

GRADO EN INGENIERIA EN TECNOLOGIA INDUSTRIAL

TRABAJO FIN DE GRADO

IDENTIFICACIÓN Y CONTROL DE UN SISTEMA DE TANQUES EN SERIE

Alumno/Alumna: Casado de Diego, Leire

Director/Directora (1): Iriondo Urbistazu, Nagore

Curso: 2017-2018

Fecha: Lunes, 16 de Julio de 2018

[ES] RESUMEN

Este trabajo consiste en el diseño de un sistema de control de la maqueta real de tanques en serie Quanser. En la fase de modelado se obtiene el modelo de comportamiento del proceso a controlar que permitirá diseñar analíticamente controladores de realimentación. En la parte de validación, se cerrará el lazo de control implementando el controlador diseñado en un PLC (Programmable Logic Controller) .

Tanto en la fase de identificación de modelo como en la de control directo, se utilizarán equipos y herramientas virtuales profesionales de la ingeniería.

[EN] ABSTRACT

This project shows the control system design of real Quanser serie tank model. In modelling phase, it is obtained a behaviour model of the process to control which will allow analytically design feedback controllers. In validation phase, control loop will close implementing the controller designed in PLC (Programmable Logic Controller).

In model identification phase as well as in direct control phase, it will use professional virtual tools and equipments of engineering.

[EU] LABURPENA

Lan honen helburua da tanga-sistema baten maketa erreal batentzako kontrol-sistema diseinatzea. Erreduztapen-fasean kontrolatu beharreko prozesuaren portaera-eredua lortuko da, berrelikadura-bidezko kontrolagailuak diseinatzeko abiapuntu gisa erabiliko dena. Diseinua baliozkotzeko, kontrol zuzena burutuko da, diseinatutako kontrolagailua PLC (Programmable Logic Controller) batean inplementatuz.

ÍNDICE DE CONTENIDO

1.	INTRODUCCIÓN	8
2.	CONTEXTO	9
3.	OBJETIVO Y ALCANCE DEL TRABAJO	11
4.	BENEFICIOS QUE APORTA EL TRABAJO	14
5.	ANÁLISIS DE ALTERNATIVAS	15
5.1.	ALTERNATIVAS METODOLÓGICAS	15
5.1.1.	Fase de identificación	15
5.1.2.	Diseño de controlador	16
5.1.3.	Controladores PID.....	16
5.2.	ALTERNATIVAS HARDWARE Y SOFTWARE.....	18
5.3.	SELECCIÓN/DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	20
5.3.1.	Metodología	20
5.3.2.	Arquitectura hardware y software	20
6.	METODOLOGIA SEGUIDA EN EL DESARROLLO DEL PROYECTO	25
6.1.	Descripción de la maqueta	25
6.2.	Configuración hardware	26
6.3.	Configuración software.....	29
6.3.1.	Modelado	30
6.3.2.	Control.....	34
7.	EJEMPLO DE APLICACIÓN	37
7.1.	Fase de modelado.....	37
7.2.	Diseño de controlador.....	41
7.3.	Control directo.....	42
8.	PRESUPUESTO.....	46

9. PLANIFICACIÓN DE TAREAS	48
10. CONCLUSIONES.....	52
11. FUENTES DE INFORMACIÓN	53
ANEXO 1: CÓDIGOS DEL PROGRAMA EN TIA PORTAL.....	54
CODIGO DEL PROGRAMA DEL EXPERIMENTO EN BUCLE ABIERTO.....	54
CODIGO DEL PROGRAMA DEL EXPERIMENTO EN BUCLE CERRADO	56

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Hardware usado	10
Ilustración 2. Maqueta y amplificador de potencia usados	11
Ilustración 3. Señales del sistema	11
Ilustración 4. Variables del sistema.....	12
Ilustración 5. Esquema simple bucle abierto	13
Ilustración 6. Esquema simple del sistema de control realimentado	13
Ilustración 7. Esquema PID no interactivo	17
Ilustración 8. Ejemplo de PC convencional y tarjeta de adquisición de datos.....	18
Ilustración 9. PLC (Programmable Logic Controller) SIMATIC S7-1500.....	19
Ilustración 10. Ejemplo variedad de marcas	19
Ilustración 11. PLC SIMATIC s7-1500 y maqueta.....	21
Ilustración 12. PC Dell Optiplex usado	22
Ilustración 13. Comparación norma IEC 61131-3 / STEP 7	23
Ilustración 14. Maqueta de tanques en serie utilizada	26
Ilustración 15. Módulos PLC.....	27
Ilustración 16. Posibles rangos salidas analógicas	27
Ilustración 17. Posibles rangos entradas analógicas.....	28
Ilustración 18. Configuración #2	29
Ilustración 19. Potenciómetros de calibración (23-24-25-26)	30
Ilustración 20. Proceso simplificado.....	30
Ilustración 21. Secuencia de llamadas entre bloques de programa.	31
Ilustración 22. Esquema experimento en bucle abierto	31
Ilustración 23. Parámetros entrada-salida FB bucle abierto.....	31
Ilustración 24. Variación tensión bomba respecto al tiempo.	32

Ilustración 25 . Función SCALE y UNSCALE.....	33
Ilustración 26. Secuencia de llamadas entre bloques	34
Ilustración 27. Esquema experimento en bucle cerrado	35
Ilustración 28. Parámetro entrada-salida FB bucle cerrado	35
Ilustración 29. Variables del sistema.....	37
Ilustración 30. OB30 experimento en bucle abierto.	38
Ilustración 31. Configuración OB30 bucle abierto.	38
Ilustración 32. Configuración Trace modelado	39
Ilustración 33. Respuesta de un sistema de primer orden a una variación en la entrada.....	40
Ilustración 34. Método de cancelación polo/cero	42
Ilustración 35. OB30 experimento bucle cerrado.	43
Ilustración 36. Configuración OB30 bucle cerrado.	43
Ilustración 37. Configuración TRACE control directo.	44
Ilustración 38. Tabla presupuesto	46

ÍNDICE DE GRÁFICOS

Gráfico 1. Representación altura H1	40
Gráfico 2. Representación TRACE de la altura a controlar.....	45
Gráfico 3. Porcentajes presupuesto	47

LISTA DE ACRÓNIMOS

DB Data Block

OB Organization Block

FB Function Block

FC Function

PC Personal Computer

PLC Programmable Logic Controller

TIA Portal Totally Integrated Automation Portal

UPV Universidad del País Vasco

DISA Departamento de Ingeniería de Sistemas y Automática

POMTM Primer Orden Más Tiempo Muerto

PO Punto de Operación

SP Set Point

1. INTRODUCCIÓN

Una de las limitaciones más fuertes con la que nos encontramos en los laboratorios son las limitaciones de espacio físico, así como el presupuesto o el personal especializado necesario para obtener, implementar y operar en dichos laboratorios basados en plantas piloto. Es por ello que cada vez se está optando más por desarrollar e implementar soluciones basadas en herramientas virtuales.

La industria continúa evolucionando y lleva una tendencia a automatizar todos aquellos procesos que resulten repetitivos en diversos procesos. Es por ello que se exige más control a las máquinas que actúan sobre los procesos industriales, incluso, introduciendo cada vez más la inteligencia artificial.

Por otra parte, para optimizar al máximo un proceso se busca obtener, no solo su máximo rendimiento, sino condiciones de seguridad óptimas para el trabajo ajustándose económicamente a lo estrictamente necesario, ya que como se dice en el mundo de la industria “cada minuto cuenta”. Es esta la razón por la cual en cada proceso se tiene en cuenta cada segundo de respuesta llegando a medir los tiempos en escalas de nanosegundos.

En lo que se refiere al diseño de sistemas de control, conocer y caracterizar distintas situaciones de funcionamiento que tiene el sistema real, es estrictamente necesario para llevar a cabo un diseño óptimo. En este sentido, cuando se trabaja con maquetas, es muy favorable disponer de modelos matemáticos que reproduzcan el comportamiento real de la misma para poder diseñar el sistema de control, y prever de antemano cual será la respuesta del sistema.

En este trabajo, se ha realizado un sistema de control de una maqueta real de tanques en serie, y se ha implementado el control en un equipo industrial profesional, lo que ha permitido, además de aplicar los conceptos básicos de automática y automatización adquiridos, conocer aspectos prácticos de su implementación así como la instrumentación necesaria.

2. CONTEXTO

Atendiendo a un marco teórico, en el diseño de un sistema de control hay que seguir varios pasos:

- En primer lugar se lleva a cabo **la definición del proceso a controlar**, en este caso, el que tiene lugar en la maqueta real de tanques en serie.
Esta definición consiste en conocer los sensores, actuadores, dinámica y configuraciones con los que se va a trabajar.
- Una vez conocido el proceso, el siguiente paso es **identificar un modelo** de comportamiento del mismo.
Para ello se seleccionan las variables significativas y se lleva a cabo un experimento que permita obtener el modelo matemático equivalente aproximado. La identificación del modelo se puede realizar analítica o experimentalmente.
El modelo obtenido no sustituye al proceso, sino que será la base para el diseño del controlador. Es por ello, que se suelen utilizar aproximaciones sencillas con el fin de que sean manejables.
- El **diseño del controlador** implica la elección del tipo de controlador y la selección de los valores que se dará a sus parámetros.
Hay diferentes métodos de diseño y sintonización de los parámetros: analíticamente, experimentalmente o autosintonización.
La validación del control se realiza en simulación previa a la implementación.
- El último paso a realizar, es el **control en tiempo real**.
Este control consiste en llevar a cabo un experimento en bucle cerrado con realimentación unitaria, para comprobar el comportamiento del controlador, y ver como la variable controlada elegida sigue a la consigna como se quiere, gracias a la manipulación de la variable manipulada seleccionada.

En este trabajo se ha utilizado equipamiento de control de Siemens, ya que son las herramientas presentes en el Departamento de Ingeniería de Sistemas y

Automatización (DISA) de la “Escuela de Ingeniería de Bilbao” en la cual se ha desarrollado el proyecto. Esta razón, junto con que en las asignaturas desarrolladas durante el grado se han usado dichas herramientas de Siemens, constituyen razones suficientes para su elección.

El entorno de desarrollo de proyectos de automatización de Siemens es TIA Portal (Totally Integrated Automation Portal)

Para actuar sobre la maqueta de tanques se elige la herramienta de ingeniería TIA Portal (Totally Integrated Automation Portal), una de las más usadas en el mundo de la industria. Esta herramienta utiliza el lenguaje STEP 7 basado en la norma IEC 61131-3.

La norma IEC 61131-3 establece los lenguajes estandarizados, lo que permite estandarizar cada vez más el modelo software en usos industriales. Este hecho facilita mucho la reutilización de los programas entre distintos dispositivos de distintos fabricantes, queriéndose llegar a una unificación total del modelo software.

Por tanto, el equipamiento utilizado es:

- PLC (Programmable Logic Controller) SIMATIC S7-1500, con TIA Portal V13.
- PC Dell Optiplex, en el que se ejecuta el TIA Portal y que se comunica vía Ethernet con el PLC.
- Maqueta de tanques en serie QUANSER y amplificador de potencia (UPM).



Ilustración 1. Hardware usado

3. OBJETIVO Y ALCANCE DEL TRABAJO

La motivación de este proyecto parte de la necesidad de implementar una herramienta de diseño de sistemas de control para controlar el sistema real de tanques presente en el laboratorio de automática y control de la universidad (maqueta de tanques configurados en serie), utilizando el hardware de control siemens SIMATIC S7-1500 y su entorno de desarrollo: TIA Portal V13.

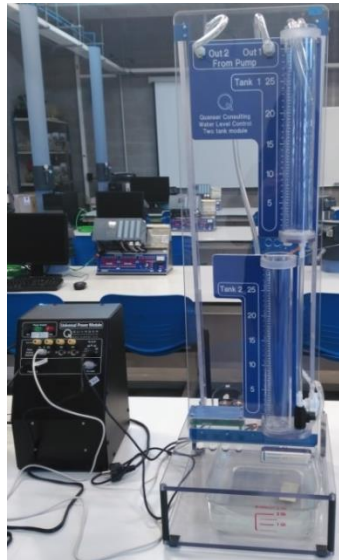


Ilustración 2. Maqueta y amplificador de potencia usados

En primer lugar, se estudian las señales de entrada y salida que tiene la planta con objeto de poder actuar sobre las entradas y analizar el efecto que producen sobre las salidas.



Ilustración 3. Señales del sistema

En este caso, la única señal de entrada que existe es la de la bomba de impulsión que dirige el agua hacia los tubos que alimentan la planta. En cuanto a las señales de salida de las cuales se estudian sus cambios, tenemos, tanto la altura del tanque 1 (h_1), como la altura del tanque 2 (h_2).

La variable manipulada por tanto es el caudal de la bomba (Q), y la variable controlada es la altura de los tanques (H).

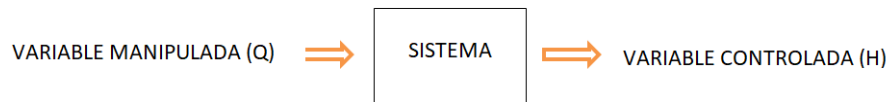


Ilustración 4. Variables del sistema

Para llevar a cabo los procesos experimentales es necesario realizar previamente la conexión adecuada de la maqueta, así como, la calibración de los sensores los cuales, como se detalla más adelante, permiten obtener los valores de las alturas de los tanques.

Una vez establecidas las distintas variables tanto de salida como de entrada, y de tener todo perfectamente conectado y calibrado se crean una serie de bloques funcionales (FBs) para el estudio en bucle abierto de los distintos cambios que sufren las salidas en función de las variaciones introducidas a la entrada. Como se ha explicado anteriormente, el método elegido para realizar dichas funciones y bloques funcionales, es a través de un PLC de Siemens SIMATIC S7-1500, implementándolos con el programa *Tia Portal*.

Para obtener el modelo de comportamiento de la planta se realiza el experimento en **bucle abierto**. La función de transferencia de la planta se obtiene llevando al tanque que se quiere identificar hasta un estacionario (punto de operación), y una vez estabilizado, introducir una variación en la entrada hasta que se llegue a un nuevo punto de trabajo estacionario.

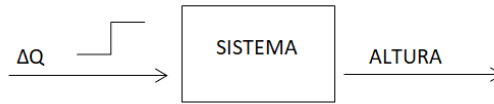


Ilustración 5. Esquema simple bucle abierto

La variación introducida a la bomba es un escalón, y se registra cuál es el comportamiento de las variables de salida (h_1 y h_2), variables medidas a través de dos sensores de presión que se encuentran en el fondo de sendos tanques, y cuyo valor es extrapolado para obtener los distintos valores de altura que existen.

Este modelo permite diseñar el controlador, en este caso un PID debido a que es uno de los más usados industrialmente. El controlador diseñado analíticamente se implementa en el PLC (Programmable Logic Controller) que cierra el lazo de control de nivel llevándose a cabo el experimento en **bucle cerrado**.

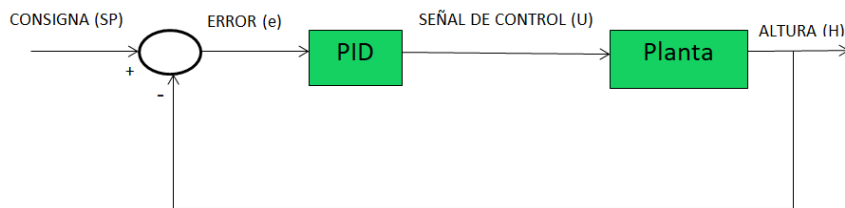


Ilustración 6. Esquema simple del sistema de control realimentado

Primeramente es necesario realizar un estudio de los distintos PID's que oferta Siemens a través del TIA Portal y que podrán ser usados para el control de la planta en bucle cerrado. En este caso, en vez de usar controladores PID ya existentes, se crea uno de usuario implementado a través de su ecuación en diferencias. Los valores de las constantes del controlador (K , T_d , T_i) se eligen partiendo del modelo en función de transferencia de la planta de tanques obtenida en el experimento en bucle abierto y de las características del sistema que se deseen.

4. BENEFICIOS QUE APORTA EL TRABAJO

En este apartado se va a proceder a hablar de los múltiples beneficios que aporta este trabajo.

Entre los primeros se encuentra el hecho de probar la validez del **método de identificación** usado en este proyecto. Para el desarrollo del mismo, como ya se sabe, se ha usado la identificación experimental, que permite realizar un análisis del modelo para representar el comportamiento de la planta (en este caso la maqueta de tanques en serie), a través de un modelo lo más cercano posible al comportamiento real de la misma.

Como es de esperar, el llegar hasta un modelo identificativo de la planta es muy útil a la hora de diseñar un sistema de control en procesos industriales. Este hecho permite, sin contar con la propia maqueta de tanques, **simular el proceso** y conocer los resultados que se obtendrán al ponerlo en funcionamiento.

Otro beneficio que se obtiene de este trabajo es el profundo aprendizaje sobre el Tia portal y las múltiples opciones que este ofrece y que pueden ser usadas de manera muy positiva, así como el conocimiento del código de lenguaje usado que abre muchas puertas en el mundo de la programación.

Uno de los aspectos que más se ha tenido en cuenta a la hora de desarrollar el programa de identificación y control es su posible **reutilización** en futuros proyectos. Esto significa que se crean bloques funcionales con parámetros formales que se pueden instanciar tantas veces como sea necesario. Con ello se pretende que el sistema de control pueda extenderse con nuevas finalidades con la mínima dificultad gracias también a la documentación creada en base a los programas desarrollados

Por último, podría ser incluido dentro de este apartado el hecho de que, para la docencia del centro, puede resultar beneficioso el haber probado esta identificación experimental con sistemas que se encuentran en la propia escuela para incluir en sus programas de enseñanza un apartado dedicado a identificar modelos de planta experimentalmente, basándose un poco en el desarrollo de este proyecto.

5. ANÁLISIS DE ALTERNATIVAS

En este apartado se van a presentar las distintas alternativas que aparecen a la hora de realizar la identificación y control de los tanques en serie. A lo largo de todo el documento se han ido detallando los diferentes objetivos fijados en el proyecto y la metodología utilizada para llevarlos a cabo. Es ahora por tanto, cuando se va a detallar cuales podrían ser las diferentes alternativas para realizarlo, y se van a justificar las diversas razones por las cuales se ha decidido llevarlo a cabo de la manera especificada: identificación experimental con un PLC 'Programmable Logic Controller' y el software 'Tia Portal'.

A la hora de realizar este análisis de alternativas existen tres tipos de vías de clasificación: alternativas metodológicas, alternativas hardware y alternativas software. A continuación se detallan cada una de ellas.

5.1. ALTERNATIVAS METODOLÓGICAS

En el diseño del sistema de control aparecen dos fases principales: fase de identificación y diseño del controlador.

5.1.1. Fase de identificación

Por un lado en la fase de identificación o modelado el objetivo es diseñar el controlador para lo cual se utilizan modelos simples de primer orden, segundo orden y primer orden más tiempo muerto (POMTM). Esta fase se puede realizar tanto analíticamente como experimentalmente.

- En el caso de elegir **modelado teórico**, es necesario conocer a la perfección el proceso así como aplicar las leyes de conservación de la energía y de la masa, y la ley de cantidad de movimiento.
- En el caso del **modelado experimental** se aplican entradas al proceso real (variable manipulada) y se registran las respuestas obtenidas (variable controlada). A partir de estos datos, se lleva a cabo la identificación de la

planta mediante aproximación a los modelos simples conocidos, obteniéndose la función de transferencia del sistema que será usada en el diseño del controlador.

5.1.2. Diseño de controlador

Por otro lado está el diseño del controlador, cuyo objetivo es hallar el PID ideal. Esta fase existe la posibilidad de realizarla experimentalmente, analíticamente o con autosintonía.

- En el caso del **diseño experimental**, es necesario obtener un modelo de primer orden más tiempo muerto (POMTM), a partir del cual entrar en las tablas de sintonización para ir hallando los valores K, Ti y Td del controlador.
- El **diseño analítico** consiste en la aplicación de diversas fórmulas para la obtención K, Ti y Td necesarios para sintonizar el PID.
- En el caso del método de la **autosintonía**, como el propio nombre indica, la sintonización se realiza automáticamente.

5.1.3. Controladores PID

El entorno de desarrollo TIA Portal ofrece una serie de PID's que pueden ser usados a la hora de diseñar un sistema de control: PID_Compact, PID_3Step y PID_Temp. En este caso no será usado ninguno de ellos, ya que se decide crear un PID de usuario implementado a través de su ecuación en diferencias.

- **PID Compact:** La instrucción PID_Compact ofrece un regulador PID con optimización integrada para actuadores de acción proporcional. Su llamada tiene que realizarse en una base de tiempo constante de un OB de alarma cíclica, y se deben parametrizar directamente en el DB multiinstancia, poniéndose en servicio a través de una tabla de observación.
- **PID 3Step:** Al igual que en el PID anterior, el PID_3Step se usa como regulador. Sus parámetros P, I y D son seleccionados automáticamente.

- PID Temp:** Este objeto tecnológico proporciona reguladores PID continuos con optimización integrada.

 PID_Temp está especialmente diseñado para la regulación de la temperatura y puede aplicarse a sistemas de calefacción y refrigeración. Este hecho hace que no sea interesante en este proyecto ya que no se trabaja con variable de temperatura.
- PID de usuario:** El algoritmo de control usado para la creación del PID de usuario es no interactivo. Este hecho quiere decir que las acciones Integral y derivativa son independientes.

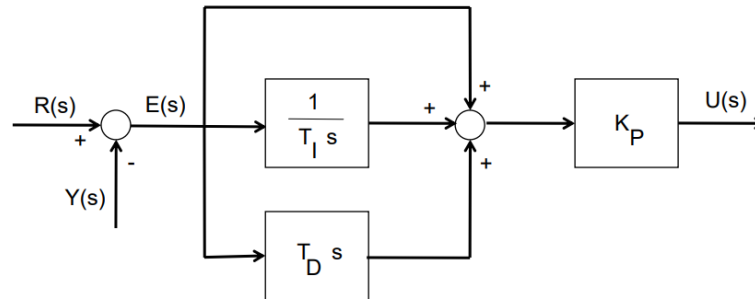


Ilustración 7. Esquema PID no interactivo

La ecuación de un PID clásico es la mostrada a continuación:

$$U(t) = Kp \left(e(t) + \frac{1}{Ti} \int e(t)dt + Td \frac{de(t)}{dt} \right)$$

En cuanto a la ecuación del PID no interactivo, es la mostrada a continuación. En ella se puede ver una constante 'N' que aparece dividiendo en la acción derivativa. Esta constante sirve para filtrar la acción derivativa ideal por un sistema de primer orden, debido a que presentan ganancias muy elevadas a altas frecuencias. Los valores que se le suelen dar se encuentran entre 2 y 20.

$$U(s) = Kc \left(1 + \frac{1}{Ti s} + \frac{Td s}{1 + \frac{Td s}{N}} \right) E(s)$$

Para implementar este PID se hace a través de su ecuación en diferencias:

$$U_p(k) = K_c e_p(k)$$

$$U_i(k) = U_i(k-1) + K_c \frac{T}{2T_i} [e_i(k) + e_i(k-1)]$$

$$U_d(k) = \frac{U_d(k-1)T_d + K_c T_d N [e_d(k) - e_d(k-1)]}{(NT + T_d)}$$

$$U(k) = U_p(k) + U_i(k) + U_d(k)$$

5.2.ALTERNATIVAS HARDWARE Y SOFTWARE

En cuanto a las alternativas hardware y software, existen diversas formas de llevar a cabo la identificación y control de la maqueta. Además, cabe decir que la elección software viene ligada a la arquitectura hardware usada

Por un lado se encuentra la posibilidad de realizarlo a través de un **PC 'convencional'**. En este caso es necesaria la utilización de tarjetas de adquisición de datos compatibles con el bus del PC con el que se quiera trabajar para la conversión de señales de analógicas a digitales y viceversa, además de un software de simulación como puede ser Matlab/Simulink, y un toolbox de conexión de las señales con el sistema de control como puede ser RTW (Real Time Window).



Ilustración 8. Ejemplo de PC convencional y tarjeta de adquisición de datos

Por otro lado, se encuentra la alternativa de usar un **Programmable Logic Controller 'PLC'**. Este ya tiene incorporadas las tarjetas analógicas tanto de entradas como de salidas con las cuales se va a trabajar, constituyendo una herramienta más compacta. Aparte del PLC es necesaria la utilización de un PC para el desarrollo del programa.



Ilustración 9. PLC (Programmable Logic Controller) SIMATIC S7-1500

Como es de esperar, existen gran variedad de marcas que ofertan programas con los cuales crear el código para poder identificar y controlar un proceso. Asimismo cabe destacar que, al poder trabajar con distintas marcas, es necesario comprobar la compatibilidad de fabricantes existente entre software y hardware. Es por ello, que conviene utilizar todo el material del mismo fabricante para evitar problemas.



Ilustración 10. Ejemplo variedad de marcas

Como ya se ha dicho en el apartado anterior, en el caso de usar un PC convencional es necesario el uso de un software de simulación como puede ser matlab/simulink.

En el caso de usar un PLC, también es necesario el uso de un software para realizar el código del sistema de control. Dicho software necesariamente ha de ser compatible con el PLC empleado.

5.3. SELECCIÓN/DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Ante el estudio de las distintas alternativas, para llevar a cabo el proyecto, se ha elegido realizarlo con un *PLC de Siemens s7-1500* y con el software Tia Portal V13. La fase de modelado se realizara de forma experimental, y el diseño del controlador de modo analítico.

A continuación se exponen las razones por las cuales se han tomado dichas decisiones.

5.3.1. Metodología

Como bien se ha dicho, la fase de **modelado** se lleva a cabo experimentalmente. Esta decisión deriva del interés por obtener la función de transferencia de la planta llevando a cabo diversos experimentos y pudiéndose hacer uso de la maqueta presente en el laboratorio de Automática y Control. Este hecho además permite adquirir conocimientos sobre cómo enfrentarse a una situación real y los problemas que esto conlleva.

A diferencia del modelado, el **diseño del controlador** se realiza de manera analítica. En este caso la decisión se lleva a cabo por descarte de las demás opciones. La opción de autosintonía no interesa por realizar la sintonización del controlador automáticamente, y la opción experimental por la necesidad de un POMTM (Primer Orden Más Tiempo Muerto) y tener un Primer orden.

5.3.2. Arquitectura hardware y software

En cuanto al hardware usado, se opta por un **PLC SIMATIC S7-1500** de la marca Siemens.

El PLC es un equipo electrónico inteligente diseñado en base a microprocesadores, que consta de unidades o módulos que cumplen funciones específicas, tales como, una unidad central de procesamiento (CPU) que se encarga de casi todo el control del

sistema, módulos que permiten recibir información de todos los sensores y comandar todos los actuadores del sistema .



Ilustración 11. PLC SIMATIC s7-1500 y maqueta.

Las **ventajas** que ofrece este método frente a otros posibles son:

- Por un lado, los menores costos, menor consumo de energía y la menor ocupación de espacio que suponen respecto a la lógica convencional ya que es un tablero de gobierno mucho más compacto, sin necesidad de dispositivos electromecánicos y electrónicos.
- Además son dispositivos con alta fiabilidad ya que su fallo es muy improbable; con alta versatilidad ya que es posible modificar el funcionamiento de un sistema automático con solo realizar un nuevo programa y cambios mínimos en el cableado; y, con poco mantenimiento, siendo solo necesario mantenerlo limpio y con sus terminales ajustados a los bornes y puesta a tierra.
- Por otro lado un PLC es de fácil instalación con cableados iguales tanto para señales de entrada como de salida, y de fácil programación ya que no es necesario poseer conocimientos avanzados en el manejo del ordenador (PC), sino solamente unos conceptos básicos son suficientes. Una de las ventajas

más destacadas del PLC podría ser la detección de fallos que ofrece y que muestra al usuario a través de unos leds indicadores de diagnóstico, que indican el estado de la CPU, la batería, los terminales de entrada/salida, y otros implementos.

En cuanto a sus **desventajas** podría decirse que:

- el lugar de la instalación debe ser elegido a conciencia ya que son necesarios una serie de requisitos para su correcto funcionamiento como son: nivel de temperatura, humedad, ruido, variaciones de tensión, etc; aunque es bastante sencillo encontrar una planta adecuada donde instalarlo, y en este caso se contaba con ella.

Por otro lado, dentro de la arquitectura hardware, también es necesario un PC para el desarrollo del programa.

En este caso el PC usado es el presente en el laboratorio de desarrollo del proyecto:

- PC Dell Optiplex.



Ilustración 12. PC Dell Optiplex usado

En cuanto a la arquitectura software, el entorno de desarrollo de Siemens es TIA Portal.

Dicho programa se encuentra instalado en el PC con el que se va a trabajar y el cual constituye la interfaz gráfica con el usuario.

Por otro lado, el lenguaje de programación que usa TIA Portal, conocido como STEP 7, ha sido desarrollado y trabajado durante el presente curso y está basado en la norma IEC 61131-3 la cual establece los lenguajes estandarizados.

Lenguajes estandarizados en IEC 61131-3 / STEP7

Lenguaje	IEC 61131-3	STEP 7
Diagrama de funciones	FBD <i>Function Block Diagram</i>	FBD/FUP Function Block Diagram
Diagrama de contactos	LD <i>Ladder Diagram</i>	LAD/KOP Ladder Logic
Lista de instrucciones	IL <i>Instruction List</i>	STL/AWL Statement List
Texto estructurado	ST <i>Structured Text</i>	SCL Structured Control Language
Gráfico secuencial	SFC <i>Sequential Function Chart</i>	GRAPH7
Diagrama de bloques		CFC Continuous Function Chart
Diagrama de gráfico de estados		HIGRAPH7

Ilustración 13. Comparación norma IEC 61131-3 / STEP 7

Entre las ventajas que ofrece el lenguaje STEP 7, el cual permite estructurar el programa, son:

- Los programas pueden dividirse en bloques, entre los que se encuentran:
 - ◆ **módulos de organización (OB):** en función de su numeración son de distintos tipos (ejecución cíclica, alarmas horarias, errores...). En este proyecto se usa el OB1 de ejecución cíclica y el OB30 de alarma cíclica en el que se desarrolla el programa.
 - ◆ **módulo de función (FB):** módulos con memoria que pueden tener parámetros de entrada/salida, variables estáticas y temporales.
 - ◆ **función (FC)** módulos programables sin memoria. Devuelve siempre un valor y no soporta variables estáticas.
 - ◆ **módulo de datos (DB):** usados para almacenar datos. Los FB siempre llevan un DB asociado.
 - ◆ **función del sistema (SFC)**
 - ◆ **módulo de función del sistema (SFB).**
- Se pueden estandarizar secciones individuales del programa.

- Se simplifica la organización del programa.
- Las modificaciones del programa pueden realizarse más fácilmente.
- Se simplifica la depuración del programa, ya que puede ejecutarse por partes.
- Se simplifica la puesta en servicio.

6. METODOLOGIA SEGUIDA EN EL DESARROLLO DEL PROYECTO

En este apartado se va a desarrollar cuál es la metodología seguida para llevar a cabo el proyecto. En primer lugar se describe la maqueta que se usara para el desarrollo del trabajo, para continuar con una descripción de los pasos seguidos a la hora de realizar la identificación y control de la maqueta de tanques en serie.

6.1.Descripción de la maqueta

La maqueta a usar es la maqueta Quanser de tanques en serie presente en el laboratorio de automática y control de la universidad.

Dicha maqueta consta de dos sensores presentes en el fondo de sendos tanques, los cuales nos darán un valor de tensión en función de la altura del agua del tanque. Los voltajes de los sensores van de 0 a 5V, dato que hay que tener en cuenta a la hora de realizar la configuración hardware de nuestro sistema. Estos sensores están conectados a entradas analógicas por ser, como se ha dicho anteriormente, las variables controladas cuyo valor entra al PLC.

Además la maqueta consta de una bomba la cuál es la encargada de impulsar el agua hacia los tanques y que posee un sensor para poder conocer cuál es la tensión con la que está actuando dicha bomba. En este caso el sensor va de 0-12V y está conectado a una salida analógica por ser variable manipulada a través del PLC.

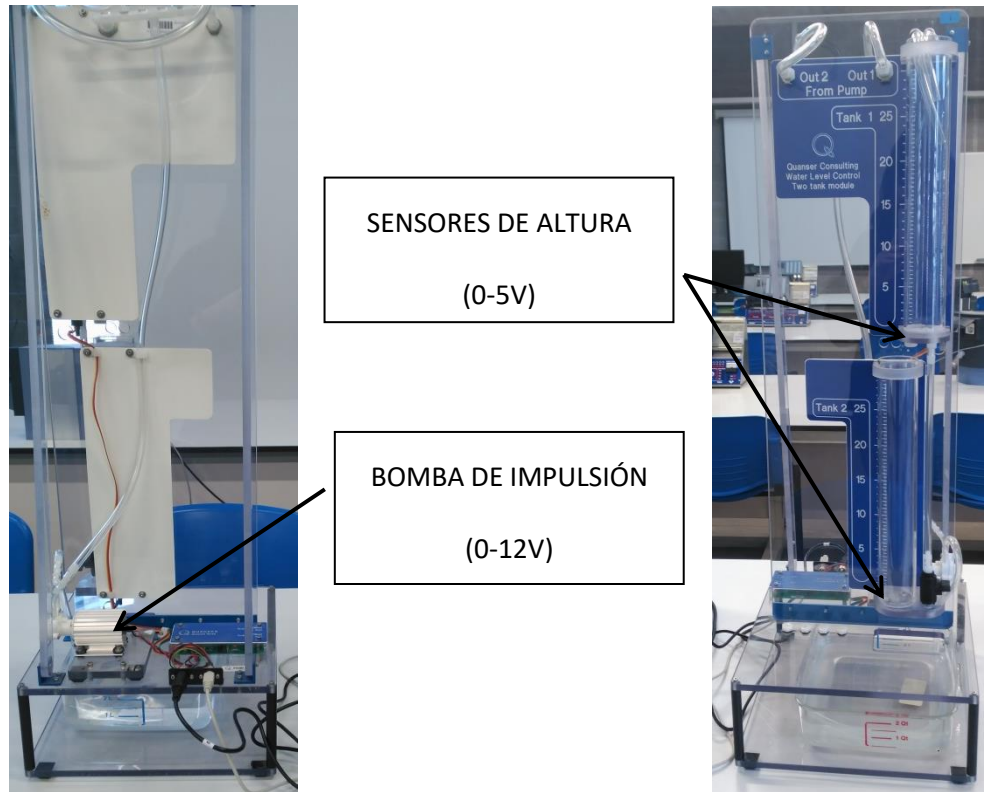


Ilustración 14. Maqueta de tanques en serie utilizada

6.2. Configuración hardware

El siguiente paso a realizar una vez identificada la instrumentación (sensores y actuadores) de la maqueta, es la configuración hardware del PLC usado para realizar el sistema de control.

Como ya se ha dicho anteriormente, el PLC está compuesto por distintos módulos cada uno con una función, y entre los que se encuentran los módulos de entrada y salida analógicas de especial interés en este proyecto, ya que son con los que se trabaja.



Ilustración 15. Módulos PLC

Los módulos 4 y 5 corresponden a las entradas y salidas analógicas respectivamente. Para su configuración, a la hora de parametrizar las entradas analógicas y las salidas analógicas, es necesario elegir el rango en el que se encuentran los sensores.

Como se puede ver en la *ilustración 16. Posibles rangos salidas analógicas*, ninguno de los rangos de los sensores de salidas analógicas coincide a la perfección con el rango real de la bomba (0-12V), por lo que se decide elegir la opción de +/-10V. Este hecho, como se explicará más adelante, lleva a la necesidad de caracterizar los sensores.

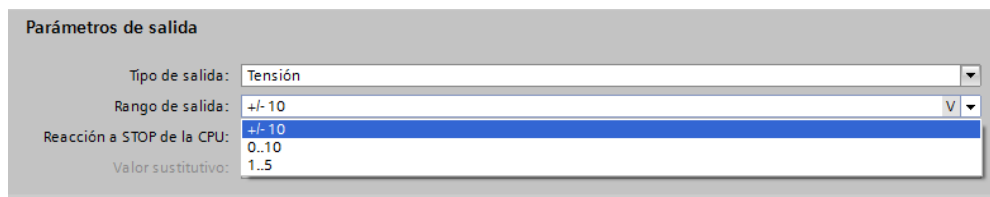


Ilustración 16. Posibles rangos salidas analógicas

En cuanto a las entradas analógicas, se ve en la *ilustración 17. Posible rangos de entradas analógicas*, como tampoco ninguno de los rango coincide a la perfección con los sensores de altura (0-5V), por lo que se decide elegir el mismo rango que para el caso de las salidas analógicas (+/-10V).



Ilustración 17. Posibles rangos entradas analógicas

El hecho de que la parametrización de sensores no coincida exactamente con la realidad lleva a introducir unos valores de caracterización de los mismos.

Hasta el momento, en ninguna de las prácticas realizadas con el Tia Portal, se había necesitado llevar a cabo este paso, ya que todas ellas eran prácticas docentes preparadas perfectamente para que esta dificultad no se diera.

En este caso, para solventar este problema, simplemente basta con dividir el rango configurado en el hardware entre el rango real de nuestro sensores, obteniéndose así el factor de caracterización de cada sensor, valor por el cual habrá que multiplicar los límites reales para que todas las medidas sean coherentes y estén relacionadas.

$$\text{factor de caracterización} = \frac{\text{rango configuración hardware}}{\text{rango real}}$$

$$\text{límite caracterizado} = \text{factor caracterización} * \text{límite real sensor}$$

6.3. Configuración software

Como resumen podría decirse que la configuración software está basada en los pasos siguientes: modelado y control, pero antes de realizar todo esto es necesario llevar a cabo la conexión de la maqueta y la calibración manual de los sensores.

Para ello:

1. Configurar la maqueta en función de la configuración #2 (descrita en el manual de usuario [6] *QUANSER Coupled Water Tanks User Manual*)

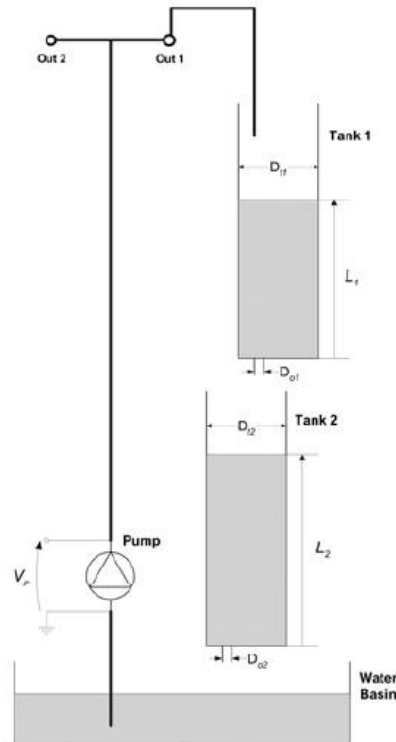


Ilustración 18. Configuración #2

2. Para comprobar el correcto funcionamiento de la bomba introducimos valores de tensión reales y vemos como se muestra la señal en la salida analógica (QW4).

3. Calibración de los sensores:

- leer valor de los sensores de alturas H1 (entrada analógica IW6) y H2 (entrada analógica IW8), y, con los tanques vacíos, regular el sensor con ayuda de un destornillador hasta que su lectura sea 0 (calibración offset).
- para calibrar la ganancia, llenar cada tanque hasta los 25 cm y mover el tornillo 'gain' hasta que los sensores lean entre 4 y 4.2 V (11059 y 11612 como valores en 16 bits). Para el llenado de tanques simplemente se introduce una señal a la bomba procedente de la entrada analógica IW4 la cual está dotada de una ruleta para poder regular su valor.

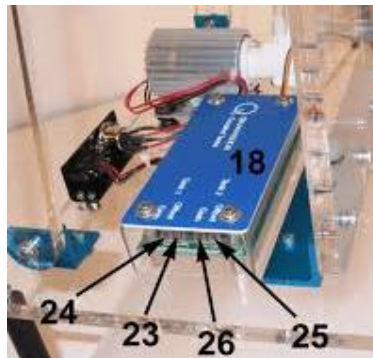


Ilustración 19. Potenciómetros de calibración (23-24-25-26)

6.3.1. Modelado

Una vez realizada la conexión y calibración de la maqueta se procede a comenzar con el modelado.



Ilustración 20. Proceso simplificado

Para ello se lleva a cabo el experimento en bucle abierto, cuyos pasos se detallan posteriormente, y para el cual se crea únicamente un FB (con su DB asociado) y un OB30 de alarma cíclica desde el cual se llama al FB Y se instancian los parámetros formales del mismo.

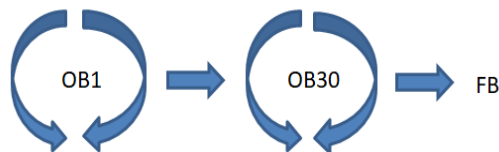


Ilustración 21. Secuencia de llamadas entre bloques de programa.

La estructura del FB sigue el siguiente esquema:

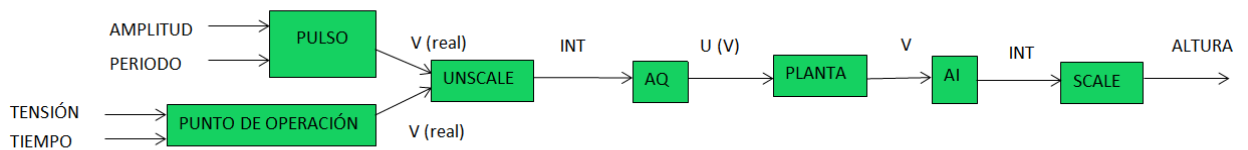


Ilustración 22. Esquema experimento en bucle abierto

y sus parámetros de entrada y salida son los mostrados a continuación:

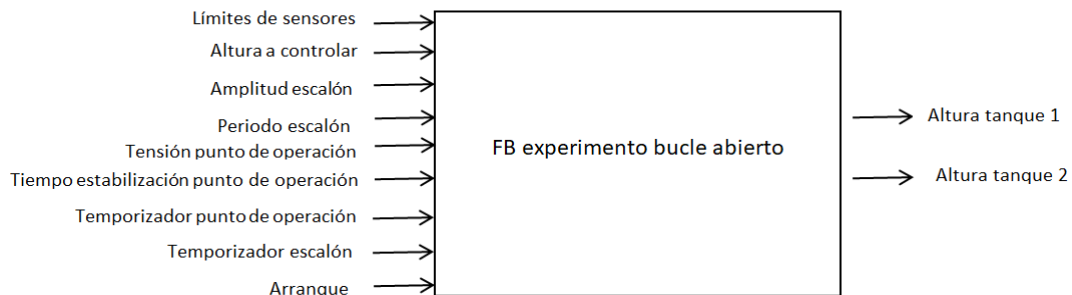


Ilustración 23. Parámetros entrada-salida FB bucle abierto.

En cuanto a las fases del modelado:

1. Establecer el tanque en un punto de operación.

Antes de introducir la variación en la señal de entrada (señal de la bomba), se decide que, para visualizar mejor dicha variación, se va a introducir primero una señal a la bomba hasta que el tanque a controlar llegue a un punto de operación determinado y a partir de ahí se introduce la variación deseada. El tiempo durante el cual se mantiene esta primera excitación es fijado experimentalmente realizando distintas pruebas hasta que se llega a un tiempo suficiente para que el tanque se estabilice.

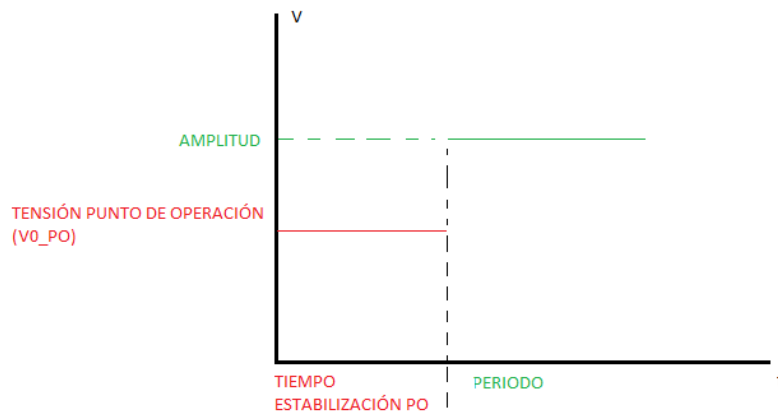


Ilustración 24. Variación tensión bomba respecto al tiempo.

2. Introducir cambio en la entrada.

Una vez que el tanque a controlar se encuentra en el punto de operación deseado, se introduce la ya citada variación en la entrada. En este caso consistirá en una onda cuadrada creada en otro bloque funcional y cuyo periodo y amplitud serán determinados por el usuario. En el caso concreto del periodo debe ser tal que permite a la altura del agua estabilizarse en un nuevo punto de equilibrio.

3. Registrar los valores de los sensores de altura extrapolando las medidas.

Los datos obtenidos a través de los distintos sensores, son leídos en el PC a través de entradas y salidas analógicas. Estos terminales obtienen el valor en binario, por lo que para poder pasarlos a un valor real es necesaria realizar una transformación.

Para llevar a cabo este cambio se usa tanto la función *SCALE* como *UNSCALE*, presentes en la biblioteca del programa y que permiten pasar de binario a número en coma flotante y viceversa.



Ilustración 25 . Función SCALE y UNSCALE

Además, en función de los límites superiores e inferiores que se introduzcan en las funciones *SCALE* y *UNSCALE*, hallaremos el valor de la medida en distintas unidades, unidades que hay que tener muy presentes a la hora de usar los datos obtenidos.

En este caso la medida de los sensores de los tanques puede darse en tensión (voltios) o en altura (centímetros), y la medida de la bomba en tensión (voltios) o en caudal (cm³/s).

4. Identificar el modelo.

Para poder identificar el modelo de la planta (objetivo de esta fase), es necesario registrar las medidas obtenidas del experimento en bucle abierto, en este caso, las alturas de los tanques.

Para llevar a cabo este registro de medidas de los sensores, se usa el comando TRACES de TIA Portal.

Este comando se puede encontrar en cada accionamiento y controlador, y sirve para medir gráficamente cualquier variable con la que estos cuenten.

El comando TRACE permite elegir el tiempo de muestreo en las representaciones y el tiempo total de la medición. Hay que tener en cuenta que la capacidad tanto en la cantidad de puntos de muestreo, como en la selección de las variables está limitada, aunque este hecho no ha supuesto ningún problema en este caso.

6.3.2. Control

Una vez que se tiene el modelo de la maqueta, se procede a llevar a cabo el control de la misma, es decir, se lleva a cabo el **experimento en bucle cerrado**.

Para ello simplemente hay que crear un bloque funcional (FB) en el cual, cerrando el lazo con una realimentación unitaria, y considerando como entradas un valor de referencia y el valor de salida de los sensores de presión, y como acción de control (u) la señal que se dirige a la bomba, conseguimos, en función del PID que posteriormente implementemos, controlar nuestra planta de la manera deseada.

El FB creado para el control de la maqueta es llamado desde el OB30 de alarma cíclica donde se instancian los parámetros formales del mismo.

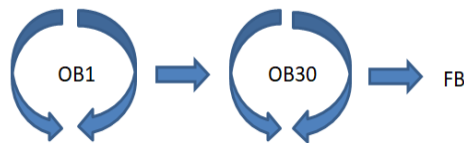


Ilustración 26. Secuencia de llamadas entre bloques .

La estructura del FB sigue el siguiente esquema:

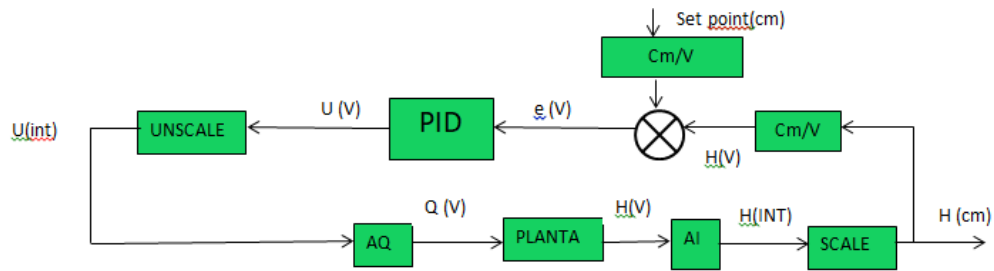


Ilustración 27. Esquema experimento en bucle cerrado

y sus parámetros de entrada y salida son los mostrados a continuación:

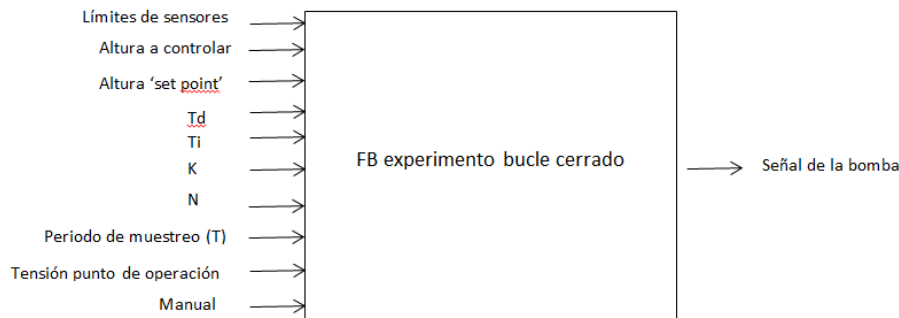


Ilustración 28. Parámetro entrada-salida FB bucle cerrado

Las distintas etapas del bucle cerrado son:

1. **Calcular el error a partir de la señal de referencia ('set point') y de la señal de lectura de uno de los sensores de altura.**

Para calcular el error simplemente se calcula la diferencia entre la señal de referencia introducida y la lectura de los sensores.

Como ya se ha explicado antes, es necesario extrapolar las medidas de lectura obtenidas en binario a valores reales y viceversa, a través de las funciones SCALE y UNSCALE explicadas con anterioridad.

2. Introducir la ecuación en diferencias del PID elegido.

Simplemente se inicializan las variables y se implementa la ecuación en diferencias del PID, cuyos valores K, Td y Ti se introducirán al instanciar los parámetros formales en el OB 30.

Se establece que el PID comience a operar cuando se activa la señal 'Manual' a través de un pulsador del PLC. Hasta que esto ocurra, la bomba estará dando una señal de tensión introducida como 'tensión punto de operación'.

3. Escribir en la bomba la acción de control necesaria.

En esta última fase simplemente se extrapola el valor calculado por la señal de control (u), para poder ser escrito en la bomba. Para ello se utilizan las funciones SCALE y UNSCALE explicadas con anterioridad.

7. EJEMPLO DE APLICACIÓN

Ahora que ya está el procedimiento perfectamente descrito, se van a mostrar los resultados obtenidos en el ejemplo de aplicación realizado.

Para estructurar dicho ejemplo, se van a diferenciar 3 fases: fase de modelado, diseño de controlador y control directo.

7.1. Fase de modelado.

1.1. Identificación de variables significativas

En primer lugar, para poder llevar a cabo el sistema de control, hay que conocer cuáles son las variables de proceso. En este caso, la variable manipulada es la tensión de la bomba, y la variable controlada la altura del tanque 1 (H1).

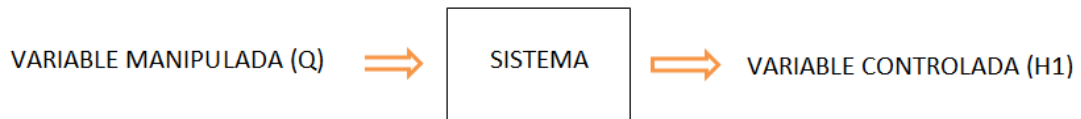


Ilustración 29. Variables del sistema.

1.2. Modelado

Una vez conocidas las variables se procede a hallar la función de transferencia del sistema.

Para ello es ejecutado el FB creado para realizar el experimento en bucle abierto. Este FB es llamado desde el OB de alarma cíclica (OB30) en el que se instancian las variables formales de entrada y salida del bloque funcional (FB).

```

//Llamada a la FB
"Experimento_BA_DB"(lim_inf_cm3s:=0.0,
                    lim_sup_cm3s:=39.6,
                    lim_inf_B_V:=0.0,
                    lim_sup_B_V:=12.0,
                    lim_inf_H_V:=0.0,
                    lim_sup_H_V:=5.0,
                    lim_inf_cm:=0.0,
                    lim_sup_cm:=25.0,
                    H_control_bin:="H1_binario",
                    Amplitud:=1.5,
                    Periodo:=S5T#40s,
                    V0_PO:=1.0,
                    tiempoestabilizacion_PO:=S5T#30s,
                    Arranque:="ARRANQUE",
                    Temporizador_PO:="temporizador_PO",
                    Temporizador_impulso:="temporizador_impulso",
                    altura1_cm=>#H1_cm,
                    altura2_cm=>#H2_cm);
  
```

Ilustración 30. OB30 experimento en bucle abierto.

El tiempo de muestreo elegido para la alarma cíclica del OB30 es de 0.1s como se muestra en la siguiente ilustración. Este valor es considerado como adecuado ya que permite tomar un valor cercano a 10 muestras por cada ciclo, como se comprueba más adelante, al hallar la constante de tiempo del sistema (τ).

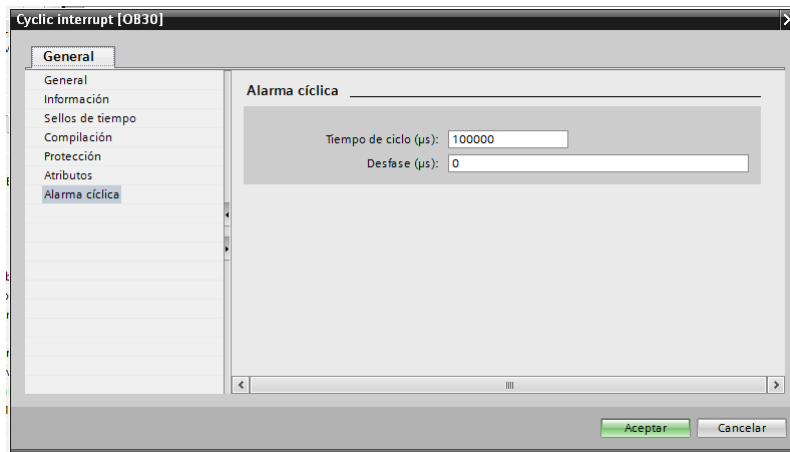


Ilustración 31. Configuración OB30 bucle abierto.

Para hallar la función de transferencia, es necesario representar la variación que sufre la altura H1 ante la variación introducida en la entrada (bomba), por lo que se crea un TRACE cuya configuración se muestra a continuación:

Configuración

Señales

	Nombre	Tipo de d...	Dirección	Color	Comentario
1	*Experimento_BA_DB*.altura1_V	Real		Rojo	
2	*Experimento_BA_DB*.altura2_V	Real		Cian	
3			<Agregar>		

Condiciones de registro

> Muestreo

Momento de registro: "Cyclic interrupt" %OB30

Registrar todos: 1 Ciclo 0,1 s

Duración de registro (a): 700 Puntos de medición 69,9 s

Duración de registro máx.: 32766 puntos de medición / 3276,5 s Utilizar la duración de registro máx.

> Disparo

Disparador: Registro inmediato

Valor:




Ilustración 32. Configuración Trace modelado

La representación obtenida es la siguiente, y a partir de ella se halla la función de transferencia del sistema:

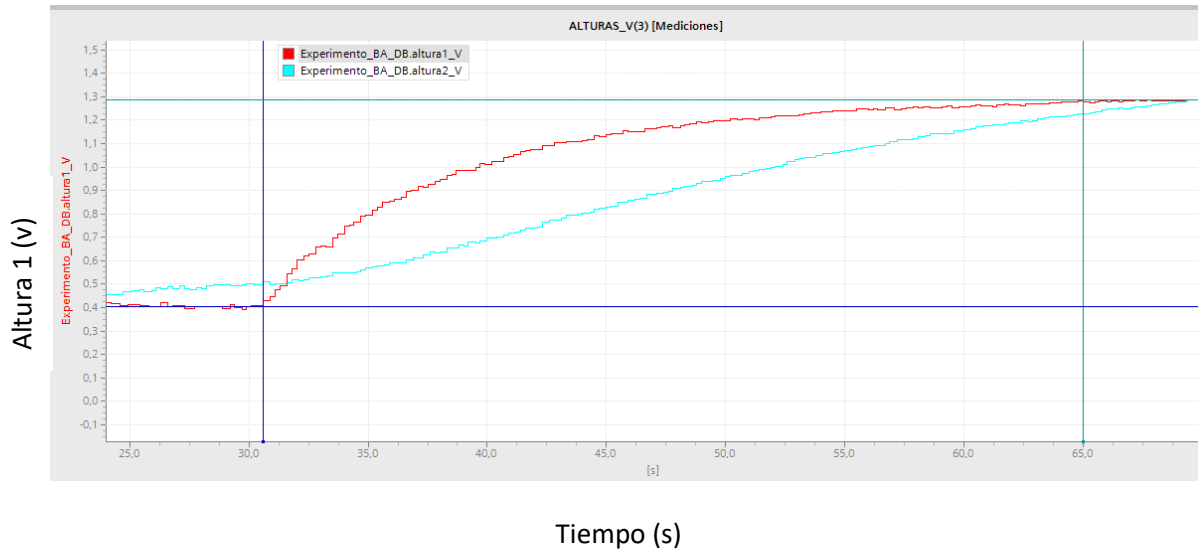


Gráfico 1. Representación altura H1

A partir de la gráfica obtenida del experimento en bucle abierto se puede obtener un modelo función de transferencia del mismo.

En este caso se ha decidido obtener la función de transferencia de la entrada respecto de la salida con ambas señales en voltios, por lo que la función representada es la lectura del sensor de altura del taque 1 en voltios.

Como se puede observar en el *Gráfico 1. Representación altura H1*, el sistema se puede aproximar a un primer orden cuya representación es la siguiente:

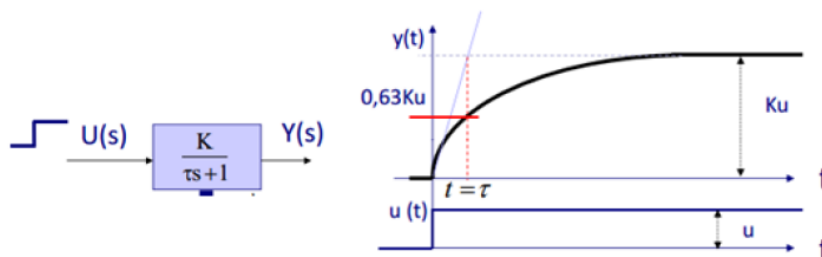


Ilustración 33. Respuesta de un sistema de primer orden a una variación en la entrada

siendo:

- K la ganancia estática

$$K = \frac{\Delta y}{\Delta u} = \frac{0.88}{0.5} = 1.76 \frac{\text{voltios de salida}}{\text{voltios de entrada}}$$

- τ la constante de tiempo, valor que se corresponde al tiempo necesario para que el sistema alcance el 63% del valor final.

$$0.63 * \Delta y + y_0 = 0.63 * 0.88 + 0.4 = 0.95616$$

$$\tau = t_{63\%} - t_0 = 38.31 - 30 = 8.31s$$

Todos los valores son sacados del *Gráfico 1. Representación altura H1*, por lo que la función de transferencia de la planta queda:

$$G(s) = \frac{K}{1+\tau s} = \frac{1.76}{1+8.31s} = \frac{H1(S)}{E(S)}$$

7.2. Diseño de controlador.

Como bien se ha dicho a lo largo del presente informe, el controlador será un PID diseñado analíticamente cuya forma es:

$$Gp(s) = \frac{U(S)}{E(S)} = Kc * (1 + \frac{1}{Ti * s} + Td * s)$$

Como se puede ver en su algoritmo, el controlador PID consta de tres acciones: la acción proporcional (K), la acción derivativa (Td) y la acción integral (Ti).

En este caso lo que se quiere es eliminar el error en el estacionario, por lo que se necesita la acción integral. Es por ello que se elige un controlador PI diseñado por el método de cancelación polo/cero, introduciendo un polo en el origen (integrador) y un cero sobre el polo de la planta para anular su efecto.

En este caso el valor de Td es cero por tratarse de un controlador PI al tener como única especificación que el error en estacionario sea cero.

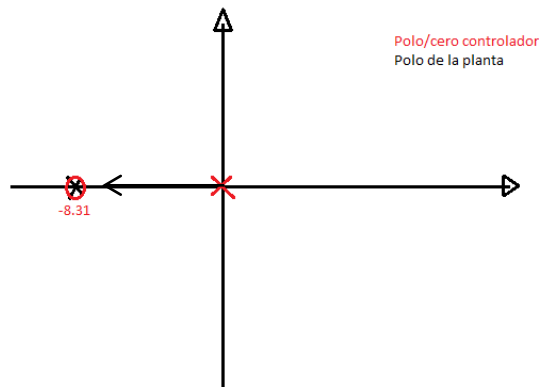


Ilustración 34. Método de cancelación polo/cero

Al elegirse el método de cancelación polo cero, el cero se encuentra sobre el polo de la planta para compensar la dinámica del proceso, por lo que $T_i = \tau = 8.31s$.

Como no se tienen más especificaciones a parte de la del error, la ganancia del controlador K_c se elige simplemente teniendo en cuenta que el sistema no se sature. Con el ajuste del valor de K_c puede hacerse el sistema más rápido. En este caso se ha elegido el valor de $K_c=1.5$.

7.3.Control directo.

La última fase del ejemplo de aplicación, es la comprobación de que el controlador diseñado cumple su objetivo: seguir a la señal de referencia sin error.

Para ello se ejecuta el FB creado para realizar el experimento en bucle cerrado. Este FB es llamado desde el OB de alarma cíclica (OB30) en el que se instancian las variables formales de entrada y salida del bloque funcional (FB).

```

1 //LLAMADA A FB.
2 "Experimento bucle cerrado_DB"(lim_sup_H_V:=5.0,
3     lim_inf_H_V:=0,
4     lim_sup_B_V:=12.0,
5     lim_inf_B_V:=0.0,
6     lim_sup_cm3s:=39.6,
7     lim_inf_cm3s:=0.0,
8     lim_sup_H_cm:=25.0,
9     lim_inf_H_cm:=0.0,
0     SP_cm:=10.0,
1     h_bin:="H1_binario",
2     Td:=0.0,
3     Ti:=8.31,
4     K:=1.5,
5     manual:="MANUAL",
6     N:=20.0,
7     T:=0.1,
8     PO_V:=1.0,
9     B_bin=>"B_binario");
0
  
```

Ilustración 35. OB30 experimento bucle cerrado.

El tiempo de muestreo elegido para la alarma cíclica del OB30 es de 0.1s , es decir el mismo que para el caso del experimento en bucle abierto.

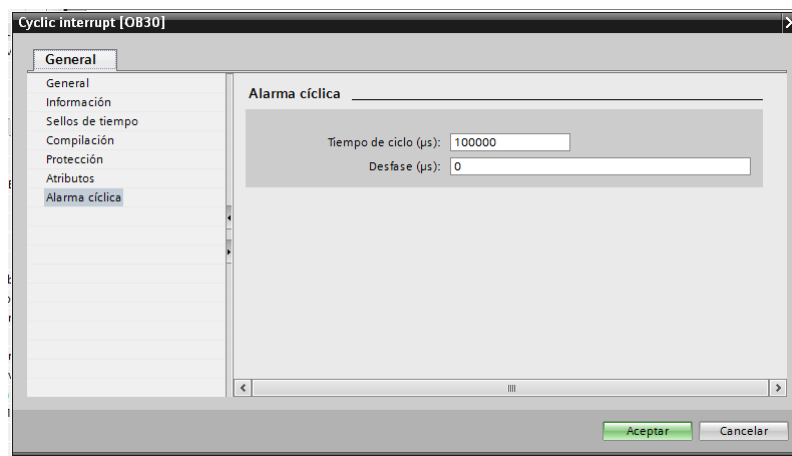
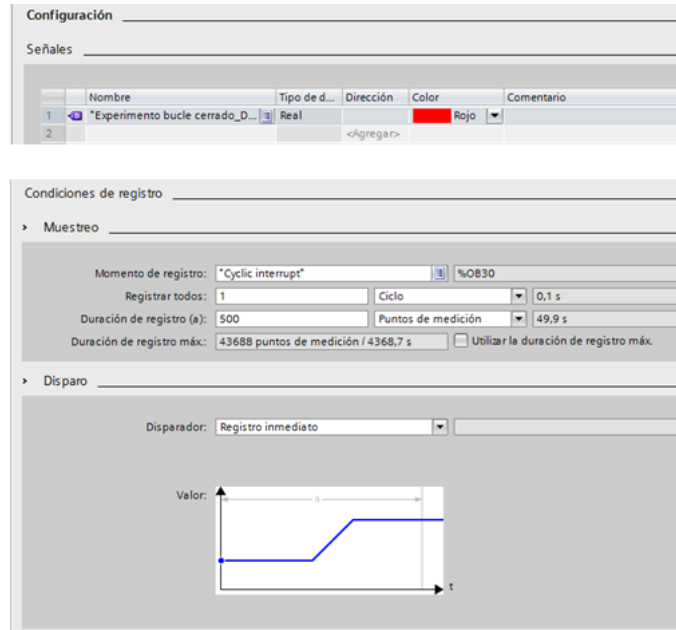


Ilustración 36. Configuración OB30 bucle cerrado.

En esta última fase también es necesario crear un TRACE para representar el seguimiento de la altura del tanque 1 (H1) a la entrada de referencia. La configuración de dicho TRACE es la siguiente:



The screenshot shows the configuration interface for a TRACE system, divided into two main sections: "Señales" (Signals) and "Condiciones de registro" (Recording Conditions).

Señales (Signals):

Nombre	Tipo de d...	Dirección	Color	Comentario
*Experimento bucle cerrado_D...	Real		Rojo	
		<Agregar>		

Condiciones de registro (Recording Conditions):

- Muestreo (Sampling):**
 - Momento de registro: "Cyclic interrupt" / %OB30
 - Registrar todos: 1 / Ciclo / 0,1 s
 - Duración de registro (a): 500 / Puntos de medición / 49,9 s
 - Duración de registro máx.: 43688 puntos de medición / 4368,7 s
 - Utilizar la duración de registro máx.
- Disparo (Trigger):**
 - Disparador: Registro inmediato

At the bottom of the "Disparo" section, there is a small graph showing a step function. The vertical axis is labeled "Valor" and the horizontal axis is labeled "t". The graph shows a constant value that then increases to a higher constant value, representing a step change in the reference input.

Ilustración 37. Configuración TRACE control directo.

La representación obtenida es la que se muestra a continuación, y en ella se puede ver como la variable controlada (altura tanque 1) sigue a la entrada de referencia, que en este caso son 10 cm, bastante de cerca.

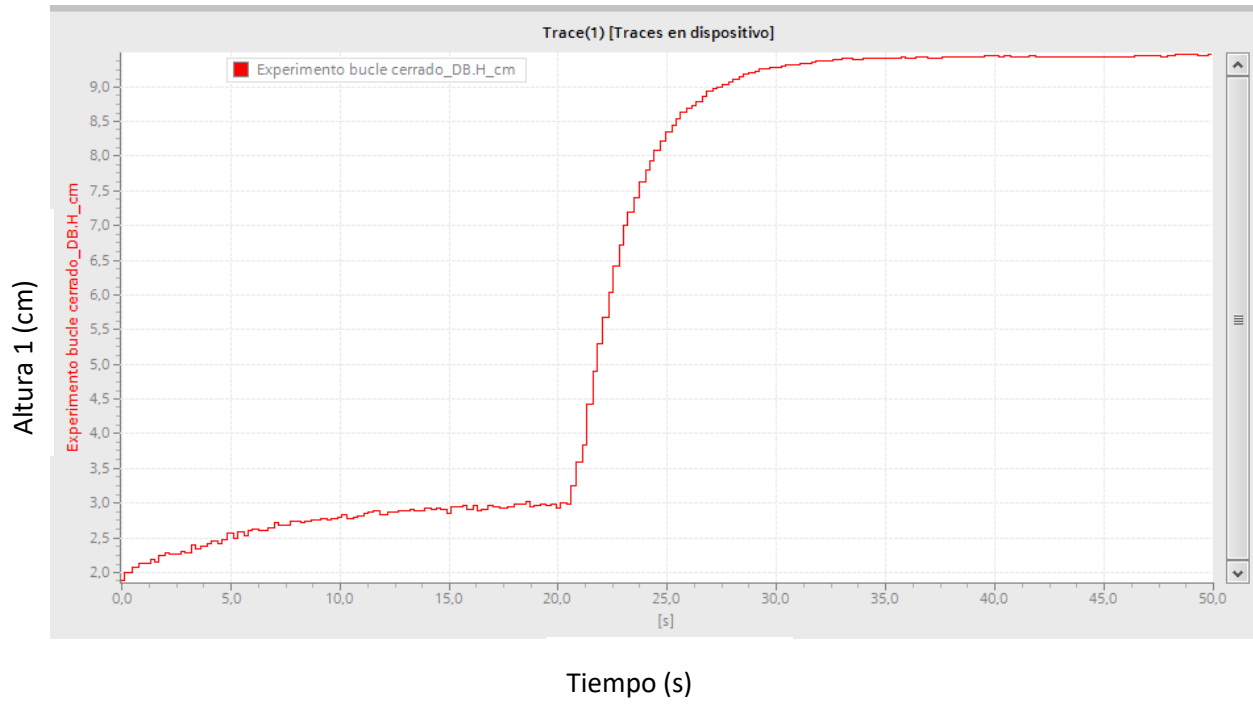


Gráfico 2. Representación TRACE de la altura a controlar

8. PRESUPUESTO

En este apartado se procede a mostrar cuál es el presupuesto necesario para la realización de este proyecto. En él encontramos 5 partidas:

- Horas internas
- Gastos
- Amortizaciones
- Costes indirectos
- Imprevistos

A continuación se muestra una tabla en la que se recogen los datos de cada partida, desglosadas tanto en unidades necesitadas como en el coste de cada una de ellas:

Concepto	Unidades	Nº unidades	Coste Unitario	Coste	TOTAL PARTIDA
HORAS INTERNAS					5.000,00 €
Ingenieria	horas	200	25,00 €	5.000,00 €	
AMORTIZACIONES					81.275,00 €
PLC	porcentaje	5%	3.500,00 €	175,00 €	
Ordenador	horas	200	400,00 €	80.000,00 €	
Maqueta tanques y amplificador de potencia(UPM)	porcentaje	5%	8.000,00 €	400,00 €	
Licencia Tia Portal	-	-	-	700,00 €	
GASTOS					20,00 €
Material de oficina	-	-	-	20,00 €	
SUBTOTAL 1					86.295,00 €
Costes indirectos (3%)					2.588,85 €
SUBTOTAL 2					88.883,85 €
Imprevistos(10%)					8.888,39 €
TOTAL					97.772,24 €

Ilustración 38. Tabla presupuesto

Como se puede observar el coste más elevado se encuentra en la partida de amortizaciones. En ella se incluyen tanto la licencia del software, como las amortizaciones de los dispositivos utilizados. El valor tomado para la amortización de la maquinaria es de un 5% ya que el uso que se les da es muy pequeño comparado con su vida útil. Por último la amortización del ordenador se ha calculado en relación a sus horas de uso.

En el segundo lugar se encuentra la partida de horas internas. Esto es debido a que mayormente, para cumplir el objetivo del proyecto, hay que llevar a cabo un largo estudio de todas las herramientas a usar. De hecho, en el siguiente punto 10. *Planificación de tareas*, se ve como la tarea más duradera es la quinta ‘estudio del correcto uso de los dispositivos’.

Por detrás de estas partidas se encuentran los costes indirectos, los imprevistos y los gastos, los cuales no son muy elevados ya que no se necesitan herramientas específicas que no puedan ser útiles al final del proyecto.

Para poder ver la relación existente entre el coste de las distintas partidas, a continuación se muestra un gráfico del porcentaje que supone cada una de ellas sobre el coste final.

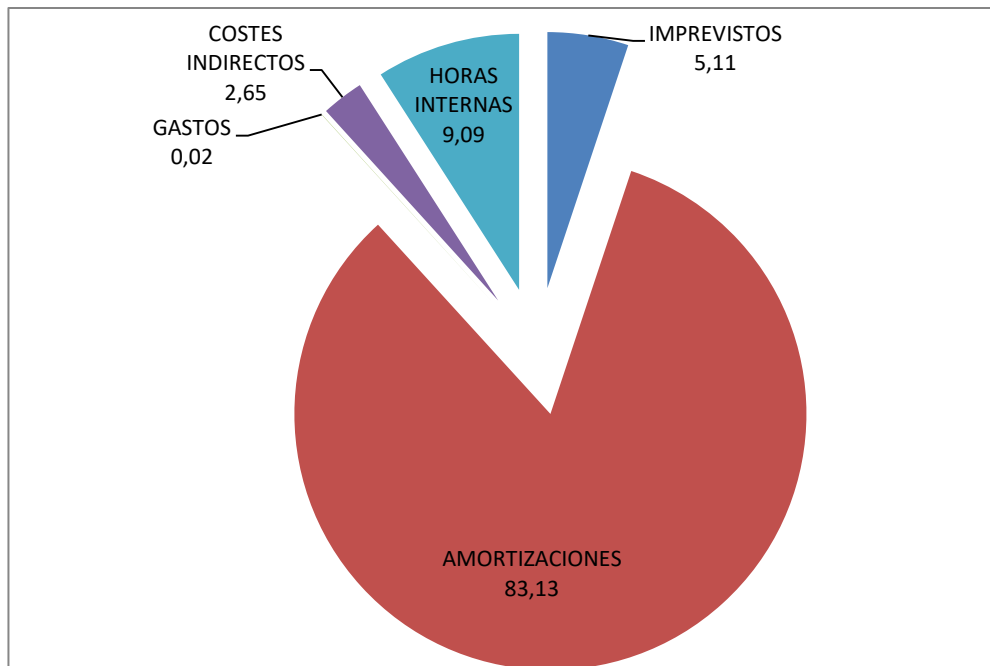


Gráfico 3. Porcentajes presupuesto

9. PLANIFICACIÓN DE TAREAS

Como ya se sabe, la planificación de tareas se lleva a cabo en función de la duración del proyecto fijada por la Universidad del País Vasco (UPV), para poder así ajustarse a los plazos establecidos. Es por ello, que el trabajo queda dividido en 9 fases detalladas posteriormente.

El inicio del trabajo se lleva a cabo sobre principios de febrero, una vez terminados los exámenes de primer cuatrimestre. Este proyecto se desarrolla en paralelo a las clases lectivas del centro, por lo que durante los primeros meses (hasta la mitad de la quinta fase aproximadamente) el horario de trabajo es de unos dos días a la semana unas 3 horas al día. Debido a la convocatoria de exámenes de segundo cuatrimestre, se lleva a cabo un parón en el mes de mayo, reanudándose el trabajo tras esta con un horario más intenso ya que no hay que compaginar los horarios lectivos por haber terminado las clases. Es por ello que en esta última etapa, reanudada en la quinta fase, el horario de trabajo será de lunes a viernes con unas 6 horas de trabajo al día. El trabajo queda finalizado a fecha de: 16 de julio de 2018.

En cuanto a las fases llevadas a cabo en el proyecto, nos encontramos con:

1. **Análisis de alternativas.**

Para comenzar el trabajo, el primer paso que hay que dar es conocer todas las posibles alternativas metodológicas, de hardware y de software, para decidir cuál de ellas se va a usar.

Esta fase tiene una duración aproximada de 1 día.

2. **Conocimiento de los distintos sistemas a usar y estudio previo de la materia.**

Constituye una de las fases más importantes de todo el proyecto ya que marca la calidad del mismo. En ella se repasan todos los conceptos que necesarios, relacionados con las asignaturas de 'Automática y control', 'Automatización industrial' y 'Control por computador'. Además, se estudia el funcionamiento de la maqueta, conociendo todas las características de sus sensores y actuadores.

Por otro lado también se aprenden a usar comandos propios del Tia Portal como el Trace, el Scale o el Unscale, que serán de vital importancia en el desarrollo del sistema de control. Toda la información es obtenida de los manuales que ofrece Siemens, y el conocimiento sobre los comandos se adquiere ensayando varias veces con distintas configuraciones comprobando así su correcto funcionamiento.

Realizar esta tarea conlleva aproximadamente 4 días.

3. Conexión de equipos, configuración hardware y calibración de la maqueta.

Para poder realizar el proyecto es necesario conectar correctamente la maqueta al amplificador de señal y al PLC., y realizar la configuración hardware del sistema. Además en esta tarea se lleva a cabo la calibración de los distintos sensores que posee la maqueta (sensores de altura de los tanques)

La duración de esta tarea es de unos 2 días.

4. Desarrollo del código del programa.

Esta tarea consiste en la creación del código del sistema de control. Para ello solamente se necesitan crear dos FB, uno para el modelado (experimento en bucle abierto) y otro para la validación del controlador (experimento en bucle cerrado), y un OB30 de interrupción cíclica.

La duración de esta tarea es de aproximadamente 4 días.

5. Estudio del correcto uso de los dispositivos.

Constituye una de los periodos más largos de trabajo. En esta etapa se prueban distintas configuraciones de los dispositivos y mediante varios ensayos se comprueba que el código creado es el correcto tanto para obtener el modelo de la planta, como para llevar a cabo el control de la misma.

Esta etapa tiene una duración de unos 15 días.

6. Realización de ensayos.

En esta fase se lleva a cabo el experimento en bucle abierto para obtener el modelo del sistema. Para ello se introduce la variación a la bomba de impulsión y se registra la variación de las alturas de los tanques a través del comando Traces previamente estudiado.

Esta tarea se realiza en 1 día.

7. Tratamiento de datos e identificación del modelo.

Con los datos obtenidos en la fase anterior se identifica el modelo de la planta y se crea analíticamente el controlador del sistema.

La duración de la tarea también es de 1 día.

8. Validación del controlador.

Esta tarea consiste en comprobar el correcto funcionamiento del sistema de control llevando a cabo el experimento en bucle cerrado. Para ello se usa el FB creado en la *fase 4.Desarrollo del código de programa.*

Su duración aproximada es de 1 día.

9. Documentación.

Redacción de la memoria de trabajo. La elaboración de esta tarea dura unos 15 días

Puede concluirse que la ejecución total del trabajo conlleva aproximadamente unas 200 horas. En la siguiente página se muestra la planificación del trabajo de forma gráfica.

10. CONCLUSIONES

Con el desarrollo del presente trabajo se ha conseguido, en primer lugar, obtener un modelo matemático aproximado que representa el comportamiento de la maqueta de tanques en serie QUANSER. Se habla de modelo aproximado porque existen ciertas desviaciones aceptables respecto del modelo exacto.

Las causas por las cuales es imposible calcular el modelo exacto de la planta engloban las siguientes:

- Las mediciones reales siempre difieren de las teóricas debido a la aparición de ruidos por tratarse de señales eléctricas.
- Calibración manual y aproximada de los sensores que puede llevar a pequeñas desviaciones de las medidas tomadas por ellos.

Aun así, la diferencia existente entre el modelo matemático y el real no es muy grande, por lo que puede ser usado para algunas aplicaciones en las que no se requiera una precisión milimétrica.

Por otro lado también se ha llegado al objetivo principal del proyecto: obtener el sistema de control del mismo.

Para ello como bien se sabe se ha implementado un PID cuya respuesta es bastante acertada. Esto se puede ver en las gráficas obtenidas del experimento en bucle cerrado creado para validar el controlador obtenido. En el *gráfico 2. Representación TRACE de la altura a controlar*, se ve como la respuesta del sistema difiere levemente de la que se quiere obtener (la altura se acerca bastante al set point de 10 cm). Esta pequeña desviación, puede deberse a la sintonización del propio PID, es decir, a la elección de sus valores (K, Td y Ti) o también a errores en la toma de medidas por las razones explicadas anteriormente.

11. FUENTES DE INFORMACIÓN

- [1] Orive,D., y López,F., Armentia, A., Iriondo,N. (2017) *Automatización Industrial. Bilbao, España.*
- [2] Iriondo,N., Marcos,M., Cabanes,I., Zubizarreta,A. (2016) *Automática y Control. Bilbao, España.*
- [3] Hans Berger *Automating with SIMATIC S7-1500. Configuring, Programming and Testing with STEP 7 Professional.*
- [4] QUANSER *Coupled Water Tanks.Student Handout.*
- [5] QUANSER *Coupled Water Tanks.Instructor Manual.*
- [6] QUANSER *Coupled Water Tanks.User Manual.*

ANEXO 1: CÓDIGOS DEL PROGRAMA EN TIA PORTAL

CODIGO DEL PROGRAMA DEL EXPERIMENTO EN BUCLE ABIERTO

A continuación se muestra el FB creado para realizar el experimento en bucle abierto correspondiente al modelado, además de los parámetros formales declarados en su cabecera.

```

1  |//caracterización sensores por estar configurados de +-10V pero ser de otros rangos(bomba y sensores de altura)
2
3  #factorcaracterización_B_V := 10 / #lim_sup_B_V;
4  #factorcaracterización_H_V:= 10 / #lim_sup_H_V;
5  #rangosensor_B_V := #factorcaracterización_B_V * #lim_sup_B_V;
6  #rangosensor_H_V := #factorcaracterización_H_V * #lim_sup_H_V;
7  #rangosensor_H_cm := #factorcaracterización_H_V * #lim_sup_cm;
8  #rangosensor_B_cm3s:=#factorcaracterización_B_V*#lim_sup_cm3s;
9
10
11 //llevar al tanque a un punto de operación y a partir de ahí meterle impulso
12 E"R_TRIG_DB"(CLK:=#Arranque,
13   Q=>#flanco_arranque);
14
15
16 IF #Arranque THEN
17
18   #tiempo_PO := S_ODTS(I_NO := #Temporizador_PO, S := #flanco_arranque, TV := #tiempoestabilizacion_PO, Q => #incremento, R := #off);
19   #tiempo_impulso := S_ODT(I_NO := #Temporizador_impulso, S := #incremento, TV := #Periodo, Q => #off);
20
21
22 IF #incremento = 0 THEN
23   #B_v := #V0_PO;
24 ELSE
25   #B_v := #Amplitud;
26
27 END_IF;
28 ELSE
29   #B_v := 0;
30 END_IF;
31
32
33 //Pasamos la tension de la bomba (V) a caudal (cm3/s)
34
35 #error_code_6 := UNSCALE(IN := #B_v, HI_LIM := #rangosensor_B_V, LO_LIM := #lim_inf_B_V, BIPOLAR := FALSE, OUT => #B_bin);
36
37 #error_code_5 := SCALE(IN := #B_bin, HI_LIM := #rangosensor_B_cm3s, LO_LIM := #lim_inf_cm3s, BIPOLAR :=FALSE, OUT => #B_cm3/s");
38
39
40
41 //Pasamos de tension(real) a word para escribir en la bomba
42
43 #error_code:=UNSCALE(IN := #B_v, HI_LIM :=#rangosensor_B_V, LO_LIM :=#lim_inf_B_V, BIPOLAR := FALSE, OUT =>"B_binario");
44
45 //Registro de las alturas de los sensores
46
47 #error_code_1:=SCALE(IN:="H1_binario", HI_LIM:=#rangosensor_H_cm, LO_LIM:=#lim_inf_cm, BIPOLAR:=FALSE, OUT=>#altural1_cm);
48
49 #error_code_2 := SCALE(IN := "H2_binario", HI_LIM :=#rangosensor_H_cm, LO_LIM :=#lim_inf_cm, BIPOLAR := FALSE, OUT => #altura2_cm);
50 //alturas de los sensores en voltios
51
52 #error_code_7:=SCALE(IN:="H1_binario", HI_LIM:=#rangosensor_H_V, LO_LIM:=#lim_inf_H_V, BIPOLAR:=FALSE, OUT=>#altural1_V);
53 #error_code_8:=SCALE(IN:="H2_binario", HI_LIM:=#rangosensor_H_V, LO_LIM:=#lim_inf_H_V, BIPOLAR:=FALSE, OUT=>#altural2_V);
54

```

En cuanto a los parámetros formales tenemos los siguientes:

Experimento_BA								
	Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible d...	Visible en ..	Valor de a..	Comentario
1	Input							
2	lim_inf_cm3s	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	lim_sup_cm3s	Real	39.6	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	lim_inf_B_V	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	lim_sup_B_V	Real	12.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	lim_inf_H_V	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7	lim_sup_H_V	Real	5.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	lim_inf_cm	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	lim_sup_cm	Real	25.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	H_control_bin	Word	16#0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
11	Amplitud	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
12	Periodo	S5Time	S5T#0ms	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
13	VO_PO	Real	5.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
14	tiempoestabilizacion...	S5Time	S5T#0ms	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
15	Arranque	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
16	Temporizador_PO	Timer	0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
17	Temporizador_impulso	Timer	0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
18	Output							
19	altura1_cm	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
20	altura2_cm	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
23	Static							
24	off	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
25	flanco_arranque	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
26	PO_bin	Word	16#0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
27	B_bin	Word	16#0	No remane...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
28	B_cm3/s	Real	0.0	No remane...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
29	B_v	Real	0.0	No remane...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
30	factorcaracterizacion...	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
31	factorcaracterización...	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
32	rangosensor_B_V	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
33	rangosensor_H_V	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
34	rangosensor_H_cm	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
35	rangosensor_B_cm3s	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
36	altura1_V	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
37	incremento	Bool	false	No remane...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
38	altura2_V	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
39	Temp							
40	tiempo_PO	S5Time			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
41	tiempo_impulso	S5Time			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
42	error_code	Word			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
43	error_code_1	Word			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
44	error_code_2	Word			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
45	error_code_3	Word			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
46	error_code_4	Word			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
47	error_code_5	Word			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
48	error_code_6	Word			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
49	error_code_7	Word			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
50	error_code_8	Word			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

CODIGO DEL PROGRAMA DEL EXPERIMENTO EN BUCLE CERRADO

A continuación se muestra el FB creado para realizar el experimento en bucle cerrado correspondiente al control directo, además de los parámetros formales declarados en su cabecera.

```

1 //caracterización sensores por estar cpnfigurados entre +-10V y ellos ser de distinto rango
2 #caracterizacionsensor_B_V := 10 / #lim_sup_B_V;
3 #caracterizacionsensor_H_V := 10 / #lim_sup_H_V;
4 #rangosensor_B_V := #caracterizacionsensor_B_V * #lim_sup_B_V;
5 #rangosensor_H_V := #caracterizacionsensor_H_V * #lim_sup_H_V;
6 #rangosensor_H_cm := #caracterizacionsensor_H_V * #lim_sup_H_cm;
7 #"rangosensor_B_cm3/s" := #caracterizacionsensor_B_V * #lim_sup_cm3s;
8
9 //Calculo del error a partir del set point y la lectura de los sensores
10 //Pasas la lectura de la altura de binario a real(V)
11 #Error_code:=SCALE(IN:=#h_bin, HI_LIM:=#rangosensor_H_V, LO_LIM:=#lim_inf_H_V, BIPOLAR:=FALSE, OUT=>#Y_V);
12
13 //Pasas el set point en altura(cm) a set point tension(V)
14 #Error_code_1 := UNSCALE(IN := #SP_cm, HI_LIM := #rangosensor_H_cm, LO_LIM := #lim_inf_H_cm, BIPOLAR := FALSE, OUT => #SP_bin);
15 #Error_code_2 := SCALE(IN := #SP_bin, HI_LIM := #rangosensor_H_V, LO_LIM := #lim_inf_H_V, BIPOLAR := FALSE, OUT => #SP_V);
16
17
18
19
20
21 IF #manual=0 THEN
22   #u_V := #PQ_V;
23
24 ELSE
25   //cálculo del error
26
27   #e := #SP_V-#Y_V;
28
29   //algoritmo PID
30   //inicialización de variables
31   #upl := 0;
32   #udl := 0;
33   #uil := 0;
34
35   //PID discretizado
36   #up := #K * #e;
37   #ui := #uil + #K * #T * (#e + #e1) / (2 * #Ti);
38   #ud := (#udl * #Td + #K * #Td * #N * (#e - #e1)) / (#N * #T + #Td);
39
40   #u_V := #up + #ui + #ud;
41
42   //#u1 := #upl + #udl + #uil;
43
44   //actualización variables e1,u1
45   #upl := #up;
46   #uil := #ui;
47   #udl := #ud;
48   #e1 := #e;
49 END_IF;
50
51
52
53 //pasar las señal u(v) a u(cm3/s) y a binario (B_bin)
54 //Pasas la señal u(V) a señal binaria B_bin
55
56 #Error_code_3 := UNSCALE(IN := #u_V, HI_LIM := #rangosensor_B_V, LO_LIM := #lim_inf_B_V, BIPOLAR := FALSE, OUT => #B_bin);
57 #Error_code_4 := SCALE(IN := #B_bin, HI_LIM := #"rangosensor_B_cm3/s", LO_LIM := #lim_inf_cm3s, BIPOLAR := FALSE, OUT => #"u_cm3/s");
58 //pasar la señal de lectura de altura a cm
59
60 #Error_code_5 := SCALE(IN := #h_bin, HI_LIM := #rangosensor_H_cm, LO_LIM := #lim_inf_H_cm, BIPOLAR := FALSE, OUT => #H_cm);
61

```


En cuanto a los parámetros formales tenemos los siguientes:

Experimento bucle cerrado_DB								
	Nombre	Tipo de datos	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a..	Comentario
1	Input			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	lim_sup_H_V	Real	5.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	lim_inf_H_V	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	lim_sup_B_V	Real	12.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	lim_inf_B_V	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	lim_sup_cm3s	Real	39.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7	lim_inf_cm3s	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	lim_sup_H_cm	Real	25.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	lim_inf_H_cm	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	SP_cm	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	set point
11	h_bin	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	lectura altura sensor_binario
12	Td	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
13	Ti	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
14	K	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
15	manual	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
16	N	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
17	T	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
18	PO_V	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
19	Output			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
20	B_bin	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	señal a la bomba (binaria)
22	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
23	e	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
24	e1	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
25	up	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
26	ud	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
27	ui	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
28	u_V	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
29	up1	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
30	ud1	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
31	ui1	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
32	u1	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
33	SP_bin	Word	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
34	SP_V	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
35	u_cm3/s	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
36	Y_V	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	lectura altura en voltios
37	caracterizacionsensor...	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
38	caracterizacionsensor...	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
39	rangosensor_H_V	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
40	rangosensor_B_V	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
41	rangosensor_H_cm	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
42	rangosensor_B_cm3/s	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
43	H_cm	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
44	B_V	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	