I
SPB    ET01
L      0
T      "ANALOG".NIVEL_VALOR DB1.DBD0  -- NIVEL DEPOSITO SALIDA DE LA FUNCIÓN ESCALA
S      "ALARMA_NIVEL"      M20.0
ET01: NOP 0
```

Los valores límites máximo y mínimo de la se han invertido para obtener una lectura del sensor inversa y que corresponda con la lectura serigrafiada en el depósito.

En el ejemplo se puede apreciar como accedemos al valor de periferia PEW752 mediante la función FC105. Los parámetros de la función son:

- **IN – Word - entrada:** valor de periferia indicado en la parte de hardware de Step 7 para dicho canal a leer. (PEW752)
- **HI_LIMIT – real - entrada:** límite superior a escalar el valor de periferia. Es aconsejable dejarlo como parámetro de una DB para poder modificar el escalado de la analógica a posterioridad desde un panel o scada.
- **LO_LIMIT – real - entrada:** límite inferior a escalar el valor de periferia. Es aconsejable dejarlo como parámetro de una DB para poder modificar el escalado de la entrada analógica a posterioridad desde un panel o scada.
- **BIPOLAR – bool - entrada:** Parámetro que indica si el valor a leer en la periferia va a poseer valores negativos o no. En configuraciones +/-10V y +/-20 mA y PT100, en nuestro caso deberá de configurarse como FALSE.
- **RET_VAL – word - entrada:** Código de error de la lectura. Si el valor analógico de la periferia rebasa los márgenes establecidos por hardware o software, desde esta función el valor de esta palabra es distinto de cero.
- **OUT –real – salida:** Valor escalado y convertido a real.

Electro-Válvula Proporcional



Figura 29: Válvula de control proporcional.

La electro válvula está situada en una de las tuberías de conexión entre los depósitos B101 y B102, identificada como Y106. Lleva incorporado un control electrónico que permite regular su apertura y por tanto el caudal de flujo circulante en función de la tensión aplicada en un rango de 0 – 10 V.

Para el ajuste y calibrado de la válvula, disponemos de tres potenciómetros, etiquetados como R1, R2 y R3 que nos permitirán ajustar los valores de cero, ganancia y tiempo de respuesta, según el siguiente procedimiento:

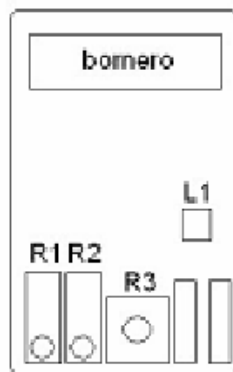
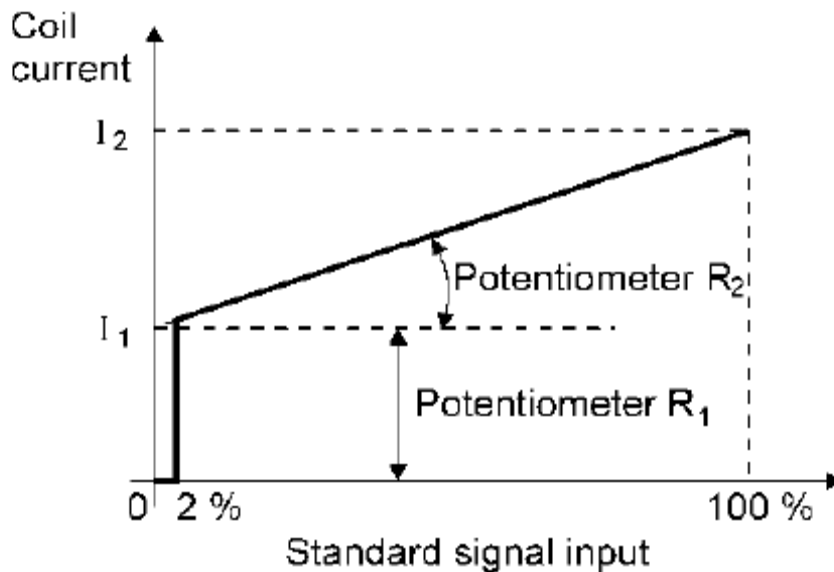


Figura 30: componentes de ajuste de válvula.

Destapar la cubierta plástica. Quedarán a la vista los potenciómetros de ajuste R1, R2 y R3, Mediante estos potenciómetros podremos determinar:

- R1: El “cero”, o valor de señal por debajo del cual la válvula está cerrada.
- R2: El valor de fondo de escala, o caudal máximo.
- R3: El tiempo de respuesta del actuador.

En la siguiente gráfica se ve el efecto de los potenciómetros R1 y R2 que permiten ajustar el rango de trabajo.



Ajuste del cierre de válvula:

1. Con la bomba a régimen máximo, y controlando la electro-válvula con una señal de 0 – 10V, aplicamos una tensión de control del 2% del fondo de escala (0,2V).
2. Con esta tensión mínima, ajustar R1 al mínimo (válvula cerrada)

3. Aumentar R1 progresivamente hasta que tengamos la mínima lectura de caudal, con ayuda del caudalímetro insertado en la misma línea.
4. Volver a disminuir lentamente R1 hasta que la válvula cierre completamente.

Ajuste del máximo caudal:

1. En las mismas condiciones, aplicar una tensión de 10V, correspondiente a la apertura total de la electro-válvula.
2. Ajustar R2 al máximo, hasta obtener la lectura máxima en el caudalímetro insertado en línea.
3. Ir reduciendo el valor de R2 hasta el punto en que se observe que la lectura del caudalímetro disminuye.
4. Volver a aumentar R2 hasta que el caudal ya no aumente más.
5. Reducir otra vez R2 justo al valor que empiece a afectar al caudal.

Ajuste de la velocidad de respuesta:

Mediante el potenciómetro R3 podemos ajustar el tiempo de respuesta del control electrónico del actuador ante un cambio de consigna.

5.- Control PID con S7

Introducción

La regulación PID en los S7 se realiza por software a través de funciones, por lo que la efectividad de la regulación dependerá del código programado y de la velocidad de la CPU. En los S7-300 las funciones de control PID pueden encontrarse programadas en el firmware de la CPU o mediante funciones importadas de la librería, la primera opción es más rápida y eficaz pero sólo está disponible en algunas CPU de la serie, mientras que la segunda es más general y puede implementarse en todas las CPU con el inconveniente del consumo de recursos de memoria, aproximadamente 2 k de memoria por lazo de control.

Funciones de la librería S7

Dentro de la librería “*Standard Library*” podemos encontrar el bloque de funciones para control PID, en el que están incluidas las siguientes funciones:

- **FB 41 – CONT_C:** Función para regulación continua PID.
- **FB 42 – CONT_S:** Función para regulación discontinua.
- **FB 43 – PULSEGEN:** Función para regulación por ancho de pulso.

Para poder utilizar estas funciones basta con copiarlas dentro de los bloques de nuestro proyecto, y posteriormente llamarlas desde nuestro programa, Al copiarlas en el proyecto se crearan bloques de datos DB asociados a las FB's. Asociando cada bloque de datos para cada una de

las llamadas que realicemos a estas funciones, lo cual constituirá un regulador independiente por llamada.

La llamada debe realizarse de forma periódica, con un intervalo de tiempo fijo, para lo que se utiliza el OB 35, programando su periodo en función del proceso que estemos controlando y siempre con un intervalo superior al ciclo de scan del PLC.

Función para regulación PID continua (FB 41)

Se trata de la función **FB 41**, etiquetada como “**CONT_C**”. El regulador puede aplicarse como regulador PID de consigna fija, autónomo o también en regulaciones de varios lazos como regulador en cascada. Su modo de trabajo se basa en el algoritmo de regulación PID del regulador muestreado con señal de salida analógica, implementando un regulador PID completo con salida continua de magnitud manipulada y posibilidad de influenciar manualmente el valor manipulado.

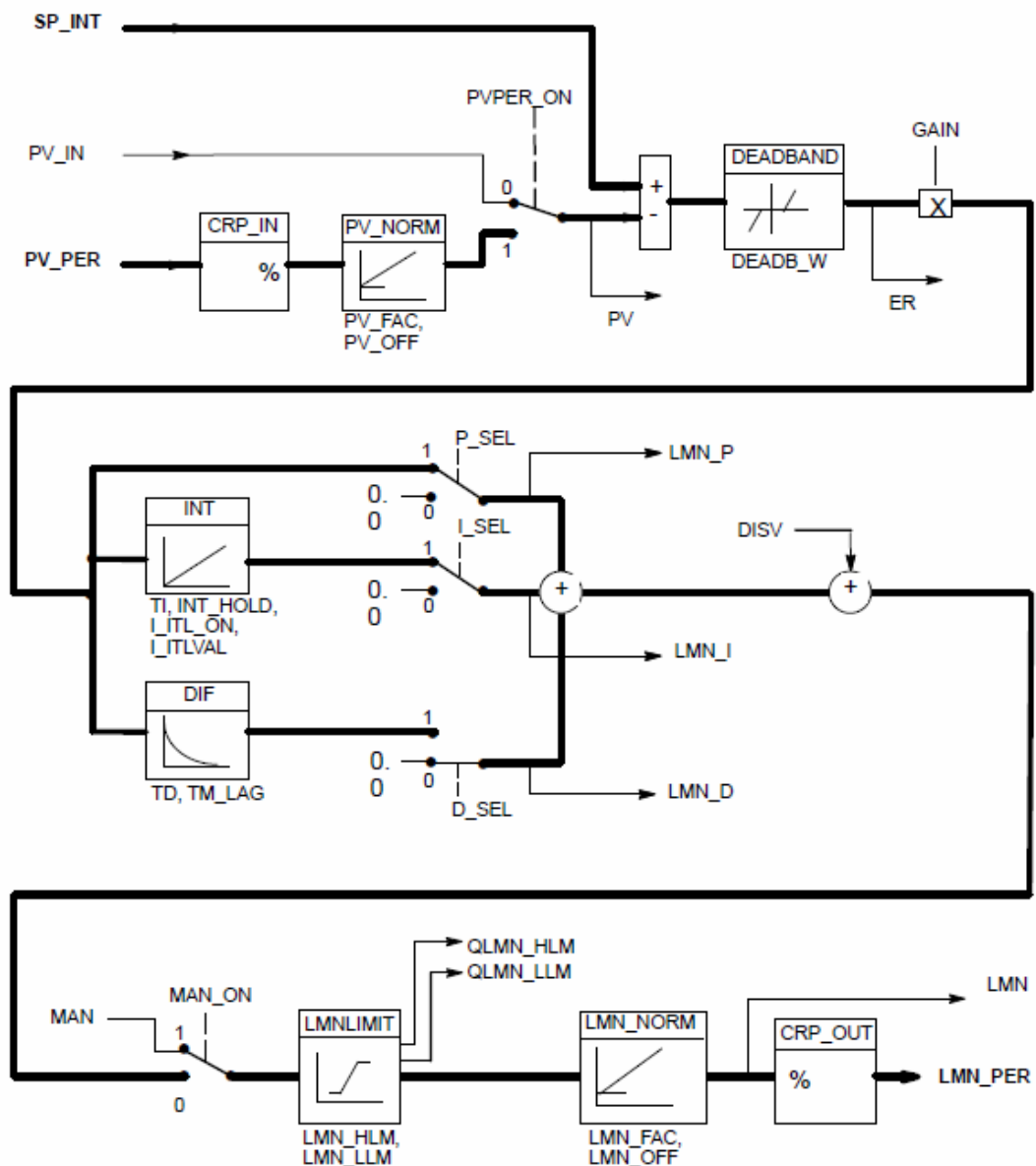


Figura 31: Diagrama de bloques del regulador PID (FB41).

En este esquema de bloques, pueden identificarse los siguientes bloques y señales que se describen a continuación:

Rama de valor de consigna: SP_INT: El valor de consigna que se introduce en formato en coma flotante.

Rama de valor real: El valor real puede ser leído en formato de periferia y en formato en coma flotante. El bloque CRP_IN transforma el valor de periferia PV_PER en un formato en coma flotante de -100 ... +100 %, según la fórmula siguiente:

$$CPR_IN = PV_PER * (100 / 27648)$$

La función PV_NORM normaliza la salida de CRP_IN según la fórmula siguiente:

$$PV_NORM = CPR_IN * PV_FAC + PV_OFF$$

El valor por defecto de PV_FAC es 1 y el de PV_OFF es 0.

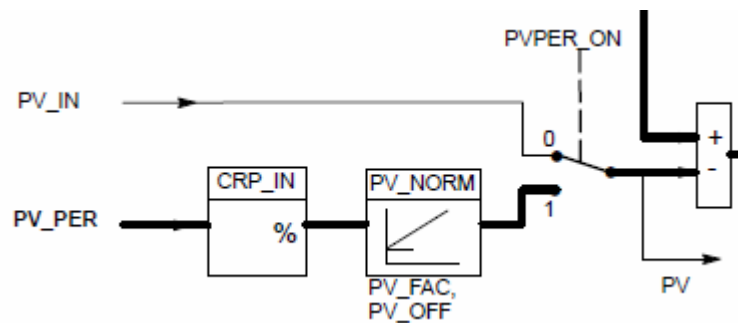


Figura 32: Regulador PID, rama de valor real.

Formación del error de regulación: La diferencia entre el valor de consigna y el valor real es el error de regulación. Para suprimir la pequeña oscilación permanente debida a la cuantificación de la magnitud manipulada el error de regulación se hace pasar por una zona muerta (DEADBAND). Cuando DEADB_W = 0, la zona muerta está desconectada.

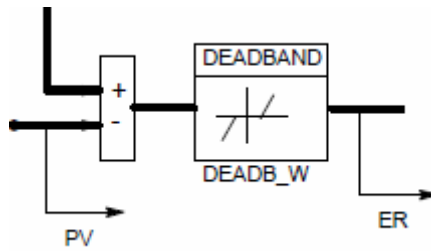


Figura 33: Regulador PID, Error de regulación.

Algoritmo PID: El algoritmo PID trabaja en el algoritmo de posición. Las acciones proporcional (GAIN), integral (INT) y derivativa (DIF) están conectadas en paralelo y pueden conectarse y desconectarse individualmente. De esta forma pueden parametrizarse reguladores P, PI, PD y PID. Pero también son posibles reguladores I puros.

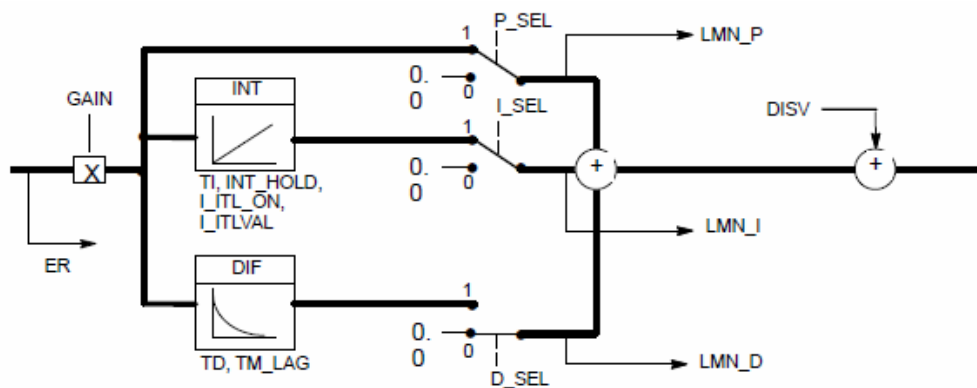


Figura 34: Regulador PID, Algoritmo.

Procesamiento de valores manuales: Es posible conmutar entre modo manual y modo automático. En el modo manual, la magnitud manipulada sigue a un valor ajustado manualmente.

Para conmutar al modo automático sin cambios bruscos, el integrador (INT) se pone internamente a $LMN - LMN_P - DISV$ y el diferenciador (DIF) se pone a 0 y se compensa internamente.

Procesamiento de valores manipulados: El valor manipulado se limita con el bloque LMNLIMIT a valores prefijables. La superación de los límites por parte de la magnitud de entrada se señala mediante bits al efecto.

El bloque LMN_NORM normaliza la salida de LMNLIMIT según la fórmula siguiente:

$$LMN = LMNLIMIT * LMN_FAC + LMN_OFF$$

El valor por defecto de LMN_FAC es 1 y el de LMN_OFF es 0.

El valor manipulado está también disponible en formato de periferia. La función CRP_OUT transforma el valor en coma flotante LMN en un valor de periferia según la fórmula siguiente:

$$LMN_PER = LMN * (100 / 27648)$$

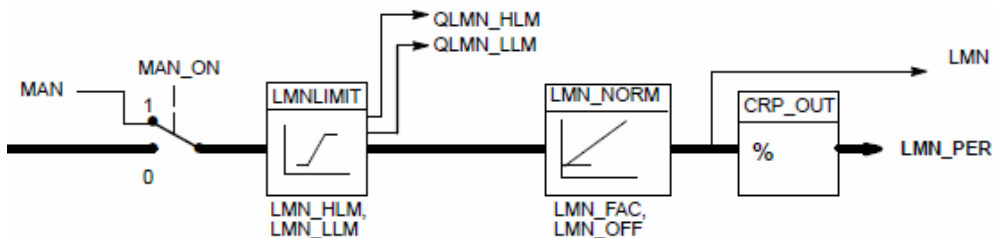


Figura 35: Regulador PID, procesamiento de valores manipulados.

Aplicación de magnitud perturbadora (control anticipativo): En la entrada **DISV** puede aplicarse aditivamente una magnitud perturbadora.

Rearranque completo / Rearranque

El FB "CONT_C" dispone de una rutina de rearmado completo que se ejecuta cuando el parámetro de entrada COM_RST = TRUE.

Al arrancar, el integrador se ajusta internamente al valor de inicialización I_ITVAL. Si se llama en un bloque de alarma cíclica, continúa trabajando a partir de este valor. Todas las demás salidas se ponen a sus valores por defecto.

Parámetros de ENTRADA de la función FB 41

Parámetro (tipo de dato)	Valores posibles (Valor por defecto)	Descripción
COM_RST (BOOL)	(FALSE)	<p>COMPLETE RESTART / Rearranque completo del PID.</p> <p>La activación de esta entrada produce la inicialización de los valores de salida del PID. Esta acción debe de realizarse en el paso de STOP-RUN del PLC ya que si se desconecta por cualquier motivo la sonda que indica el valor de retorno del PID (señal <i>PV_IN</i>) el PID va acumulando cálculos de error, y en su caso abriendo o cerrando la salida, según sea una regulación positiva o negativa. Si se vuelve a conectar la sonda, hasta que desaparezca el error acumulado en el PID no volverá la regulación a entrar en los márgenes de regulación, por lo que continuará abierta o cerrada según el caso. Para evitar esto, se puede colocar una marca que se activará siempre en la OB100, y que se reseteará después del último PID. Esta marca activará <i>COM_RST</i>, de tal manera que cada vez que se pare la instalación se reinician los reguladores.</p>
MAN_ON (BOOL)	(TRUE)	<p>MANUAL VALUÉ ON / Conectar modo manual.</p> <p>Si está activada la entrada "Conectar modo manual", está interrumpido el lazo de regulación. Como valor manipulado se fuerza un valor manual. Por lo que si no se modifica, el regulador continuará en manual dando en la salida el valor que indiquemos en el parámetro MAN. Es recomendable dejar este parámetro como una opción seleccionable desde un equipo de visualización, para poder pasar el regulador a manual/automático.</p>

Parámetro (tipo de dato)	Valores posibles (Valor por defecto)	Descripción
PVPER_ON (BOOL)	(FALSE)	<p>PROCESS VARIABLE PERIPHERY ON / Conectar valor real de periferia</p> <p>Existen dos formas de introducir el valor de la magnitud a regular (la temperatura, la presión, la humedad...): leer una entrada analógica mediante la función FC105, que convierte el valor de periferia a un valor real, que se suele almacenar en un dato real de una DB, o leer directamente el valor de la periferia si ser tratado antes, directamente en formato WORD. Mediante el parámetro <i>PVPER_ON</i> podemos seleccionar con FALSE, que tome el valor de periferia del parámetro <i>PV_IN</i>, y con TRUE, que tome el parámetro de <i>PV_PER</i>.</p> <p>Es generalmente interesante dejarlo a FALSE, que es como viene por defecto, e introducir el valor de proceso para el regulador escalándolo anteriormente mediante una llamada a la FC105, ya que así podemos por un lado parametrizar el escalado de la magnitud del proceso mediante los límites superior e inferior, y por otro, disponemos del valor ya escalado, para poder visualizarlo en un equipo OP o scada. Pro defecto viene a FALSE.</p>
P_SEL (BOOL)	(TRUE)	<p>PROPORTIONAL ACTION ON / Conectar acción P</p> <p>En el algoritmo PID pueden conectarse y desconectarse individualmente las acciones PID. La acción proporcional está conectada si está activada la entrada "Conectar acción P".</p>
I_SEL (BOOL)	(TRUE)	<p>INTEGRAL ACTION ON / Conectar acción I</p> <p>En el algoritmo PID pueden conectarse y desconectarse individualmente las acciones Pfd. La acción integral está conectada si está activada la entrada "Conectar acción I".</p>
INT_HOLD (BOOL)	(FALSE)	<p>INTEGRAL ACTION HOLD / Congelar acción I</p> <p>La salida del integrador puede congelarse, de tal manera que el regulador no actúa frente a errores estacionarios. Para ello se ha de activar la entrada "Congelar acción I".</p>

Parámetro (tipo de dato)	Valores posibles (Valor por defecto)	Descripción
I_ITL_ON (BOOL)	(FALSE)	INITIALIZATION OF THE INTEGRAL ACTION / Inicializar acción I La salida del integrador puede inicializarse a la entrada I_ITLVAL. Para ello se ha de activar la entrada "Inicializar acción I".
D_SEL (BOOL)	(FALSE)	DERIVATIVE ACTION ON / Conectar acción D En el algoritmo PID pueden conectarse y desconectarse individualmente las acciones PID. La acción derivativa está conectada si está activada la entrada "Conectar acción D".
CYCLE (TIME)	≥ 1ms (T#1s)	SAMPLE TIME / Tiempo de muestreo El tiempo entre las llamadas del bloque debe ser constante. La entrada "Tiempo de muestreo" indica el tiempo entre las llamadas del bloque. La función debe ser llamada desde una subrutina que no dependa del tiempo de ciclo del PLC, que es variable dependiendo de si se ejecutan unas instrucciones u otras. Esto se consigue con la ejecución del PID dentro de una OB de tiempo (OB 35 p. Ej). El tiempo de ciclo indicado en este parámetro deberá de ser mayor o igual al tiempo de llamada de la OB (en el caso de la OB 35 viene prefijado a 100 ms, aunque se puede cambiar desde hardware de Step 7). Valores de CYCLE inferiores no son efectivos.
SP_INT (REAL)	-100.0... 100.0(%) ó magnitud física (1) (0.0)	INTERNAL SETPOINT / Consigna interna La entrada "Consigna interna" sirve para ajustar un valor de consigna.
PV_IN (REAL)	-100.0... 100.0 (%) ó magnitud física (1) (0.0)	PROCESS VARIABLE IN / Entrada de valor real En la entrada "Entrada de valor real" puede parametrizarse un valor de puesta en servicio, o aplicarse un valor real externo en formato en coma flotante. La función del PID es controlar una magnitud física. En este parámetro se refleja el valor actual de dicha magnitud (la temperatura de la sala, la presión de la tubería, la humedad del ambiente...). Normalmente aquí se indica un valor que es obtenido anteriormente de la FC105, escalando una entrada analógica.

Parámetro (tipo de dato)	Valores posibles (Valor por defecto)	Descripción
PV_PER (WORD)	(W# 16#00 00)	<p>PROCESS VARIABLE PERIPHERY / Valor real de periferia</p> <p>El valor real en formato de periferia se aplica al regulador en la entrada "Valor real de periferia". No es necesario tratar anteriormente la entrada analógica, ya que se puede indicar en este parámetro dicho valor sin tener que escalarlo. Para seleccionar entre este valor y el anterior, ver el parámetro anterior PVPER_ON.</p>
MAN (REAL)	-100.0...100.0 (%) ó magnitud física (2) (0.0)	<p>MANUAL VALUE IN / Valor manual</p> <p>La entrada "Valor manual" sirve para establecer un valor manual mediante función de manejo/visualización (interface hombre máquina). Este parámetro se traslada a la salida del PID cuando indicamos el funcionamiento manual del PID. Observar que lo que se indica en este parámetro no es la temperatura, sino la posición de la compuerta de calor que la regula, no es la presión, sino las revoluciones del variador que la genera, no es la humedad, sino la cantidad de aire frío que la condensa, etc...</p>
GAIN (REAL)	(2.0)	<p>PROPORTIONAL GAIN / Ganancia proporcional</p> <p>La entrada "Ganancia proporcional" indica la ganancia del regulador. La ganancia indica la constante de la parte proporcional del PID. Cuanto mayor sea el parámetro GAIN, mayores serán los incrementos en la evolución de la salida del PID y viceversa. Esto tiene como resultado efectivo que un valor reducido de ganancia genera unas evoluciones lentas en la salida del PID, pero por el contrario el error estacionario que se produce en la regulación es menor. La regulación es lenta pero exacta, tendiendo a un sistema de control sub-amortiguado.</p> <p>Por el contrario, valores altos de ganancia producen evoluciones rápidas en la salida, pero aumenta el error estacionario y se tiende a sistemas sobre-amortiguados. Valores de 1 o inferiores son interesantes para la mayoría de los procesos "tranquilos" en su evolución.</p>

Parámetro (tipo de dato)	Valores posibles (Valor por defecto)	Descripción
TI (TIME)	>= CYCLE (T#20s)	<p>RESET TIME / Tiempo de acción integral</p> <p>La entrada "Tiempo de acción integral" determina el comportamiento temporal del integrador. Este parámetro determina el tiempo que se tardará en realizar la integración. Esto no quiere decir que hasta que no se cumpla este tiempo no se adiciona la parte integral a la salida, sino que dicha parte se reparte en este periodo de tiempo. Según esto, con periodos de tiempo cortos en este parámetro la evolución de la salida es brusca, conduciendo al sistema a procesos sobre-amortiguados, mientras que valores grandes de tiempo en él generan evoluciones suaves, tendiendo a sistemas sub-amortiguados.</p> <p>Un tiempo de 120 segundos en este parámetro es usualmente interesante para la mayoría de los procesos.</p>
TD (TIME)	>= CYCLE (T#10s)	<p>DERIVATIVE TIME / Tiempo de diferenciación (acción derivativa)</p> <p>La entrada "Tiempo de diferenciación" determina el comportamiento temporal del diferenciador. El equivalente al anterior pero para la parte derivativa. Igualmente, valores reducidos generan inestabilidad y rápida evolución, mientras que valores grandes tienden a reducir la propia acción derivativa.</p>
TM_LAG (TIME)	>= CYCLE/2 (T#2s)	<p>TIME LAG OF THF DERIVATIVE ACTION / Tiempo de retardo de la acción D</p> <p>El algoritmo de la acción D contiene un retardo que puede parametrizarse en la entrada "Tiempo de retardo de la acción D". La acción derivativa se encuentra en función de la derivada de la magnitud del proceso, es decir, de la pendiente de variación de dicha magnitud. Se permite con este parámetro retardar dicha acción, para que la propia entrada de la acción derivativa no influya en la variación de la salida, y se retroalimente a sí misma disparándose la parte derivativa del PID.</p>

Parámetro (tipo de dato)	Valores posibles (Valor por defecto)	Descripción
DEADB_W (REAL)	>= 0.0 (%) ó magnitud física (1) (0.0)	<p>DEAD BAND WIDTH / Ancho de zona muerta</p> <p>El error de regulación se conduce por una zona muerta. La entrada "Ancho de zona muerta" determina el tamaño de la zona muerta. Cuando el valor de proceso alcanza la consigna seleccionada, la salida del PID disminuye hasta alcanzar si fuese el caso el valor nulo. Una disminución de la salida suele producir un alejamiento del valor real de su consigna, con lo cual vuelve a actuar la acción del PID, recuperando la consigna, y repitiéndose el ciclo. Estas pequeñas fluctuaciones (o grandes, según sea el parámetro GAIN) en algunos procesos generan más complicaciones que beneficios, no siendo recomendables. El ejemplo más claro es el control de llenado de un recipiente. Si deseamos mantener su altura constante, independientemente del agua que extraigamos a través de un grifo que vacía el recipiente, aplicaremos un PID a la altura del mismo. Pero al alcanzar la consigna de altura, el variador que impulsa el agua de llenado tenderá a parar. Ya sea por pequeñas olas del recipiente o por el propio vaciado, inmediatamente se generará un error que volverá a arrancar el variador, pero debido a que este error es muy pequeño, la acción del PID será igualmente baja, por lo que funcionará a bajas revoluciones, con el consiguiente perjuicio a la refrigeración del motor conectado.</p> <p>La solución pasa por una banda muerta, que se adiciona a la consigna, y sólo después de haber disminuido la altura del recipiente por debajo de dicho valor actuará el algoritmo PID. Como en ese momento el error a corregir será suficientemente elevado, el variador arrancará a una frecuencia prudente para la vida útil del motor.</p>

Parámetro (tipo de dato)	Valores posibles (Valor por defecto)	Descripción
LMN_HLM (REAL)	LMN LLM ... 100.0 (%) ó magnitud física 2)	MANIPULATED VALUE HIGH LIMIT / Límite superior del valor manipulado El valor manipulado tiene siempre un límite superior y uno inferior. La entrada "Límite superior del valor manipulado" indica el límite superior. El valor máximo que puede alcanzar la salida del PID, medido en unidades del actuador del proceso: velocidad máxima del variador que controla el ventilador, posición máxima de la compuerta que controla el aire, posición máxima de la electro-válvula que deja pasar el agua caliente, etc...
LMN_LLM (REAL)	-100.0... LMN HLM (%) ó magnitud física 2) (0.0)	MANIPULATED VALUE LOW LIMIT / Límite inferior del valor manipulado El valor manipulado tiene siempre un límite superior y uno inferior. La entrada "Valor manipulado, límite inferior" indica el límite inferior. El valor mínimo de dicha salida. Normalmente es cero en la mayoría de los procesos.
PV_FAC (REAL)	(1.0)	PROCESS VARIABLE FACTOR / Factor de valor real La entrada "Factor de valor real" se multiplica por el valor real. La entrada sirve para la adaptación del margen de valor real. Valor por el que multiplicar el valor de PV_PER. Es importante destacar que esta constante sólo afecta al valor real de periferia, no al valor interno PV_IN, que suele ser el utilizado en la mayoría de las regulaciones.
PV_OFF (REAL)	(0.0)	PROCESS VARIABLE OFFSET / Offset del valor real La entrada "Offset del valor real" se suma con el valor real. La entrada sirve para la adaptación del margen de valor real. este valor se suma al valor PV_PER * PV_FAC, resultando: Valor real de periferia = PV_OFF + PVPER * PV_FAC
LMN_FAC (REAL)	(1.0)	MANIPULATED VALUE FACTOR / Factor del valor manipulado La entrada "Factor del valor manipulado" se multiplica por el valor manipulado. La entrada sirve para la adaptación del margen de valor manipulado. Constante que multiplica a la salida del PID

Parámetro (tipo de dato)	Valores posibles (Valor por defecto)	Descripción
LMN_OFF (REAL)	(0.0)	MANIPULATED VALUE OFFSET / Offset del valor manipulado. La entrada "Offset del valor manipulado" se suma al valor manipulado. La entrada sirve para la adaptación del margen de valor manipulado. Este valor se suma al valor de salida del PID después de haberlo multiplicado por LMN_FAC. Por lo tanto, el valor de salida del PID será: Valor de salida del PID = LMN_OFF + salida * LMN_FAC
I_ITLVAL (REAL)	-100.0... 100. ó magnitud física (2) (0.0)	INITIALIZATION VALUE OF THE INTEGRAL ACTION / Valor de inicialización de la acción I. La salida del integrador puede forzarse con la entrada I_ITL_ON. En la entrada "Valor de inicialización de la acción I" está el valor de inicialización. Se puede seleccionar un valor de comienzo para la acción integral. Esto puede ser interesante en procesos en los cuales al arrancar el PID existe un gran error que debe de ser corregido inmediatamente, sin embargo se desea que una vez alcanzada la consigna evolucione muy lentamente la salida del PID. Se selecciona un valor de tiempo de integración alto, pero se inicializa la acción integral a un valor que permita evolucionar la principio con presteza.
DISV (REAL)	-100.0.. .100. 0(%) ó magnitud física (2) (0.0)	DISTURBANCE VARIABLE / Magnitud perturbadora Para control anticipativo de la magnitud perturbadora, ésta se conecta en la entrada "Magnitud perturbadora". Si se conoce una magnitud que puede influir en el proceso, además lógicamente de la magnitud a controlar, se puede indicar dicho valor en este parámetro, aunque su influencia debe de ser reducida, ya que en caso contrario nos conduciría a un sistema de segundo orden que se convertiría en inestable.

(1) Parámetros en las ramas de valor de consigna y de valor real con la misma unidad.

(2) Parámetros en la rama de valor manipulado con la misma unidad.

Parámetros de SALIDA de la función FB 41

Parámetro	Valor por defecto	Descripción
LMN (REAL)	0.0	MANIPULATED VALUÉ / Valor manipulado En la salida "Valor manipulado" se saca en formato en coma flotante el valor manipulado que actúa efectivamente. Este valor se envía a la salida analógica que regule el actuador de la instalación. Lógicamente, será necesario después del PID realizar una llamada a la FC106 para convertir el valor real a una palabra de la periferia del autómeta.
LMN_PER (WORD)	W#16#0000	MANIPULATED VALUÉ PERIPHERY / Valor manipulado periferia Esta salida entrega el valor manipulado en formato de periferia. Si no se desea realizar un escalado de la variable mediante la FC106, en este parámetro disponemos de la salida ya en formato de 16 bits para poder ser enviada a la periferia del autómeta.
QLMN_HLM (BOOL)	FALSE	HIGH LIMIT OF MANIPULATED VALUÉ REACHED / Alcanzado el límite superior del valor manipulado El valor manipulado tiene siempre un límite superior y un límite inferior. La salida "Alcanzado el limite superior del valor manipulado" indica la superación de la limitación superior.
QLMN_LLM (BOOL)	FALSE	LOW LIMIT OF MANIPULATED VALUÉ REACHED / Alcanzado el límite inferior del valor manipulado El valor manipulado tiene siempre un límite superior y un límite inferior. La salida "Alcanzado el límite inferior del valor manipulado" indica la superación de la limitación inferior.
LMN_P (REAL)	0.0	PROPORTIONALITY COMPONENT / Acción P La salida "Acción P" contiene la componente proporcional de la magnitud manipulada.
LMN_I (REAL)	0.0	INTEGRAL COMPONENT / Acción I La salida "Acción I" contiene la componente integral de la magnitud manipulada.
LMN_D (REAL)	0.0	DERIVATIVE COMPONENT / Acción D La salida "Acción D" contiene la componente diferencial de la magnitud manipulada.
PV (REAL)	0.0	PROCESS VARIABLE / Valor real Por la salida "Valor real" se emite el valor real que actúa efectivamente.

Parámetro	Valor por defecto	Descripción
HR (REAL)	0.0	ERROR SIGNAL / Error de regulación Por la salida "Error de regulación" se emite la diferencia o error de regulación que actúa efectivamente.

Características de los parámetros en los controladores PID [6]

Un controlador proporcional (K_p) tendrá el efecto de reducir el tiempo de subida, pero nunca eliminar el error de estado estacionario.

Un control integral (K_i) tendrá el efecto de eliminar el error de estado estacionario, pero puede que la respuesta transitoria empeore.

Un control derivativo (K_d) tendrá el efecto de aumentar la estabilidad del sistema, reduciendo el sobrepico, y mejorando la respuesta transitoria.

Los efectos de cada uno de los parámetros de un controlador K_p , K_d , y K_i en un sistema realimentado se resumen en la siguiente tabla:

RESPUESTA	Tiempo de subida	Sobrepaso	Tiempo de establecimiento	ERROR
K_p	Disminución	Aumento	Pequeños cambios	Disminución
K_i	Disminución	Aumento	Aumento	Eliminado
K_d	Pequeños cambios	Disminución	Disminución	Pequeños cambios

Hay que tener en cuenta que estas características están relacionadas, ya que K_p , K_i y K_d son dependientes el uno del otro. De hecho, el cambio de una de estas variables puede cambiar el efecto de las otras dos. Por esta razón, la tabla sólo se debe utilizar como referencia cuando estén determinados los valores de K_i , K_p y K_d .

Sintonía de los parámetros del PID con S7

Existen numerosos métodos para la sintonización de los parámetros del regulador PID pero sin duda los más adecuados son algoritmos adaptados al propio software del PLC, en la gama Simatic encontramos como algoritmo de sintonía PID Self-Tuner con la función "TUNING_C".

Con este algoritmo se consigue ajustar el lazo de regulación en un primer ajuste completo online, así como un ajuste de adaptación online sobre la marcha, es ideal para usar lazos de regulación de temperatura, nivel y caudal y es compatible con las CPU 313 o superiores.

PID Self-Tuner TUNING_C

Los bloques de función tienen los modos de operación siguientes:

- Ajuste inicial online del regulador PID actuando en un proceso desconocido. Excitación mediante escalón de consigna.
- Adaptación online del regulador PID actuando en un proceso ajustado por primera vez. Reoptimización utilizando escalones de consigna sensiblemente inferiores.
- Conmutación de estructura en reguladores PID con escalones de consigna positivos. Para evitar rebases al calentar sistema regulados en temperatura.
- Modo manual para el control manual del regulador.

FB "TUNING_C" sintoniza automáticamente un controlador de PID continuo. Los cambios de punto de trabajo y los ligeros cambios en el comportamiento del proceso, pueden ser optimizado online. Si hay un

cambio positivo en el punto de ajuste, puede ajustarse un sobrepaso máximo que puede evitar situaciones de riesgo para el proceso.

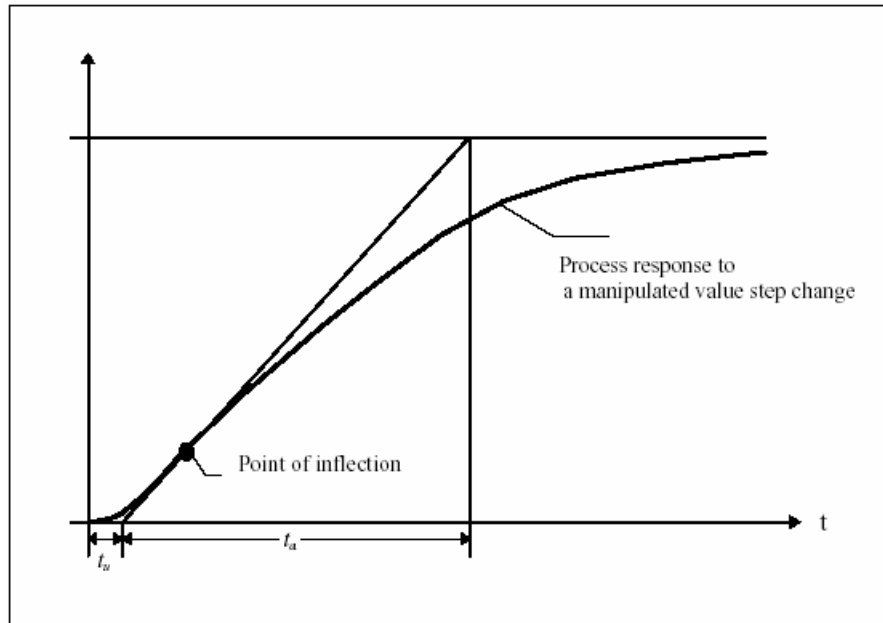
Los requisitos de la CPU son los siguientes:

Memoria requerida	Memoria de carga	Memoria de trabajo
Espacio ocupado por FB en memoria	aprox. 6542 bytes	aprox. 5956 bytes
Espacio ocupado por DB en memoria	644 bytes	294 bytes
Tiempos de ejecución		
en S7-300	1,0 ... 1,5 ms	
en S7-400	0,06 ... 0,19 ms	

Esta función no se encuentra en la librería estándar de step 7, sino que hay que instalarla como una librería adicional. Los requisitos del proceso son los siguientes:

- El proceso debe ser estable con una respuesta asintótica
- Una sola variable controlada
- La ganancia no debe ser demasiado elevada
- Adecuada calidad de las señales medidas

El proceso debe tener un Estado estable, la respuesta transitoria asintótica con un desfase de tiempo. Después de un cambio de consigna en la variable manipulada (LMN) la variable de proceso debe cambiar a un estado estacionario como se muestra en la siguiente figura. Esto excluye por lo tanto procesos que tienen una respuesta oscilante sin control y los procesos que no se auto-regulan (integrador en el proceso).



El proceso no debe implicar grandes retrasos de tiempo. El rango de aplicación puede ser especificado basado en la relación del tiempo de retardo, t_u .

$$10 t_u < t_a$$

En el ajuste inicial de un controlador PID, la duración de la fase de aprendizaje puede ser significativamente larga y el rebasamiento puede ocurrir durante la fase de aprendizaje sobre todo con las combinaciones elevadas de la ganancia de proceso.

Parámetros de entrada TUNING_C

Parámetro (tipo de dato)	Comentario	Rango de Valores Permitido	Valor por Defecto
SP (REAL)	consigna	valores según parámetro a controlar	0,0
PV (REAL)	variables de proceso	valores según parámetro a controlar	0,0
LMN (REAL)	valor manipulado	0,0 a 100,0 (%)	0,0

Parámetro (tipo de dato)	Comentario	Rango de Valores Permitido	Valor por Defecto
MIN_STEP (REAL)	mínima consigna	>10% del rango de operación de la consigna y la variable del proceso	10,0
LHLM_TUN (REAL)	máximo valor de la variable manipulada en el autoajuste	0,0 a 100,0 (%)	80,0
MAN (REAL)	valor manual	0,0 a 100,0 (%)	0,0
MAN_ON (BOOL)	modo manual		FALSO
STRUC_ON (BOOL)	control variable para saltos de consigna		VERDADERO
PID_ON (BOOL)	PID activado		VERDADERO
COM_RST (BOOL)	reset completo		FALSO
CYCLE (TIME)	tiempo de muestreo	> 1 ms	100 ms

Parámetros de salida TUNING_C

Parámetro (tipo de dato)	Comentario	Rango de Valores Permitido
MAN_OUT (REAL)	valor de salida manual	0,0
CAIN (REAL)	ganancia proporcional	1,0
TI (TIME)	tiempo integral	IOs
TD (TIME)	tiempo derivativo	Os
TM_LAG (TIME)	tiempo de retraso	1 s
PHASE (INT)	fase 0 a 7	0

Parámetro (tipo de dato)	Comentario	Rango de Valores Permitido
QP_INFL (BOOL)	punto de inflexión encontrado	FALSO
QMAN_ON (BOOL)	modo manual	FALSO
QI_SEL (BOOL)	activar acción integral	VERDADERO
QD_SEL (BOOL)	activar acción derivativa	FALSO
QWRITE (BOOL)	escribe parámetros auto-sintonizados en el controlador PID	FALSO

Parámetros de entrada/salida

Parámetro (tipo de dato)	Comentario	Defecto
TUN_ON (BOOL)	ajuste automático completo cuando se produzca un cambio de consigna	FALSO
ADAPT_ON (BOOL)	ajuste fino de adaptación cuando se produzca un cambio de consigna	FALSO
STEADY (BOOL)	Estado estable	FALSO

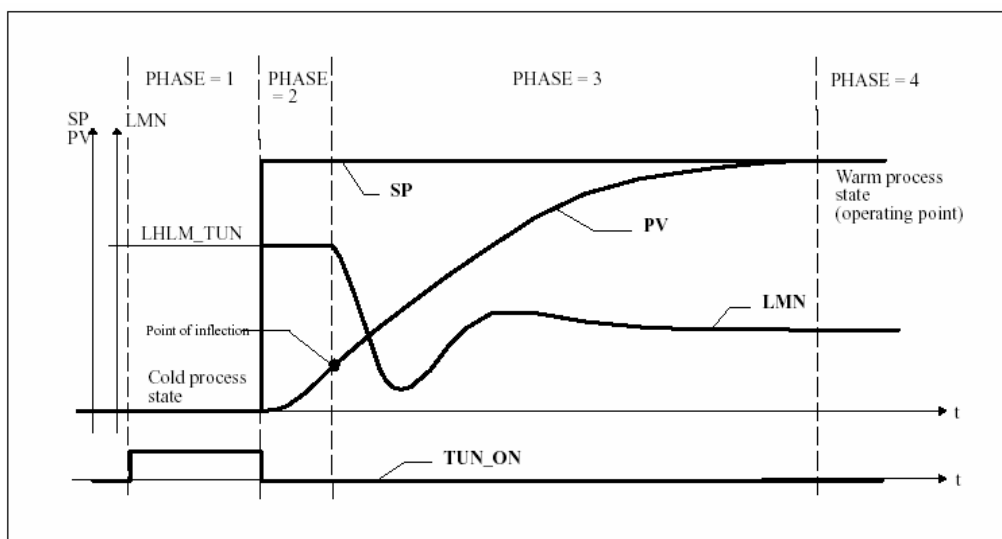
Modos de Funcionamiento

	TUN_ON	ADAPT_ON	STRUC_ON	MAN_ON
Ajuste inicial del controlador PID para un proceso desconocido	VERDADERO	FALSO	cualquier	FALSL
Adaptación del controlador PID para un proceso previamente identificado	FALSO	VERDADERO	cualquier	FALSO

	TUN_ON	ADAPT_ON	STRUC_ON	MAN_ON
Variable PID estructura de mando con cambios de paso consigna positivos	FALSO	FALSO	VERDADERO	FALSO
Modo manual	Cualquier	Cualquier	Cualquier	VERDADERO

Modo "Initial Controller Tuning"

Si TUN_ON = TRUE y hay un cambio de la consigna, y se cumple que este cambio es mayor de MIN_STEP en una dirección positiva, comienza la optimización del proceso de identificación del controlador. Si se desea cancelar el ajuste inicial, debe restablecer TUN_ON a FALSE o cambiar al modo manual, MAN_ON = TRUE.



Fases del proceso de auto-ajuste

El proceso de aprendizaje implica los siguientes pasos o fases:

FASE = 0: Cuando un DB se crea para la función FB "TUNING_C", el parámetro PHASE está por defecto a cero.

FASE = 1: Después de activar TUN_ON, la variable de proceso se inicializa a valor cero. A continuación, debe esperar hasta que la variable

del proceso se mantenga constante. Con ello se logra un estado de equilibrio ("frío" el estado inicial del proceso).

FASE = 2: Tan pronto como se aplica un cambio de consigna mayor que MIN_STEP en una dirección positiva hacia el punto de funcionamiento, MAN_OUT asume el valor de LHLM_TUN y QMAN_ON es puesto a TRUE. Ambos valores se transfieren al controlador PID. El controlador PID es por lo tanto controlado en el modo manual. MIN_STEP debe ser superior al 10% del rango de trabajo de la consigna y la variable del proceso.

FASE = 3: Cuando el punto de inflexión de la respuesta al escalón se detecta (QP_INFL = TRUE) o la variable de proceso ha alcanzado el 60% del cambio de valor de la consigna (QP_INFL= FALSO), ahora un controlador PID se ha sintonizado con cautela. El controlador opera inmediatamente como un controlador PI, intentando llevar este proceso a un estado estacionario. Si se tarda un tiempo extremadamente largo hasta que el estado estacionario llega (ej: respuesta transitoria en los procesos de la temperatura) puede iniciar el diseño de control con los datos actuales, cuando el estado de equilibrio prácticamente se ha logrado mediante STEADY=TRUE. Esto a menudo trae algunas mejoras al diseño del controlador.

Si se produce un sobrepaso o no se encuentra ningún punto de inflexión, la razón puede ser que en el valor manipulado se produjese cambio de LHLM_TUN demasiado alta y no significa necesariamente que esto implique un mal ajuste del controlador. Durante la siguiente puesta a punto inicial, hay que seleccionar LHLM_TUN aproximadamente un 20% inferior.

Si se detecta un estado de equilibrio o si el tiempo es $10 \times T_I$ (T_I : tiempo de reposición del controlador PI establecidos en la FASE = 3), un diseño mejorado del controlador se inicia y el sintonizador se mueve a la FASE = 4. Si PID_ON = TRUE, un nuevo controlador PID es diseñado, de lo contrario un controlador PI. Con procesos difíciles, el bloque siempre diseña un controlador PI independientemente del valor de PID_ON. El valor calculado para la ganancia durante el ajuste inicial se limita, pues,

de manera que la ganancia del bucle del bucle abierto (el producto de la ganancia del controlador y el del proceso) este entre 0,4 y 15.

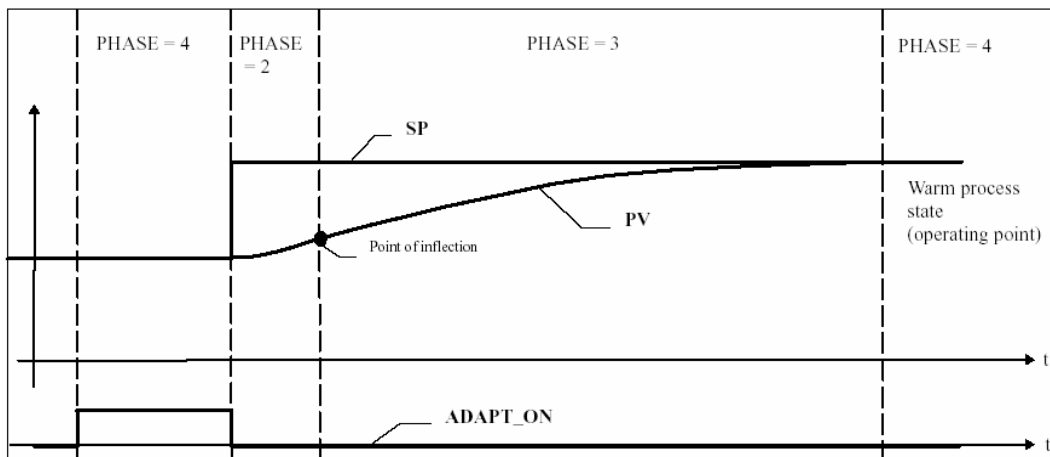
FASE = 4: En esta fase el controlador opera con los parámetros optimizados.

Modo “Controller Adaptation to an Identified Process”

Si ya se ha ajustado el PI o PID, y sólo se desea optimizarlo, se utiliza el modo de "Controlador de adaptación a un proceso identificado"

Si ADAPT_ON = TRUE y se produce un escalón de la consigna, esto desencadena un proceso de identificación con la optimización del controlador. Si desea cancelar la adaptación, debe restablecer ADAPT_ON a FALSE o cambiar al modo manual (MAN_ON = TRUE).

El modo de adaptación utiliza un cambio más pequeño de la consigna, sin embargo, debe asegurarse de que la condición de cambio de referenciasea mayor que MIN_STEP. El cambio de referencia durante la adaptación se encuentra en las proximidades del punto de funcionamiento.



Fases del proceso

El proceso de aprendizaje implica los pasos siguientes:

FASE = 4: El valor manipulado y variable de proceso son constantes, esto significa que un estado de equilibrio se ha alcanzado (punto de operación). Si hay fuertes fluctuaciones, debe cambiarse a un controlador PI. (PID_ON = FALSO). Después de la adaptación, puede cambiar de nuevo al controlador PID (PID_ON = TRUE).

FASE 2 a 4: Esta es al igual que en el proceso de aprendizaje de "Sintonía inicial del controlador PID para un proceso desconocido", con las siguientes diferencias:

Tras el cambio de valor de la consigna, el controlador no se alimenta con el valor de LHLM_TUN, sino con un valor constante a partir de la experiencia previa del proceso.

Si no hay ningún punto de inflexión durante la adaptación (QP_INFL = FALSO), el diseño del controlador no se modifica. Esto significa que el controlador continúa funcionando con los parámetros anteriores.

Controlador manual

Si se establece la entrada MAN_ON=TRUE, la salida QMAN_ON=TRUE y MAN_OUT=MAN. Esto cambia el controlador PID al modo manual (FASE = 7). **El modo manual tiene prioridad sobre todos los otros modos.**

Cualquier ajuste inicial, la adaptación o cambio de la estructura actualmente en proceso es cancelado. Al deshabilitar el modo manual (MAN_ON = FALSE), el controlador cambia al modo automático (PHASH = 4) y continúa utilizando los parámetros del controlador existente. Si no

hay parámetros del controlador, el controlador permanece en el modo manual y muestra el valor cero (PHASH = 1).

Modificación de los parámetros del controlador

Si se desea cambiar los parámetros del controlador GAIN, TI, TD o TM_LAG en el siguiente ajuste inicial o de adaptación, puede sobrescribir los parámetros de salida correspondiente en el bloque TUNING_C, por ejemplo el uso de "controlar y modificar la variable" en la etapa 7.

Si las oscilaciones se producen en el bucle de control cerrado o si hay rebasamiento en los siguientes cambios de consigna, puede reducir la ganancia del controlador (por ejemplo, para $GAIN * 0,8$) y aumentar el tiempo de reposición de TI (por ejemplo a $TI * 1,5$).

Ajuste del tiempo de muestreo

El tiempo de muestreo no deberá ser superior al 10% del tiempo de reposición (TI) calculado para el controlador. Puede establecer el tiempo de muestreo con el parámetro de FB CICLO TUNING_C y del controlador. **Debe** coincidir con la diferencia de tiempo entre dos FB de llama a TUNING_C (tiempo de la interrupción cíclica).

Ejemplo de sintonización de un PID en un proceso continuo

En este ejemplo se ha llamado "TUNING_C " el sintonizador de parámetros, el" CONT_C "controlador integrado en el STEP 7 y el "PROC_C" el proceso.

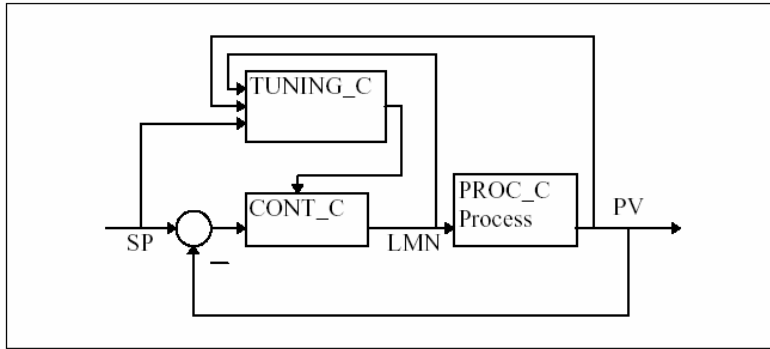


Figura 36: Ejemplo de utilización de la función Tuning_C.

El proceso está formado por una combinación de tres procesos de primer orden que constituyen un proceso de tercer orden, se ha añadido una variable perturbadora (DISV) y una ganancia estática (GAIN).

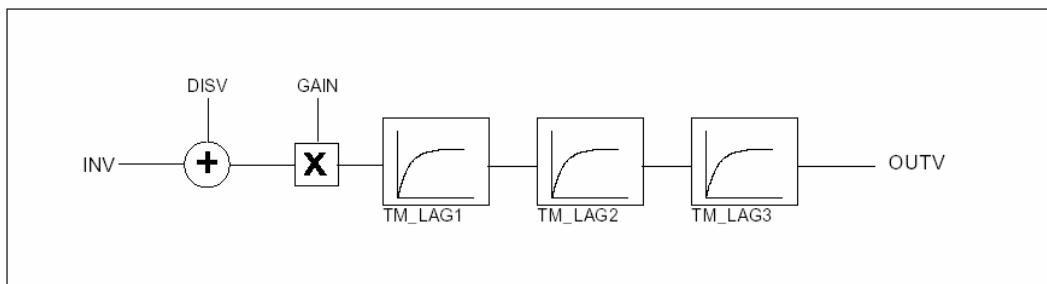


Figura 37: ejemplo de proceso de control.

Ajuste inicial

Eje límite superior de punto de ajuste, la variable del proceso y manipulado valor	100
Eje Y el nivel más bajo de punto de ajuste, la variable del proceso y manipulado valor	0
Medición ciclo	500 ms
Longitud de la línea de tiempo	200 s

PID Self-Tuner “FB_TUN_EC” [7]

Es una nueva versión de auto-sintonía de parámetros de un controlador PID, incluye algunas mejoras respecto a la función vista anteriormente “TUNING_C”, pero en lo fundamental se comporta exactamente igual que la anterior.

Antes de comenzar a utilizar la función de adaptación primero hay que dejar que el proceso llegue a un estado de equilibrio.

El inicio del proceso se establece con la activación de la variable ADAPT1ST=TRUE. A continuación, se introduce un nuevo setpoint, punto de referencia de paso positivo (nuevo punto de ajuste es mayor que el anterior) en la entrada de la SP-Auto Tuner. Hay que tener en cuenta que el salto al nuevo setpoint requiere salto mínimo definido en MIN_STEP. Se ajusta el control al modo automático (MAN_ON del Ser-Tuner = FALSO). Y comienza la optimización de controlador del auto-Tuner. Si se observa el estado en el parámetro de fase. Una vez concluido el proceso la fase vuelve al estado de trabajo (FASE = 4), se puede evaluar el diagnóstico con el parámetro STATUS_H. Si la adaptación se realiza correctamente se puede probar la respuesta de control del primer ajuste para pequeños cambios de setpoint.

Si el ajuste obtenido es el deseado, pueden guardarse los parámetros del controlador con SAVE_PAR.

Si es necesario, puede comenzar a mejorar aún más mediante el uso de ADAPT_ON. Compruebe en qué casos la respuesta a los cambios de la consigna se puede mejorar:

- Con o sin control de operación de una zona o
- Con o sin estructura de selección.

Para activar el cambio de estructura de la función tiene que ser desactivado IDB_TUN_CON_P.controller.PFDB_SEL = FALSO. Si es necesario, estos bits también tienen que ponerse a cero en el reinicio de la rutina FB_CON_P.

Si la optimización se realiza correctamente, puede utilizar UNDO_PAR para volver a cargar los parámetros del controlador anterior. Estos se activan automáticamente.

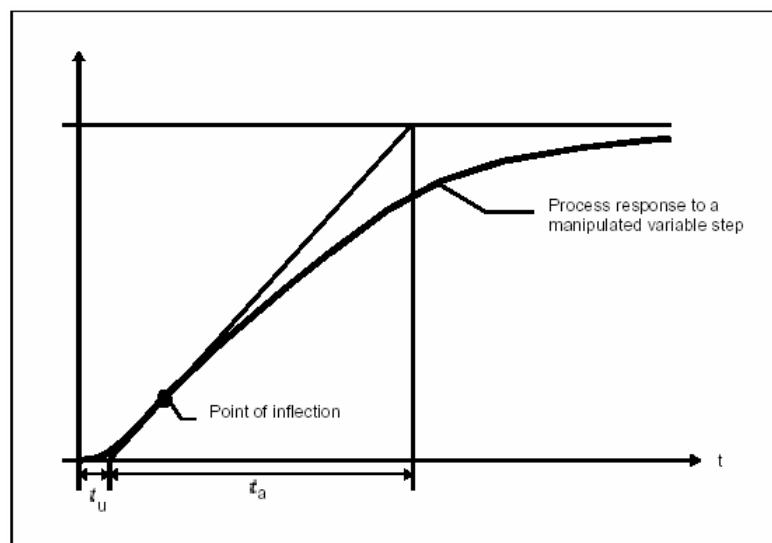
Si en la optimización no se produce el controlador de parámetros satisfactorios, se pueden cambiar a mano y activarlos mediante el establecimiento de LOAD_PAR.

Ahora puede quitar todas las partes para la simulación del proceso (FB_PROC_HCP y IDB_PROC_HCP) del proyecto.

Requisitos del proceso

Los requisitos son muy similares a los requisitos de la función "TUNING_C", el principal requisito es disponer de una variable controlada estable en el tiempo, de respuesta transitoria asintótica, con procesos de ganancia no demasiado elevada.

Después de un cambio en la variable manipulada, la variable del proceso debe cambiar a un nuevo estado estacionario como se muestra en la siguiente figura. Esto excluye por lo tanto los procesos que tienen una respuesta oscilante sin control, así como procesos que no son auto-regulados (integrador).



Descripción de los modos de funcionamiento FB “TUN_EC”

La interconexión de la FB TUN_EC con un controlador PID continuo dispone de más modos de funcionamiento que la anterior función “TUNING_C”, estos modos para controlar u optimizar el controlador son los siguientes:

Modo manual: El Auto-Tuner fuerza el controlador al modo manual y le transfiere los parámetros a este.

Manual / automático de control predictivo del valor manipulado: Después de que el controlador ha sido optimizado con éxito, el Ser-Tuner puede cambiar del modo manual al modo automático con una predicción de control de valor manipulado de tal manera que el proceso se ajuste al nuevo valor de consigna con rapidez. Este tipo de cambio no es suave. También puede configurar para que se produzca con un cambio suave en el Auto-Tuner.

Controlador de optimización: El proceso se identificó por primera vez en el momento de optimizar los parámetros del controlador. Un controlador está diseñado sobre la base de las características del proceso. Los parámetros del nuevo sistema se realizan sobre esta base y estos se activan sobre el controlador.

Primera adaptación: La adaptación primera sólo puede llevarse a cabo en caso de una variable manipulada con un cambio positivo. Se utiliza generalmente durante la primera puesta en marcha o en caso de un serio cambio en las características del sistema controlado.

On-line adaptación: El controlador puede ser optimizado posteriormente en línea durante un cambio positivo de la consigna, cambio del punto de trabajo o cambio de la respuesta del proceso.

Identificación de refrigeración: Para los sistemas de control que operan con dos elementos de accionamiento (por ejemplo, en caso de control de la temperatura: calefacción y refrigeración elementos de accionamiento) el Ser-Tuner determina la proporción de las ganancias del proceso bajo la influencia de un segundo elemento de accionamiento en el caso de una variable manipulada negativa (ejemplo de un control de la temperatura: bajo la influencia del enfriamiento de accionamiento del elemento).

Optimización de la respuesta a los cambios de consigna: El controlador está diseñado para responder a perturbaciones. Los parámetros se ajustarán para un sobrepaso del 10% al 20% de la altura del cambio de consigna.

La zona de control representa un recurso inicial. Con el fin de optimizar la respuesta en los grandes cambios de la consigna, el Ser-Tuner calcula una amplia zona de control durante la optimización. El controlador está activo dentro de la zona. Fuera de la zona del Ser-Tuner controla el controlador con la señal de salida mínimo ó máximo.

La segmentación de la estructura representa además un recurso, en el que se ajusta la acción proporcional (GAIN) y derivativa (TD) en el circuito de retroalimentación. Un cambio en el setpoint sólo actúa sobre la acción integral del controlador.

Si la segmentación de la estructura no es posible o si la zona de control no funciona, la siguiente función se puede activar con el fin de optimizar la respuesta a los cambios de la consigna:

Estructura de transición: Temporalmente se desactiva la acción integral o el control de predicción de la especificación de la

variable manipulada, para cambios de consigna positivos. Esta función no es efectiva cuando la zona de control se activa.

Guardar/Restaurar los parámetros del controlador: Los parámetros del controlador que se utiliza, puede salvarse antes de un cambio manual de estos, en el DB de instancia del Self-Tuner durante la optimización de los parámetros.

Cambio de los parámetros del controlador: Si se desea establecer los parámetros del controlador de forma manual, puede asignarse estos a través de parámetros especiales en el DB de instancia del Self-tuner y los transfiere al controlador por medio de esta función.

6.- Bibliografía Utilizada

- [1] K. Ogata, "Ingeniería de control Moderna", Prentice-Hall. Hispanoamericana, S.A. 1998, ISBN 970-17-0048-1
- [2] B. C. Kuo, "Sistemas de Control Automático", Prentice-Hall. Hispanoamericana, S.A. 1996, ISBN 968-880-723-0
- [3] E. G. Rueda, "Programación de Autómatas SIMATIC S7-300 lenguaje AWL", Ediciones CEYSA. 2006, ISBN 84-86108-51-9
- [4] Siemens, "Manual de Referencia, Listas de instrucciones AWL", 2006, Siemens.
- [5] Siemens, "Step 7: de S5 a S7 Guía para facilitar la transición", 2004, Siemens. 6ES7810-4CA07-8DW0
- [6] College of Engineering, University of Michigan, Control Tutorials for MatLab, <http://www.engin.umich.edu>
- [7] SIMATIC, PID Self-Tuner V5 Manual, 2007, Siemens. 6ES7860-4AA01-0YX0.
- [8] SIMATIC, Programar con Step 7 V5.3, 2004, Siemens. 6ES7810-4CA07-8DW0.
- [9] Step 7 Programación Avanzada, José Martínez Torres, Valencia 1999.