

Trabajo Fin de Máster

Máster en Ingeniería Industrial

Control de un sistema de cuatro tanques usando CODESYS y Citect SCADA

Autor: Antonio Jesús Palomo Cantador

Tutor: Francisco Salas Gómez

Dpto. Ingeniería de sistemas y automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021



Trabajo Fin de Máster
Máster en Ingeniería Industrial

Control de un sistema de cuatro tanques usando CODESYS y Citect SCADA

Autor:

Antonio Jesús Palomo Cantador

Tutor:

Francisco Salas Gómez

Profesor titular

Dpto. de Ingeniería de sistemas y automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2021

Trabajo Fin de Máster: Control de un sistema de cuatro tanques usando CODESYS y Citect SCADA

Autor: Antonio Jesús Palomo Cantador

Tutor: Francisco Salas Gómez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal

A mi familia

A mis seres queridos

Agradecimientos

Me gustaría agradecer estos años a mi familia por ser el soporte y el apoyo fundamental desde que empecé mi etapa universitaria hace ya unos años, dándome siempre la estabilidad que necesito para seguir adelante.

También me gustaría destacar la labor de mi tutor del trabajo, que de nuevo ha sido una ayuda fundamental en el desarrollo del trabajo, en este caso a pesar de la distancia.

Finalmente, no puedo olvidarme de las personas que han aparecido en este último de año de Erasmus, viviendo e interesándose conmigo día a día los problemas de este trabajo. En especial, mis amigos del Campus San Paolo han sido ese empujón final que necesitaba.

Resumen

El servidor OPC-UA y el control remoto están ganando terreno a otros antiguos métodos de comunicación año tras año. Son dos componentes fundamentales en la ya existente revolución industrial inteligente 4.0.

El trabajo trata de recrear un modelo basado en un sistema de cuatro tanques mediante la herramienta Matlab&Simulink. A esto se incorpora un control PID mediante el software Codesys 3.5. Finalmente se añade un SCADA sobre el control del sistema en el software Citect SCADA 2018.

La comunicación de los tres programas se realiza mediante el servidor OPC-UA, introduciendo Codesys como servidor mientras que Matlab&Simulink y Citect SCADA quedarían como clientes.

Finalmente se incluye un anexo con la guía detallada de la conexión al servidor OPC-UA de los tres programas utilizados.

Abstract

The OPC-UA server and remote control are gaining ground on other old communication methods year after year. They are two fundamental components in the already existing intelligent industrial revolution 4.0.

The project tries to recreate a model based on a four-tank system using the Matlab&Simulink tool. Added to this is a PID control using Codesys 3.5 software. Finally, a SCADA is included in the control of the system in the software Citect SCADA 2018.

The communication of the three programs is carried out through the OPC-UA server, introducing Codesys as server while Matlab&Simulink and Citect SCADA would remain as clients.

Finally, an annex is included with the detailed guide for connecting to the OPC-UA server for the three programs used.

ÍNDICE DE TABLAS	XVII
ÍNDICE DE FIGURAS	XIX
1 INTRODUCCIÓN	1
1.1. HISTORIA	1
1.2. MOTIVACIÓN Y OBJETIVOS POR CUMPLIR	3
1.3. ORGANIZACIÓN DEL DOCUMENTO	3
2 ESTRUCTURA DE CONTROL	5
2.1. SOFTWARE	5
2.2. COMUNICACIÓN	5
3 MODELADO DEL SISTEMA	9
3.1. SISTEMA IDEAL	9
3.1.1. <i>Aplicación real</i>	9
3.2. MODELO CREADO	10
3.2.1. <i>Características físicas importadas</i>	10
3.2.2. <i>Bloques de Simulink</i>	11
3.2.3. <i>Modelo final</i>	12
4 CONTROL DEL SISTEMA	15
4.1. CONTROL PID	15
4.2. CONFIGURACIONES DEL SISTEMA	17
4.2.1. <i>Primera configuración</i>	17
4.2.1.1. Control de la primera configuración	17
4.2.2. <i>Segunda configuración</i>	19
4.2.1.1. Control Ziegler-Nichols	19
4.2.1.2. Control mediante especificaciones	21
4.2.3. <i>Tercera configuración</i>	23
4.2.3.1. Control de la tercera configuración	24
4.2.4. <i>Coordinación de las tres configuraciones</i>	26
5. SCADA	29
5.1. COMPONENTES DE LAS PANTALLAS DE EXPLOTACIÓN	29
5.1.1. <i>Objetos básicos</i>	29
5.1.2. <i>Funciones Cicode</i>	30
5.1.3. <i>Representación de la evolución de las variables</i>	33
5.2. PANTALLAS FINALES DE EXPLOTACIÓN	34
6. RESULTADOS Y CONCLUSIONES	37
6.1. RESULTADOS DE LAS SIMULACIONES	37
6.1.1. <i>Resultados de la primera configuración</i>	38
6.1.2. <i>Resultados de la segunda configuración</i>	39
6.1.3. <i>Resultados de la tercera configuración</i>	41
6.2. CONCLUSIONES Y PROPUESTAS FINALES DEL TRABAJO	43
BIBLIOGRAFÍA	45

GLOSARIO	47
ANEXO-A	49
A.1. CONFIGURACIÓN DEL SERVIDOR OPC-UA EN CODESYS	49
A.1.1. <i>Ejecución del proyecto con el servidor de comunicación</i>	51
A-2. CONFIGURACIÓN DE UN CLIENTE EN EL SOFTWARE CITECT SCADA 2018	53
A.3. CONEXIÓN DEL CLIENTE EN MATLAB&SIMULINK	60
A.3.1 <i>Incorporación de la comunicación en el modelo</i>	62

ÍNDICE DE TABLAS

<i>Tabla 3-1. Características físicas del sistema</i>	<i>11</i>
<i>Tabla 3-2. Bloques del modelo Simulink</i>	<i>12</i>
<i>Tabla 4-1. Parámetros del bloque PID_FIXCYCLE</i>	<i>16</i>
<i>Tabla 5-1. Objetos básicos en Citect SCADA</i>	<i>30</i>

ÍNDICE DE FIGURAS

<i>Figura 1-1. Reloj de agua árabe</i>	1
<i>Figura 1-2. Submarino con realimentación</i>	2
<i>Figura 2-1. Iconos del software utilizado</i>	5
<i>Figura 2-2. Esquema servidor-cliente del Proyecto</i>	6
<i>Figura 2-3. Variables comunicadas a través del servidor OPC</i>	6
<i>Figura 3-1. Esquema original del sistema de cuatro tanques</i>	9
<i>Figura 3-2. Sistema real del laboratorio</i>	10
<i>Figura 3-3. Modelo final</i>	13
<i>Figura 3-4. Comunicación del modelo final</i>	13
<i>Figura 4-1. Algoritmo PID implementado por Codesys</i>	15
<i>Figura 4-2. Bloque PID_FIXCYCLE</i>	16
<i>Figura 4-3. Esquema configuración 1</i>	17
<i>Figura 4-4. Controlador PID de la configuración 1</i>	18
<i>Figura 4-5. Esquema configuración 2</i>	19
<i>Figura 4-6. Parámetros Ziegler-Nichols en bucle abierto</i>	20
<i>Figura 4-7. Tabla Ziegler-Nichols en bucle abierto</i>	20
<i>Figura 4-8. Controlador PID de la configuración 2</i>	23
<i>Figura 4-9. Esquema configuración 3</i>	23
<i>Figura 4-10. Emparejamientos de la tercera configuración</i>	24
<i>Figura 4-11. Controlador PID de la configuración 3</i>	26
<i>Figura 4-12. Tarea principal en Codesys</i>	27
<i>Figura 5-1. Pantalla para acceder a las funciones Cicode</i>	31
<i>Figura 5-2. Función Cicode menú desplegable</i>	31
<i>Figura 5-3. Objeto gráfico de la configuración</i>	32
<i>Figura 5-4. Función Cicode configuración de válvulas</i>	32
<i>Figura 5-5. Guardar y cargar la representación de variables</i>	33
<i>Figura 5-6. Representación de la evolución de variables</i>	34
<i>Figura 5-7. Pantalla principal de explotación</i>	35
<i>Figura 6-1. Iniciar PLC virtual</i>	37
<i>Figura 6-2. Iniciar el modelo de Matlab&Simulink</i>	37
<i>Figura 6-3. PageList en Citect SCADA</i>	38
<i>Figura 6-4. Respuesta ante escalón en la configuración 1</i>	38
<i>Figura 6-5. Respuesta del sistema en bucle cerrado en la configuración 1</i>	39
<i>Figura 6-6. Respuesta ante escalón en la configuración 2</i>	40
<i>Figura 6-7. Respuesta del sistema en bucle cerrado en la configuración 2</i>	41
<i>Figura 6-8. Respuesta antes escalón en la configuración 3</i>	42
<i>Figura A-1. Lista de variables globales</i>	49
<i>Figura A-2. Lista de configuración de símbolos</i>	50
<i>Figura A-3. Modo simulación del proyecto</i>	50
<i>Figura A-4. Iconos ocultos en la pantalla</i>	51
<i>Figura A-5. Inicio de la ejecución del proyecto</i>	52
<i>Figura A-6. Configuración de la comunicación</i>	52
<i>Figura A-7. Conexión manual del dispositivo a la puerta de enlace</i>	53
<i>Figura A-8. Creación de un nuevo proyecto Citect SCADA</i>	54
<i>Figura A-9. Asistente de configuración de nuevo dispositivo</i>	54

<i>Figura A-10. Pantalla de crear un nuevo dispositivo</i>	55
<i>Figura A-11: Método de comunicación del dispositivo</i>	55
<i>Figura A-12: Dirección del servidor OPC</i>	56
<i>Figura A-13. Vinculación del dispositivo E/S</i>	56
<i>Figura A-14: Selección de las variables comunicadas</i>	57
<i>Figura A-15. Variables cargadas en el servidor OPC-UA</i>	57
<i>Figura A-16. Búsqueda del configurador Citect SCADA</i>	58
<i>Figura A-17. Configurar un nuevo cliente OPC-UA</i>	58
<i>Figura A-18. Asistente de configuración</i>	59
<i>Figura A-19. Ejecución del modo demostración</i>	60
<i>Figura A-20. Función de comunicación OPC-UA en Matlab</i>	61
<i>Figura A-21. Incorporación de la comunicación en el modelo Matlab&Simulink</i>	63
<i>Figura A-22. Nomenclatura del bloque "Interpreted Matlab Fcn"</i>	63

Notación

Setpoint

Solver

Punto de referencia

Solucionador de ecuaciones en Matlab

1 INTRODUCCIÓN

El control automático ha desempeñado una función vital en el avance de la ingeniería y la ciencia.

- Katsuhiko Ogata -

Este capítulo introductorio tratará sobre la historia de los elementos del trabajo y los objetivos marcados para completar el proyecto.

1.1. Historia

Los inicios de la teoría del control comienzan con la preocupación de árabes y griegos por determinar con precisión el devenir del tiempo. Los primeros relojes de agua con un regulador flotador se sitúan en torno al 300 a.C.

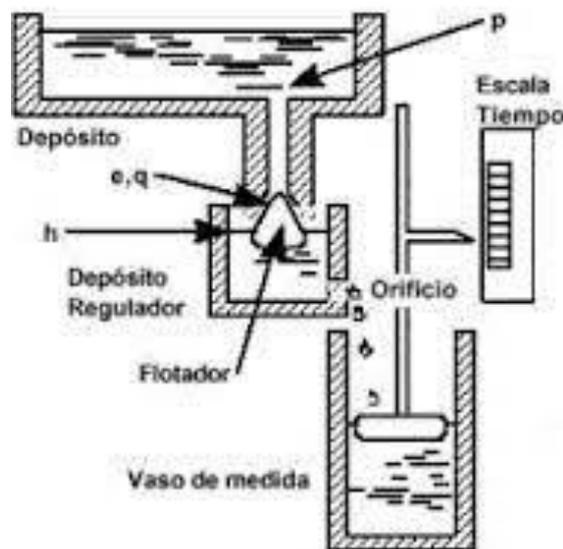


Figura 1-1. Reloj de agua árabe

No fue hasta el siglo XVII cuando Cornelius Drebbel introdujo el primer sistema con realimentación explícita para medir la temperatura. Llegó a construir un submarino en 1620 con varios sistemas de realimentación instalados.

Años después, el primer controlador con realimentación automática en un proceso industrial fue el regulador centrífugo de James Watt (1769) para controlar la velocidad de la máquina de vapor.



Figura 1-2. Submarino con realimentación

Hasta finales del siglo XIX el control automático se mantuvo como fundamentalmente intuitivo. Fue J.C. Maxwell quien creó la primera teoría de control usando el modelo de una ecuación diferencial en 1868. Por otra parte, durante la segunda guerra mundial se intensificó el desarrollo de sistemas de control.

Desde la década de 1980 y hasta la actualidad, la utilización de computadoras digitales como componentes de control se ha convertido en una rutina.

Por su parte, uno de los ejemplos más antiguos del control PID fue ideado por Elmer Sperry, mientras que el primer desarrollo teórico de los controladores tipo PID nacieron para el gobierno automático de buques y puede decirse que fueron descubiertos por el científico Nicolás Minorsky entorno al 1921. La dinámica de un buque obedecía la ecuación de movimiento del estilo de:

$$Ae'' + Be' + \tau\rho = D$$

Donde “e” es el error en dirección y “rho” es el ángulo del timón.

Finalmente queda añadir que el estándar OPC (inicialmente OLE, pero renombrado Open Protocol Communications) surgió a mediados de los años 90 como una tecnología para la comunicación del software de tipo SCADA y otros HMI con los dispositivos de control en el ámbito de la industria. Tras la irrupción en pocos años del OPC-DA, también llamado OPC clásico para algunas utilidades concretas, con el paso de los años los protocolos industriales han tenido que cumplir más requisitos para poderse adaptar a los nuevos entornos. Para cubrir estas nuevas necesidades, la OPC Foundation diseñó un nuevo estándar llamado OPC Unified Architecture (UA).

1.2. Motivación y objetivos por cumplir

Tras las últimas actualizaciones, OPC UA incorpora grandes cambios y mejoras funcionales respecto a las otras versiones, desde temas de seguridad (hoy en día tan necesaria), a nuevos modelos de datos para agrupar etiquetas y optimizar la comunicación, o facilidad en su configuración, entre otras.

Por su parte, la idea de un control remoto y la comunicación entre varias aplicaciones de manera simultánea se antoja indispensable debido a la evolución de la pandemia durante este último año.

La existencia de un sistema de cuatro tanques interconectados en el laboratorio del departamento de Ingeniería de sistemas y automática de la Escuela Superior de Ingeniería da origen al modelo del sistema con el que se trabaja en este proyecto. A partir de aquí, el objetivo fundamental del proyecto será crear un control del nivel de agua sobre un modelo en Matlab&Simulink basado en el sistema real del laboratorio.

Se procede a desglosar los objetivos del proyecto:

- Comunicación mediante el servidor OPC-UA entre los programas Codesys 3.5, Matlab&Simulink y Citect SCADA 2018. Se añadirá un anexo como guía sobre cómo establecer esta comunicación entre los tres programas.
- Modelado del sistema de cuatro tanques en un diagrama de bloques en Matlab&Simulink. Se intentarán adecuar las características físicas del sistema con la mayor exactitud posible.
- Crear un control PID teórico sobre tres configuraciones del sistema por separado. Para ello se realizará un análisis teórico previo, ajustando los parámetros de manera precisa posteriormente.
- Creación de un SCADA que sirva como interfaz para el control del sistema.
- Adquisición de resultados de las tres configuraciones y conclusiones.

1.3. Organización del documento

Una vez que se han establecido los objetivos, es importante clarificar el contenido de los siguientes capítulos del documento:

- En el capítulo 2, se explican los distintos programas utilizados y el servidor de comunicación empleado entre ellos.
- Por su parte, el capítulo 3 trata sobre el modelo para simulación creado, a partir de un sistema real. Incluye detalladamente cada uno de los componentes del modelo.
- A continuación, el capítulo 4 contiene la definición del control utilizado de manera teórica y práctica en las tres configuraciones utilizadas sobre el modelo de simulación.
- El capítulo 5 muestra el SCADA creado que serviría como interfaz HMI entre un hipotético operario y el controlador del sistema.
- Finalmente, el capítulo 6 ilustra los resultados obtenidos durante el proyecto y las conclusiones derivadas de estos resultados.

Cabe añadir que se ha incluido un anexo sobre la configuración de la comunicación en cada uno de los tres programas utilizados durante el proyecto.

2 ESTRUCTURA DE CONTROL

Se utilizará este capítulo para desgarnar los programas utilizados y la estructura de la comunicación empleada durante el proyecto.

2.1. Software

El desarrollo del proyecto se basará en tres grandes pilares: Modelo para simulación, Control y SCADA. Para cada uno de estos pilares se utilizará el software conveniente.

En primer lugar, para el modelado se recurre a Matlab&Simulink con el toolbox “Simscape” incluido. En concreto se ha utilizado una de las últimas versiones (R2020a) para que no existan problemas con la comunicación ya que se empleará un servidor relativamente actual.

Para el caso del control, se maneja el software Codesys V3.5 SP16 Patch 3. A partir de la versión 3.5 SP16 se permite la comunicación OPC-UA como servidor o cliente sin la necesidad de instalar ningún paquete adicional extra como si ocurre en el caso de la comunicación OPC-DA.

Finalmente se desarrolla un SCADA en el software Citect SCADA 2018, una de las versiones más actuales del grupo Citect que incorpora de nuevo la ya nombrada comunicación OPC-UA.



Figura 2-1. Iconos del software utilizado

2.2. Comunicación

Seguidamente, se desgarna el papel de cada programa en el servidor de comunicación ya nombrado. El servidor OPC-UA es un protocolo de tipo cliente-servidor creado para reemplazar al servidor OPC-DA en el caso de que se requieran conexiones a gran escala. Incorpora cuatro grandes ventajas con respecto a su antecesor:

- Fácil integración → lo que reduce notablemente los costes
- Fiabilidad de los datos → incorpora información imposible de descifrar

- Interoperabilidad → incorpora soluciones basadas en MacOS, Linux y PLCs con sistemas operativos propios.
- Mayor alcance de las arquitecturas en tiempo real sin la necesidad de usar archivos históricos.

Tras todas estas ventajas, se procede a clarificar que sucede realmente en el proyecto. El servidor será el autómata virtual integrado en Codesys mientras que el modelo de Matlab&Simulink y el SCADA de Citect pasarán a ser clientes. Como esquema clarificador se incorpora la figura 2-2.

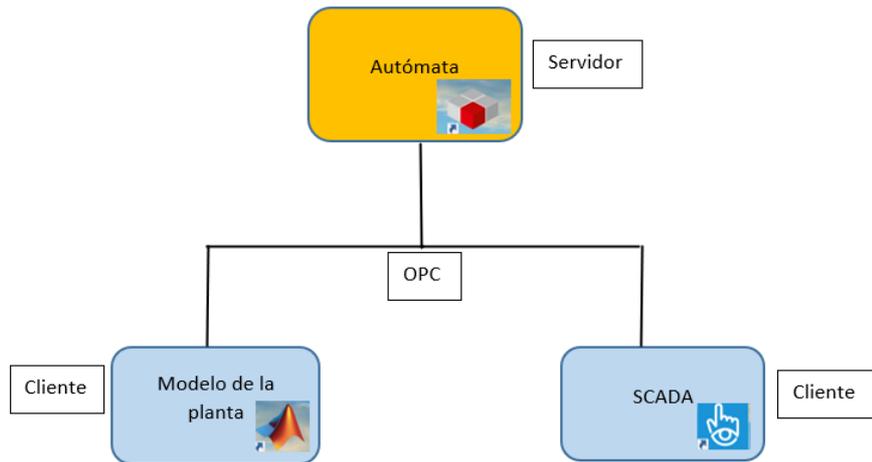


Figura 2-2. Esquema servidor-cliente del Proyecto

Además, se debe añadir que los tres programas se han coordinado de manera que la simulación y la comunicación se realiza en tiempo real gracias a la ayuda del bloque OPC Config-real-time. Los detalles de este bloque se explicarán más adelante.

Finalmente se adjunta la figura 2-3, donde se observan todas las variables que se comunican a través del servidor de Codesys a lo largo de todo el proyecto.

Variable	Valor	Valor	Tipo
Alarma_Limite			BOOL
Alarma_Overflow			BOOL
Automatico_manual			BOOL
Referencia1			REAL
Referencia2			REAL
bomba1_manual			REAL
bomba1rpm			REAL
bomba2_manual			REAL
bomba2rpm			REAL
configuracion			INT
tanque1_nivel			REAL
tanque2_nivel			REAL
tanque3_nivel			REAL
tanque4_nivel			REAL
valvula1			BOOL
valvula2			BOOL
valvula3			BOOL
valvula4			BOOL
valvula5			BOOL
valvula6			BOOL

Figura 2-3. Variables comunicadas a través del servidor OPC

Los clientes tendrán acceso de lectura y escritura a todas las variables que aparecen en la figura anterior en cualquier momento. Esto no quiere decir que ambos clientes utilicen todas ellas como se verá en los próximos capítulos.

3 MODELADO DEL SISTEMA

Este capítulo trata sobre cómo se han adaptado las características reales del sistema a un modelo de bloques empleando Matlab&Simulink.

3.1 Sistema ideal

El origen de este sistema se remonta a Karl Henrik Johansson, que introdujo un sistema de cuatro tanques con fines didácticos, concretamente nociones sobre teoría de control en condiciones de laboratorio como el efecto de los ceros en la estabilidad de un sistema. En la figura 3-1. se encuentra un esquema sobre el diagrama original.

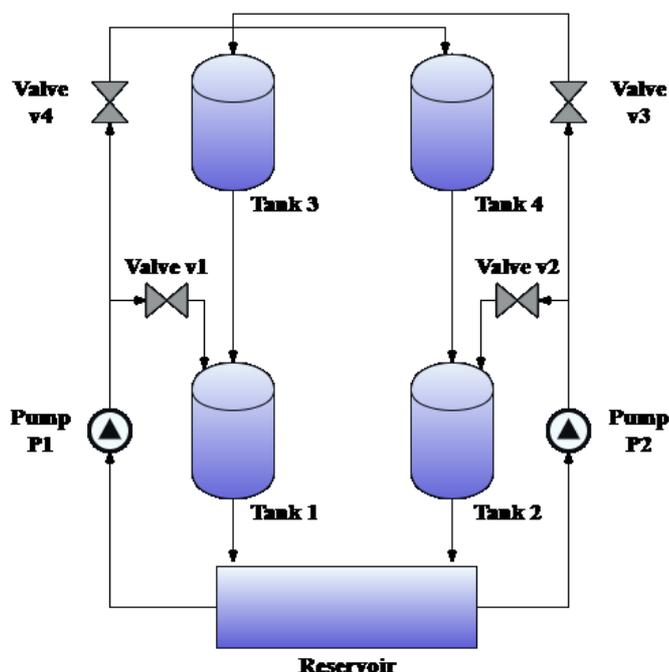


Figura 3-1. Esquema original del sistema de cuatro tanques

3.1.1. Aplicación real

A partir del sistema original, se ha creado un sistema real en el laboratorio del departamento de Ingeniería de sistemas y automática nombrado anteriormente. A diferencia del modelo original, incluye dos válvulas simétricas que conectan la bomba 1 con el tanque 3 y la bomba 2 con el tanque 4 respectivamente. Esto hace un total de 6 válvulas en el sistema.

A continuación, se adjunta una imagen del modelo real del laboratorio.



Figura 3-2. Sistema real del laboratorio

3.2 Modelo creado

A continuación, se procede a explicar el modelo creado gracias a la herramienta Matlab&Simulink. Cabe destacar que en el modelado se han intentado representar con la mayor exactitud posible las características físicas del sistema.

3.2.1. Características físicas importadas

En la siguiente tabla se adjuntan los datos más relevantes recogidos gracias al material proporcionado por el tutor del proyecto:

CARACTERÍSTICAS	DIMENSIONES
Diámetros de descarga de los tanques intermedios	$d_1=d_2=1.825\text{cm}$
Diámetros de descarga de los tanques superiores	$d_3=d_4=1.29\text{cm}$
Diámetro externo de las tuberías de polietileno	$d_{\text{ext}}=20\text{mm}$
Diámetro interno de las tuberías de polietileno	$d_{\text{int}}=15\text{mm}$
Base del depósito acumulador	$A=60 \times 150=0.9 \text{ m}^2$
Altura del depósito acumulador	$h=50\text{cm}$

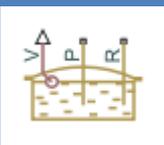
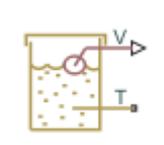
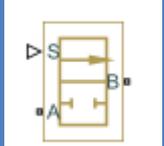
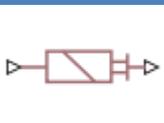
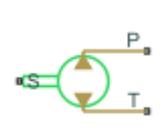
Base de los otros cuatro tanques	$A=20 \times 30=0.06m^2$
Altura de los tanques intermedios	$h_1=h_2=150cm$
Altura de los tanques superiores	$h_3=h_4=140cm$
Altura relativa de las bombas con respecto al tanque acumulador	$h_{bomba} < h_{acumulador}$

Tabla 3-1. Características físicas del sistema

3.2.2. Bloques de Simulink

En este apartado se explica cómo se han adaptado las condiciones del sistema al modelo de bloques, teniendo en cuenta tanto las características físicas como las peculiaridades de acoplar todos los bloques, la mayoría de ellos pertenecientes a la librería Simscape.

Se adjunta la tabla 3-2 explicando detalladamente los bloques utilizados y su función.

SÍMBOLOS	NOMBRE DEL BLOQUE	OBSERVACIONES Y PARÁMETROS
	RESERVOIR	Tanque acumulador con dimensiones: 150x60x50cm Volumen inicial (tanque lleno): 450 litros
	TANK	2 tanques superiores: 20x30x140 cm 2 tanques intermedios: 20x30x150 cm Volumen inicial: Setpoint establecido
	2-WAY DIRECTIONAL VALVE	6 válvulas para los cuatro tanques Se abren cuando se activa el actuador Área de paso=Área interna de la tubería= $1.77e^{-4}m^2$
	2-POSITION VALVE ACTUATOR	Inicialmente inactivo Se activa cuando su entrada “cruza” el 50% del valor nominal establecido
	CHECK VALVE	10 válvulas unidireccionales (una junto a cada entrada o salida de un tanque) Permite el paso solo en un sentido y solventa el problema de distintas presiones en el sistema
	FIXED DISPLACEMENT PUMP	Desplazamiento de $30 cm^3/rev$ que permite conseguir la propulsión adecuada en las bombas teniendo en cuenta el rango de velocidades en el que se desea trabajar

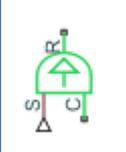
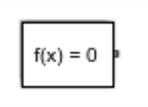
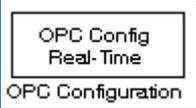
	IDEAL ANGULAR VELOCITY SOURCE	Recibe una entrada de velocidad en rpm y la transmite a la bomba de desplazamiento
	RESISTIVE PIPE LP	Representa a las tuberías en el sistema que unen depósitos, bombas y válvulas Diámetro interno: 15mm
	ELBOW	Se incluye en cada giro de 90° de una tubería
	SIMULINK-PS CONVERTER	Convierte las señales físicas en booleanas para poder medirlas y representarlas
	HYDRAULIC FLUID	Establece las propiedades del agua con la que se trabaja
	SOLVER CONFIGURATION	Define la configuración del simulador, que trabaja con variables físicas
	MEMORY	Usado en la comunicación Ayuda a resolver los bucles algebraicos
	OPC CONFIG REAL-TIME	Simula el modelo en tiempo real incluyendo la comunicación OPC con el servidor

Tabla 3-2. Bloques del modelo Simulink

3.2.3. Modelo final

A continuación, se adjunta como quedaría el modelo final incluyendo todos los bloques y parámetros mencionados en los apartados anteriores:

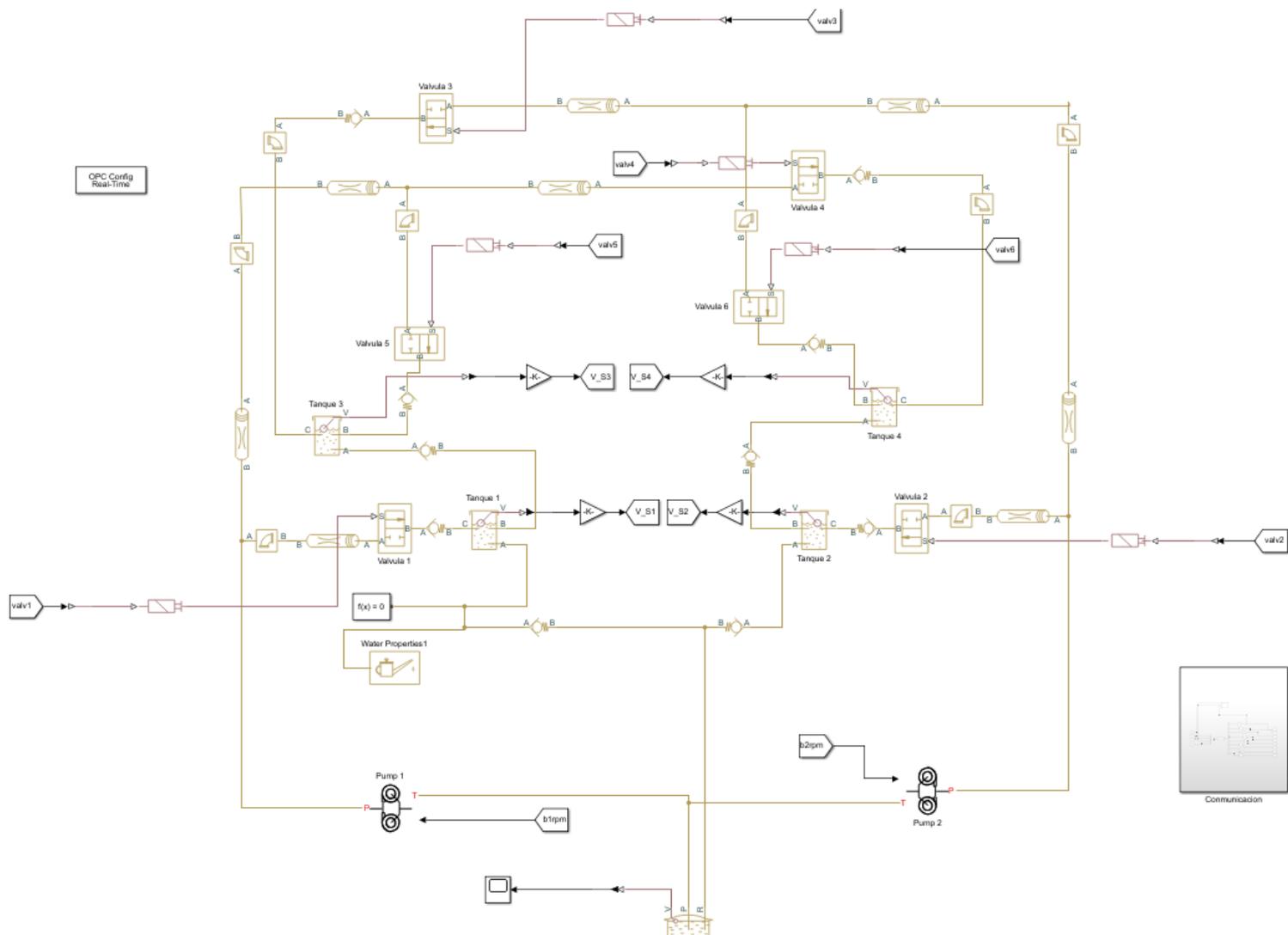


Figura 3-3. Modelo final

Por su parte la comunicación, establecida como subsistema, envía al servidor los volúmenes de agua de los 4 tanques y recibe los valores de velocidad de las bombas junto con las válvulas que se encuentran abiertas en cada momento. Esto queda representado en la figura 3-4.:

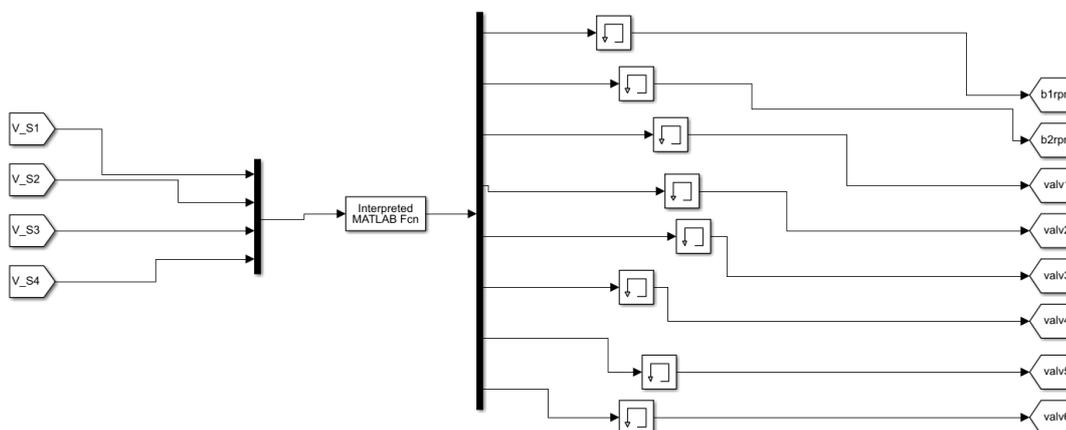


Figura 3-4: Comunicación del modelo final

La función que aparece en la figura 3-4 como “Interpreted MATLAB Fcn” engloba en una función de Matlab, las variables de entrada y salida del servidor mencionadas recientemente. Los pormenores de la configuración de la función y otros aspectos sobre la comunicación se encuentran detallados en el anexo 1.

4 CONTROL DEL SISTEMA

A continuación, una vez que el modelo ha quedado claramente definido, se procede a crear un controlador que alcance el nivel de agua del tanque deseado.

4.1 Control PID

En primer lugar, cabe destacar que el control de agua se realizará exclusivamente sobre los tanques intermedios, es decir, los tanques 1 y 2 siguiendo la nomenclatura que se ha utilizado hasta ahora.

Se debe explicar cómo implementar un control PID gracias a la ayuda de bloques predeterminados. Para el control PID, Codesys implementa un algoritmo basado en la suma de tres componentes tal y como aparece en la figura 4-1.

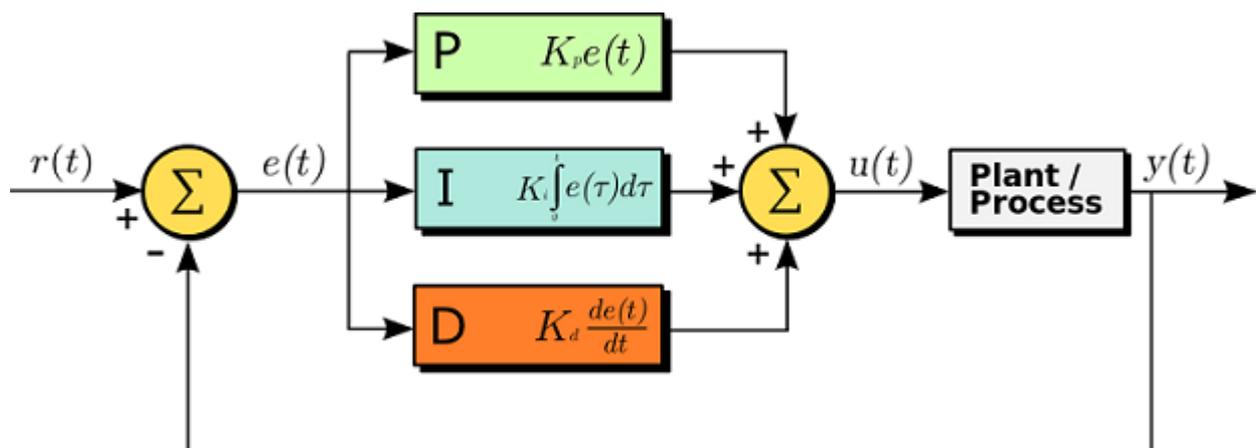


Figura 4-1. Algoritmo PID implementado por Codesys

Las tres componentes que intervienen en el control serán:

- El error de la variable a controlar con respecto a la referencia por una constante proporcional
- La integral del error en el tiempo (viene a ser el error acumulado) por una constante integral
- La derivada del error (variación del error en el instante) por una constante derivativa

En este caso, bastará con el bloque PID_FIXCYCLE que facilita Codesys. Las entradas y salidas del bloque quedan explicadas detalladamente en la tabla 4-1 que se encuentra debajo de la figura.

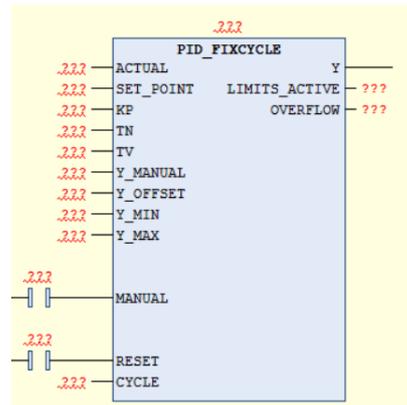


Figura 4-2. Bloque PID_FIXCYCLE

ENTRADA/SALIDA	FUNCIÓN
ACTUAL	Variable que se quiere controlar En nuestro caso sería el nivel de agua de los tanques intermedios
SET_POINT	Valor de referencia que se desea alcanzar en la variable a controlar que aparece en "ACTUAL"
KP	Constante proporcional del controlador
TN	Tiempo integral del controlador PID
TV	Tiempo derivativo del controlador PID
Y_MANUAL	Valor que se copia a la salida Y si la variable MANUAL está activa
Y_OFFSET	Offset para la variable manipulada Y
Y_MIN	Valor mínimo de saturación de la salida
Y_MAX	Valor máximo de saturación de la salida
MANUAL	Variable booleana que activa o desactiva el modo manual
RESET	Variable booleana que copia el valor de offset en la salida cuando se encuentra activa
CYCLE	Tiempo fijo de ejecución del bloque PID
Y	Variable de salida En nuestro caso sería la velocidad de las bombas
LIMITS_ACTIVE	Variable booleana que se activa automáticamente si la salida alcanza los límites de saturación
OVERFLOW	Variable booleana que se activa cuando rebosa el número de datos en el controlador

Tabla 4-1. Parámetros del bloque PID_FIXCYCLE

4.2. Configuraciones del sistema

Una vez que quedan descritas las variables del controlador PID en detalle, se procede a explicar las distintas configuraciones adoptadas por el sistema en función de las válvulas abiertas y cerradas. Cada una de estas configuraciones requerirá un control distinto.

4.2.1. Primera configuración

Esta primera disposición se caracteriza por tener una sola válvula abierta (válvula 1) quedando el resto de las válvulas todas cerradas. Además, solo estará encendida la bomba 1, quedando la bomba 2 apagada (velocidad = 0 rpm). De esta manera, el agua solo circulará por el tanque 1 y el tanque acumulador tal y como se refleja en la figura explicativa 4-3.

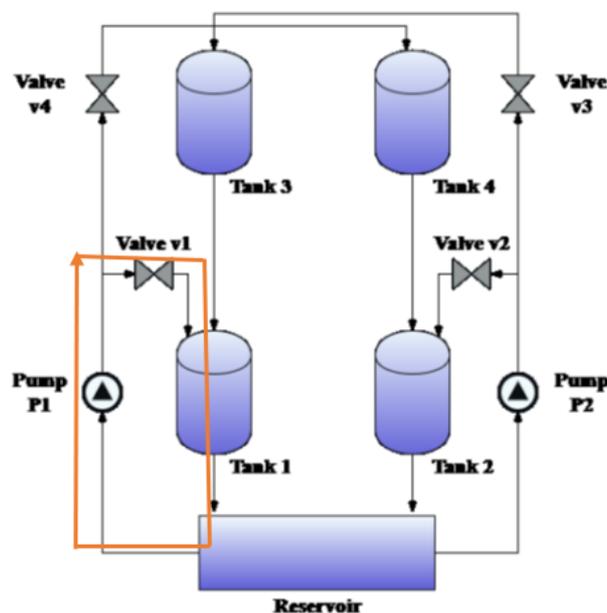


Figura 4-3. Esquema configuración 1

4.2.1.1. Control de la primera configuración

Esta disposición corresponde a un sistema de primer orden, para la que se calcularán los parámetros del controlador K_p , T_n y T_v . Primero se deducirán de manera teórica, ajustándolos posteriormente mediante ensayo de prueba y error.

Tras hacer un ensayo de respuesta frente a escalón del sistema en bucle abierto, que se encuentra representado en la figura 6-4 (capítulo de resultados), se han obtenido los parámetros característicos del sistema:

- $\tau_{ba} = 143$
- $K = 6.7e^{-2}$
- El sistema queda como $G(s) = \frac{0.067}{143s+1}$

A continuación, se aplica cancelación de dinámicas mediante el uso de un controlador PI:

$C(s) = \frac{Kp(T_n s + 1)}{T_n s}$ si $T_n = \tau = 143$ el sistema en bucle cerrado quedaría como:

$$Gc(s) = \frac{Kp \times K / T_n s}{1 + \frac{Kp \times K}{T_n}} = \frac{Kp \times K}{T_n s + Kp \times K} = \frac{1}{\frac{T_n}{Kp \times K} s + 1}$$

En este caso la constante de tiempo del sistema en bucle cerrado sería: $\tau_{bc} = \frac{T_n}{K \times Kp}$

Si se establece como especificación del sistema un tiempo de establecimiento de 400s (al 98%), la constante de tiempo en bucle cerrado quedaría como 100s ($T_{est} = 4\tau$):

$$\tau_{bc} = \frac{T_n}{K \times Kp} = 100 \rightarrow Kp = \frac{143}{0.067 \times 100} = 21.34$$

Tras introducir el término derivativo para mejorar el transitorio y ajustar los parámetros mediante ensayos de prueba y error, los parámetros finales del controlador serían:

- $Kp = 21.3$
- $Tn = 120$
- $Tv = 15$

Por lo tanto, el bloque del controlador en Codesys para la primera configuración sería el siguiente:

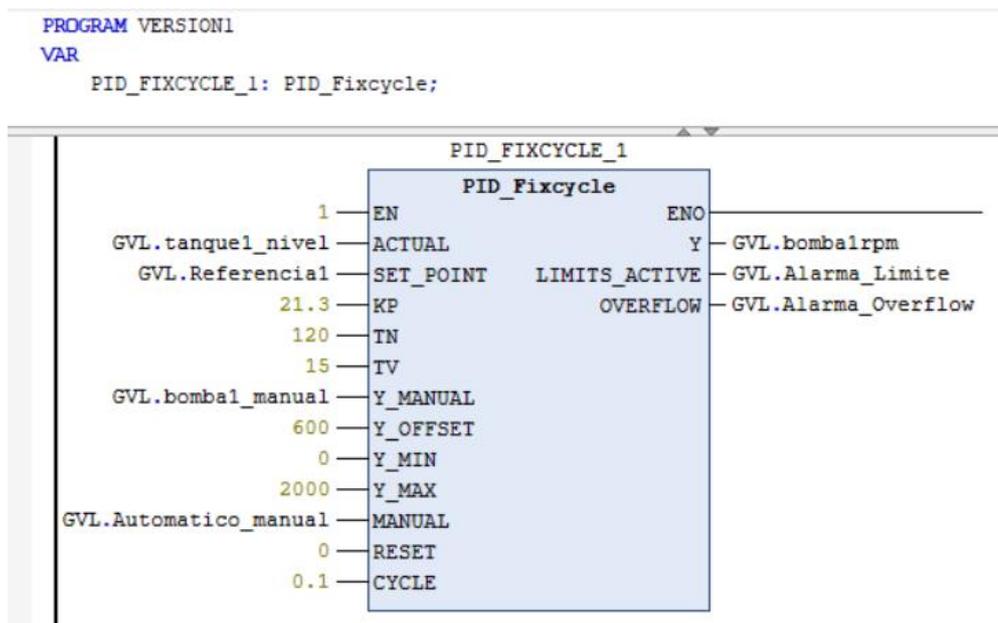


Figura 4-4. Controlador PID de la configuración 1

Cabe destacar que el “Cycle time” debe ser muy pequeño (del orden de 0.1 como aparece en la figura) para que el controlador se asemeje lo máximo posible a uno ideal.

4.2.2. Segunda configuración

La siguiente disposición se establece de manera que la válvula 5 (aquella que conecta la bomba 1 con el tanque 3) quedaría abierta y el resto de las válvulas cerradas. Al igual que en el caso anterior, la bomba 1 quedaría encendida y la bomba 2 apagada. Se adjunta en la figura 4-5. un esquema clarificador.

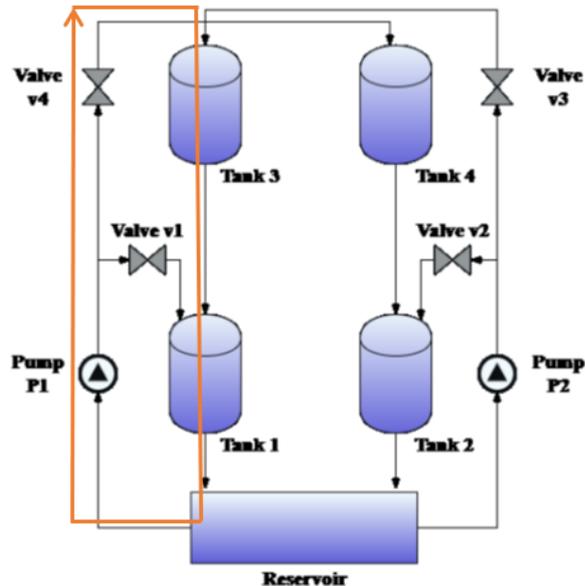


Figura 4-5. Esquema configuración 2

4.2.1.1. Control Ziegler-Nichols

En este caso el sistema sería de segundo orden. Tras hacer un ensayo frente a escalón en bucle abierto se obtiene que el sistema es sobreamortiguado. Esto quiere decir que el sistema no tiene sobreoscilación en bucle abierto, sino que tiene un pequeño retardo y se acerca a la ganancia en régimen permanente sin llegar a alcanzarla. De manera analítica, esto se explica debido a que el sistema tiene dos polos, uno de ellos muy dominante (cerca del origen) con respecto al otro.

Para controlar el sistema, se obtiene previamente la respuesta frente a escalón en bucle abierto de manera que se puedan calcular los parámetros característicos:

- La ganancia (K) sería el valor en régimen permanente
- El retardo (L) sería el corte de la tangente en el punto de cambio de curvatura (máxima pendiente) con el eje de abscisas.
- El tiempo de subida (tau) será el tiempo que transcurre desde el retardo hasta el corte de la tangente anterior con el nivel de la ganancia.

Se adjunta la figura 4-6. como ejemplo clarificador de los parámetros mencionados:

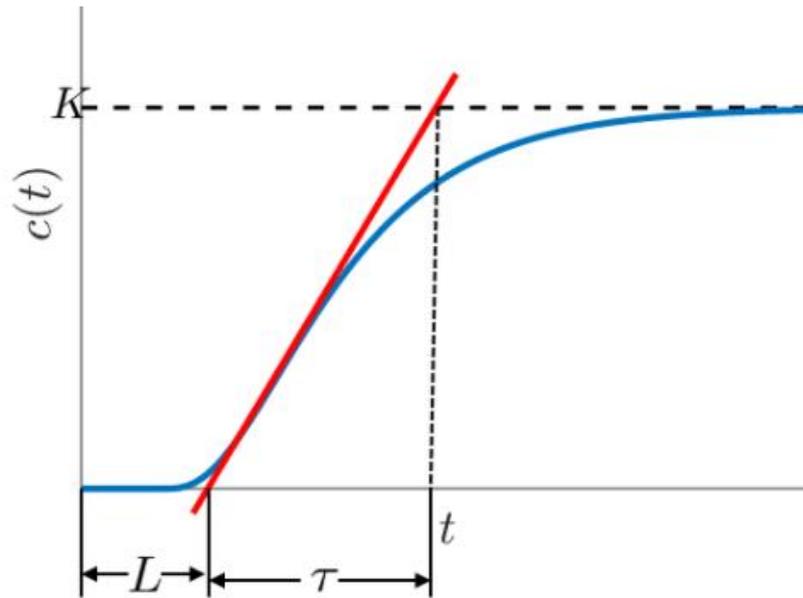


Figura 4-6. Parámetros Ziegler-Nichols en bucle abierto

En nuestro caso, los parámetros de la respuesta se obtienen analizando la figura mostrada en la figura 6-6. (capítulo de resultados). Estos parámetros serían:

- $\tau = 1468.9$
- $L=44.6$
- $K=13.31$

Controlador	K_p	τ_i	τ_d
P	$\frac{\tau}{KL}$	∞	0
PI	$0.9 \frac{\tau}{KL}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{\tau}{KL}$	$2L$	$0.5L$

Figura 4-7. Tabla Ziegler-Nichols en bucle abierto

Teniendo en cuenta la tabla de Ziegler-Nichols que se adjunta en la figura 4-7, los parámetros del controlador serían:

- $K_p = 1.2 \times \frac{1486.9}{13.31 \times 44.6} = 3.01$
- $T_n = 2 \times 7.546 = 89.02$
- $T_v = 0.5 \times L = 22.3$

4.2.1.2. Control mediante especificaciones

En este subapartado se plantea otro método de control que se contrastará con el método anterior. De la respuesta frente a escalón en bucle abierto se puede obtener la ganancia estática y el tiempo de establecimiento del sistema. Además, se sabe que al ser un sistema sobreamortiguado, un polo es dominante frente al otro por lo que se puede estimar el polo no dominante como 15 veces el polo dominante (el polo dominante será aquel que se encuentra más cerca del origen).

Con toda la información previa solo queda definir el sistema en bucle abierto como:

$$G(s) = \frac{K \times \omega_n^2}{s^2 + 2\xi \omega_n s + \omega_n^2} = \frac{K \times \sigma_1 \times \sigma_2}{(s + \sigma_1) \times (s + \sigma_2)} \text{ siendo } \sigma_1 \text{ el polo dominante y } \sigma_2 \text{ el polo no dominante}$$

Analizando la respuesta frente a escalón en bucle abierto de la figura 6-6. se obtiene que:

- $K = 0.06655$
- $\sigma_1 = 2.19 \times 10^{-3}$
- $\sigma_2 = 0.03285$

Una vez que se obtienen estos valores, se opera para escribir el sistema como uno de segundo orden en forma estándar:

$$G(s) = \frac{4.7877e^{-6}}{s^2 + 3.504e^{-2}s + 7.194e^{-5}}$$

A partir de este momento se pasa a crear un sistema en bucle cerrado añadiendo un controlador que, en principio, solo incluirá término proporcional e integral. Por lo tanto, el sistema en bucle cerrado quedaría definido mediante la siguiente expresión:

$$G_c(s) = \frac{K(K_p s + K_i)}{s^3 + a s^2 + (b + K \times K_p) s + K \times K_i}$$

(siendo K_p y K_i las constantes proporcional e integral del controlador respectivamente)

Por otra parte, también es conocido que la ecuación característica de un sistema de segundo orden estándar se escribe de la siguiente forma:

- $s^3 + (\beta + 2)\xi \omega_n s^2 + (2\beta\xi^2 + 1)\omega_n^2 s + \beta\xi \omega_n^3 = 0$
- $\beta = \text{factor de proporcionalidad entre polos del sistema}$
- $\xi = \text{factor de amortiguamiento } (0 < \xi < 1)$
- $\omega_n = \text{frecuencia natural del sistema}$

Igualando el denominador del sistema en bucle cerrado con la ecuación característica estándar se pueden despejar los parámetros requeridos:

$$\beta = \frac{a}{\xi \omega_n} - 2$$

$$K_p = \frac{(2\beta\xi^2 + 1)\omega_n^2 - b}{k}$$

$$Ki = \frac{\beta \xi \omega_n^3}{k} \quad (Ki = \frac{Kc}{Tn})$$

A partir de ahora se pueden definir las especificaciones de diseño. Para el sistema en cuestión, se ha decidido establecer:

- $SO = 25\%$
- $T_{establecimiento} = 1700s$

En primer lugar, se calcula el factor de amortiguamiento y la frecuencia natural:

$$\xi = \frac{\ln\left(\frac{SO}{100}\right)^2}{\sqrt{\pi^2 + \ln\left(\frac{SO}{100}\right)^2}} = 0.280$$

$$\omega_n = \frac{4}{\xi \times T_{establecimiento}} = 8.403e^{-3}$$

El siguiente paso será calcular los parámetros del controlador cuyas expresiones han sido definidas previamente:

$$\beta = \frac{2.496e^{-2}}{0.4037 \times 7.927e^{-3}} = 12.89$$

$$Kp = \frac{(2 \times 5.80 \times 0.4037^2 + 1) \times 7.92e^{-3} - 3.65e^{-5}}{2.041e^{-6}} = 29.53$$

$$Ki = \frac{5.80 \times 0.4037 \times 7.927e^{-3}}{2.041e^{-6}} = 0.447$$

$$Tn = \frac{Kp}{Ki} = \frac{71.11}{0.5714} = 66.06$$

Finalmente, teniendo en cuenta ambos modelos de control, introduciendo el parámetro derivativo y gracias a los ajustes de los parámetros mediante ensayos de prueba y error, los parámetros del controlador PID quedarían definidos como:

- $Kp=1$
 - $Tn=80$
 - $Td=6$

Por lo tanto, el controlador PID para la segunda configuración quedaría del siguiente modo:

```

PROGRAM VERSION2
VAR
    PID_FIXCYCLE_0: PID_FIXCYCLE;
END_VAR
    
```

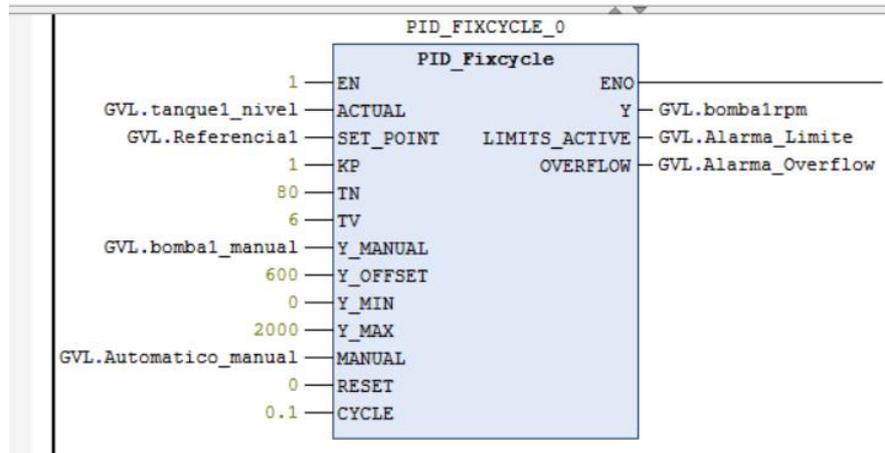


Figura 4-8. Controlador PID de la configuración 2

4.2.3. Tercera configuración

En esta última configuración, las válvulas 1,2,3 y 4 quedarían abiertas mientras que las válvulas 5 y 6 permanecerían cerradas. Por lo tanto, la bomba 1 alimentaría directamente el tanque 1 y el 4 mientras que la bomba 2 haría lo propio con el tanque 2 y 3 tal y como se observa en la figura 4-9.

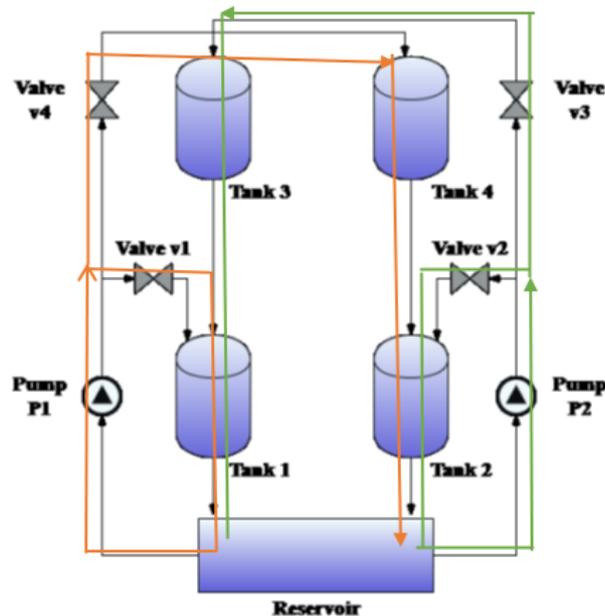


Figura 4-9. Esquema configuración 3

4.2.3.1. Control de la tercera configuración

En este caso la forma de plantear el control varía sensiblemente con respecto a las dos anteriores. En primer lugar, existen dos variables a controlar en lugar de una (volúmenes de los tanques 1 y 2) para las que se deben establecer dos valores distintos de referencia (referencia 1 y 2), además de dos variables de salida simultáneamente (velocidades de las bombas 1 y 2).

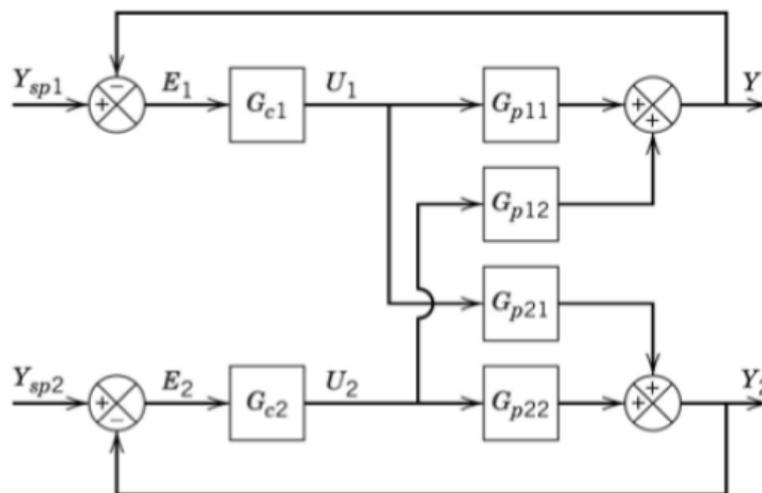
Por otro lado, la bomba 1 alimenta el tanque 1 directamente y el tanque 2 de manera indirecta (a través del tanque 4). Análogamente, la bomba 2 realiza lo mismo de manera análoga. Esto indica que independientemente del control que se establezca, existirá un breve acoplamiento en ambos tanques que deberá ser amortiguado correctamente.

Para definir los controladores se tiene en cuenta que el modelo creado en Simulink es totalmente simétrico, por lo que esta misma simetría se verá reflejada en los controladores. Se decide modelar el conjunto como dos modelos de primer orden (exactamente iguales), emparejando los volúmenes de los tanques 1 y 2 como variables de entrada al control de las velocidades de las bombas 1 y 2 respectivamente.

Al igual que en las configuraciones anteriores se obtiene una respuesta de los sistemas frente a escalón en bucle abierto a partir del punto de funcionamiento. En nuestro caso, el punto de funcionamiento se establece con ambas bombas a 600 rpm (con las cuatro válvulas abiertas). Tras observar la respuesta frente a varios escalones en ambas bombas, que se encuentra representada en la figura 6-8. (capítulo de resultados), se decide definir los sistemas a través de su ganancia y su constante de tiempo tal y como se ha realizado en las configuraciones anteriores. Los modelos quedarían como:

$$G_{p11}(s) = G_{p22}(s) = \frac{0.09}{240s + 1}$$

Como se ha mencionado anteriormente, una vez que se han definido los sistemas, el emparejamiento será bomba1-tanque1 y bomba2-tanque2 mediante sistemas de primer orden. Como clarificación de esta idea se adjunta la figura 4-10 donde se observan estos emparejamientos.



Emparejamiento (1,1)-(2,2)

Figura 4-10. Emparejamientos de la tercera configuración

Adaptando la figura anterior a nuestro caso:

- Gp11 → tanque 1 alimentado por bomba 1
- Gp12 → tanque 1 alimentado por bomba 2 de manera indirecta
- Gp21 → tanque 2 alimentado por bomba 1 de manera indirecta
- Gp22 → tanque 2 alimentado por bomba 2

Estos emparejamientos se han realizado en base a que el caudal de agua que llega al tanque 1 con el impulso de la bomba 1 es mucho mayor que el caudal que llega a través de la bomba 2 de manera indirecta. Análogamente sucede lo mismo para la bomba 2 y el tanque 2.

Tras varios ensayos de prueba y error, se asume que el acoplamiento entre bomba 1-tanque 2 y bomba 2-tanque 1 es muy bajo, así que no se emplearán técnicas de desacoplo parcial o desacoplo estático en nuestros sistemas por simplicidad ya que no es necesario para el control. Por lo tanto, el caudal que llega al tanque a través de la bomba del lado opuesto se entenderá como una perturbación que el controlador debe absorber cuando sea diseñado correctamente.

Una vez que la estrategia de control queda definida, se plantea un control por cancelación de dinámicas con un tiempo de subida en bucle cerrado=125s (especificación de diseño). Consecuentemente, los parámetros del controlador PI quedan definidos como:

$$Td = \tau = 240$$
$$Kc = \frac{\tau}{Kp \times \tau_{bc}} = \frac{240}{0.09 \times 125} = 21.3$$

A partir de estos primeros parámetros, los controladores (simétricos como queda comentado anteriormente) se empieza a probar en el sistema, incorporándole finalmente la comunicación. Tras varias pruebas, se afinan los parámetros y se incorpora un tiempo derivativo para la mejora del transitorio.

Los parámetros finales del controlador PID serían los siguientes:

- Kp=17
 - Tn=240
 - Td=10

A continuación, se adjunta como quedarían ambos controladores una vez que ya están definidos los parámetros en la figura 4-11:

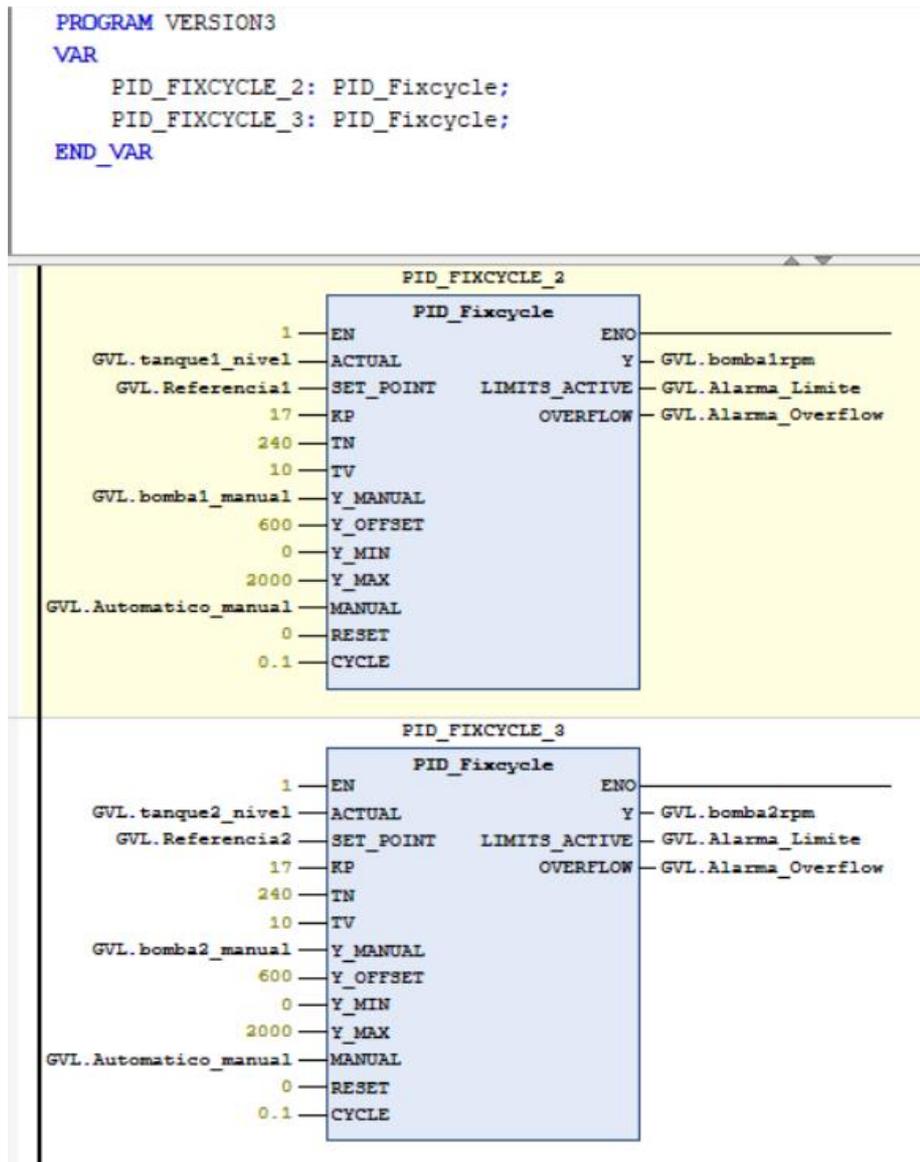


Figura 4-11. Controlador PID de la configuración 3

4.2.4. Coordinación de las tres configuraciones

Una vez que todas las configuraciones quedan definidas y explicadas adecuadamente, solo queda usar cada controlador en el caso adecuado.

La variable GVL\Configuracion es la encargada de coordinar esta tarea. Tomará el valor 1,2 o 3 y se transmitirá por el servidor OPC hasta Codesys, quien elegirá la versión correcta del controlador en cada caso. Para ello se ha creado una tarea principal "Main task" llamada PLC_PRG que aparece detallada en la figura 4-12:

```
PROGRAM PLC_PRG
VAR
END_VAR

IF GVL.Configuracion=1 THEN
    VERSION1 ();
ELSE IF GVL.Configuracion=2 THEN
    VERSION2 ();
ELSE IF GVL.Configuracion=3 THEN
    VERSION3 ();
END_IF
END_IF
END_IF
```

Figura 4-12. Tarea principal en Codesys

Se debe aclarar que “VERSION1”, “VERSION2” Y “VERSION3” son tres POU’s independientes que incluyen los controladores para las configuraciones 1, 2 y 3 respectivamente, las cuales se han visto en este capítulo.

5. SCADA

Este capítulo trata sobre el SCADA que se ha creado a través del programa Citect SCADA 2018. Se utilizará esta herramienta para supervisar, controlar, representar y adquirir datos relevantes sobre el sistema en tiempo real

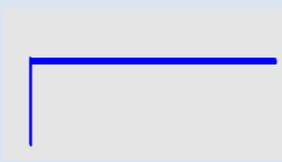
5.1. Componentes de las pantallas de explotación

En primer lugar, cabe destacar que la comunicación OPC-UA se establece como nuevo cliente que se une al servidor creado por Codesys. Para ello se debe incorporar un dispositivo nuevo de E/S en la topología del sistema. El Anexo 1 del proyecto incorpora una guía detallada de como establecer la comunicación OPC-UA con el servidor.

Una vez que la conexión está establecida, se crea una nueva pantalla de explotación en el editor de gráficos y se comienzan a incorporar todos los componentes necesarios.

5.1.1. Objetos básicos

Se adjunta una tabla donde se explica detalladamente cómo se han incorporado los elementos más básicos del sistema:

OBJETOS	SÍMBOLOS	DESCRIPCIÓN
Tuberías		Representan el flujo de agua, pero no tienen ningún evento asociado. Se usan distintos colores para distinguir los caudales proporcionados por la bomba 1 y 2
Tanques		Existen cuatro tanques con una barra azul que representa el volumen de agua en litros (de 0 a 90L). Además, el valor del volumen exacto en formato digital viene definido numéricamente tal y como aparece en el dibujo
Bombas		Símbolo que cambia de color (a verde) cuando la bomba comienza a funcionar (velocidad >0 rpm)

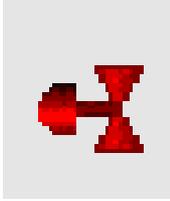
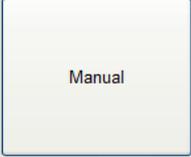
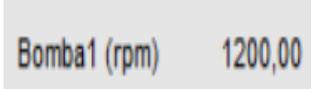
Válvulas		Válvulas de todo o nada. El color de la válvula (rojo o verde), representa si la válvula está completamente cerrada o abierta.
Botón de manual/automático		Pulsando el botón cambia de modo manual a automático y viceversa
Valores numéricos		Son valores digitales que no se pueden modificar manualmente, sino que dependen de la lógica implementada en los otros programas. Se utilizan para el valor real de la velocidad de las bombas y el de los tanques
Entradas digitales		<p>Estos valores aparecen digitalmente al igual que los anteriores, pero además se pueden modificar por teclado.</p> <p>En el caso de las bombas solo afecta al valor real si el sistema se encuentra en modo manual.</p> <p>Para los valores de referencia (setpoint), solo se podrá introducir para los tanques que se estén controlando. En el caso de las configuraciones 1 y 2, el setpoint2 no estará visible ni modificable.</p>

Tabla 5-1. Objetos básicos en Citect SCADA

5.1.2. Funciones Cicode

Cicode es un lenguaje de programación usado en el software Citect SCADA 2018 (y en tantas otras versiones de Citect) que se caracteriza por ser muy fácil de implementar en el caso de herramientas HMI o SCADA. Es un lenguaje muy parecido a 'C' pero se usa principalmente para el manejo de datos en tiempo real. Además, cabe destacar que se pueden utilizar todo tipo de variables, etiquetas, alarmas o lo que sea necesario.

Para acceder a las funciones Cicode se procede desde la barra de herramientas principal tal y como aparece en la figura 5-1.

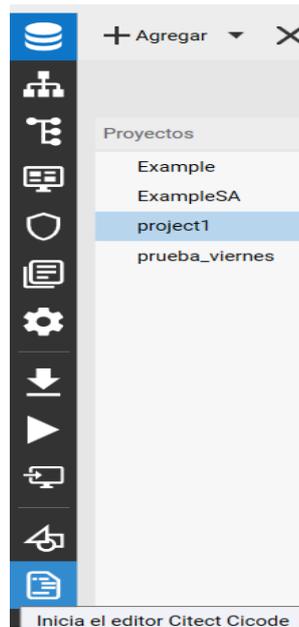


Figura 5-1. Pantalla para acceder a las funciones Cicode

En el caso del proyecto, se han creado funciones asociadas a eventos dentro de la pantalla de explotación, utilizando continuamente las variables globales que se transmiten a través del servidor OPC. En concreto se han creado dos funciones asociadas a la variable GVL\Configuracion.

La primera función sirve para crear una pequeña pantalla emergente con una lista desplegable que permite elegir la configuración 1, 2 o 3 del sistema. Una vez completada la operación, el valor se asocia a la variable GVL\Configuracion y se transmite a través del servidor OPC en tiempo real.

A continuación, se adjunta la figura donde se observa esta función.

```

FUNCTION FnMenu()
  STRING      sBuf;
  FormNew("Selecciona una configuración",30,6,1);
  FormComboBox(2,2,20,5,sBuf,1);
  FormAddList("Configuracion 1");
  FormAddList("Configuracion 2");
  FormAddList("Configuracion 3");
  FormRead(0);
  IF sBuf="Configuracion 1" THEN
    GVL\configuracion=1;
  END
  IF sBuf="Configuracion 2" THEN
    GVL\configuracion=2;
  END
  IF sBuf="Configuracion 3" THEN
    GVL\Configuracion=3;
  END
END

```

Figura 5-2. Función Cicode menú desplegable

Esta función se utiliza para elegir la configuración del sistema en el momento en que se inicie la pantalla de explotación. Esto se realiza agregando el nombre de la función en la página del editor de gráficos ya creada: Archivo\Propiedades\Eventos\En entrada de página.

Por otra parte, también se asocia la misma función a la flecha que aparece en el objeto de la figura 5-3 de manera que se pueda modificar la configuración en todo momento, tan solo pulsando la flecha.



Figura 5-3. Objeto gráfico de la configuración

La segunda función implementa cierta lógica sobre la configuración del sistema. Dependiendo de la configuración que se haya elegido (1,2 o 3) se activan las válvulas correspondientes entre las seis existentes. Todas las etiquetas se asocian a las variables del servidor OPC para que la comunicación sea instantánea.

Se adjunta la figura sobre cómo quedaría la función de configuración de válvulas:

```
FUNCTION configurarvalvulas ()
  IF GVL\Configuracion=0 THEN
    GVL\valvula1=0;
    GVL\valvula2=0;
    GVL\valvula3=0;
    GVL\valvula4=0;
    GVL\valvula5=0;
    GVL\valvula6=0;
  END
  IF GVL\Configuracion=1 THEN
    GVL\valvula1=1;
    GVL\valvula2=0;
    GVL\valvula3=0;
    GVL\valvula4=0;
    GVL\valvula5=0;
    GVL\valvula6=0;
  END
  IF GVL\Configuracion=2 THEN
    GVL\valvula1=0;
    GVL\valvula2=0;
    GVL\valvula3=0;
    GVL\valvula4=0;
    GVL\valvula5=1;
    GVL\valvula6=0;
  END
  IF GVL\Configuracion=3 THEN
    GVL\valvula1=1;
    GVL\valvula2=1;
    GVL\valvula3=1;
    GVL\valvula4=1;
    GVL\valvula5=0;
    GVL\valvula6=0;
  END
END
```

Figura 5-4. Función Cicode configuración de válvulas

Esta función, al igual que en la anterior, se ejecutará en la entrada de página y en la flecha de la configuración de la figura 5-3, de manera que cada vez que se actualice el valor de la configuración, se actualice de igual forma la activación de las válvulas.

5.1.3. Representación de la evolución de las variables

Para la correcta interpretación de las variables del sistema, se ha utilizado una página en el editor de gráficos dentro del mismo proyecto del SCADA sobre la evolución de:

- las variables que se desean controlar.
- los actuadores.

En esta página, se utiliza el objeto ‘tendencia’ que permite representar esta evolución (se encuentra dentro de las herramientas principales de la página). Una vez creada, se divide en dos paneles distintos:

- El primero de ellos representa la evolución de los niveles de agua de los tanques y los niveles de referencia a los que se debe llegar.
- El segundo representa las velocidades de las bombas en revoluciones por minuto.

Se ha separado la representación en estos dos paneles ya que las escalas son completamente distintas, por lo que no tendría sentido representar todas las variables conjuntamente.

En ambos paneles, una vez que se han agregado todas las variables correspondientes, se pueden ocultar o visualizar las variables correspondientes en función de la configuración que se esté utilizando de manera manual.

En la figura 5-5 se muestra cómo guardar y cargar un formato concreto de representación de variables por defecto (incluyendo las variables concretas que se quieren representar, las que quedan ocultas, las propiedades de la gráfica y el tiempo de representación).

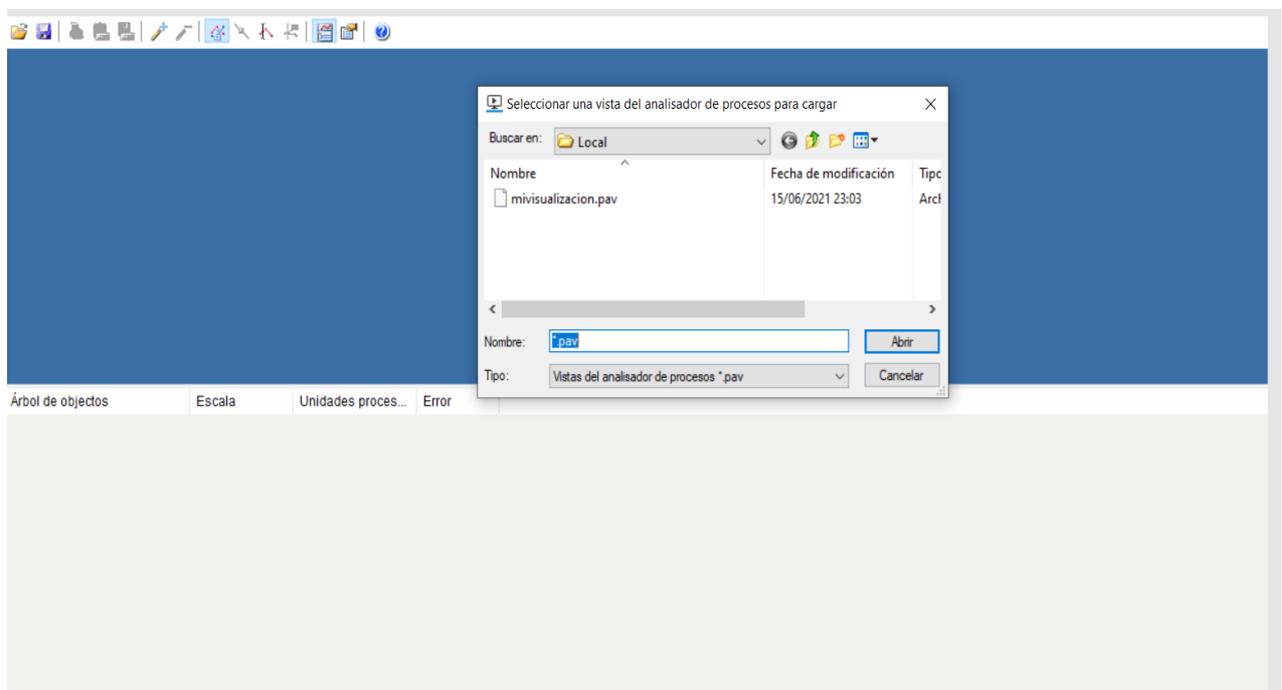


Figura 5-5. Guardar y cargar la representación de variables

Cabe destacar que, dentro de las propiedades, se ha decidido escalar de manera automática las variables, no mostrar ejes secundarios y diferenciar las curvas con valores distintos por claridad de las figuras.

5.2. Pantallas finales de explotación

Una vez que se han explicado todos los objetos y propiedades empleados, se procede a mostrar cómo quedarían las pantallas de explotación.

En primer lugar, se debe mencionar una vez más que el proyecto está comunicado mediante el servidor OPC creado a través del programa de Codesys y todas las variables que se emplean en los objetos de las pantallas finales son transmitidos en tiempo real mediante este servidor.

La primera pantalla sería la de representación de datos. En la figura 5-7, se muestra cómo quedaría la representación final en dos paneles como se ha mencionado anteriormente.

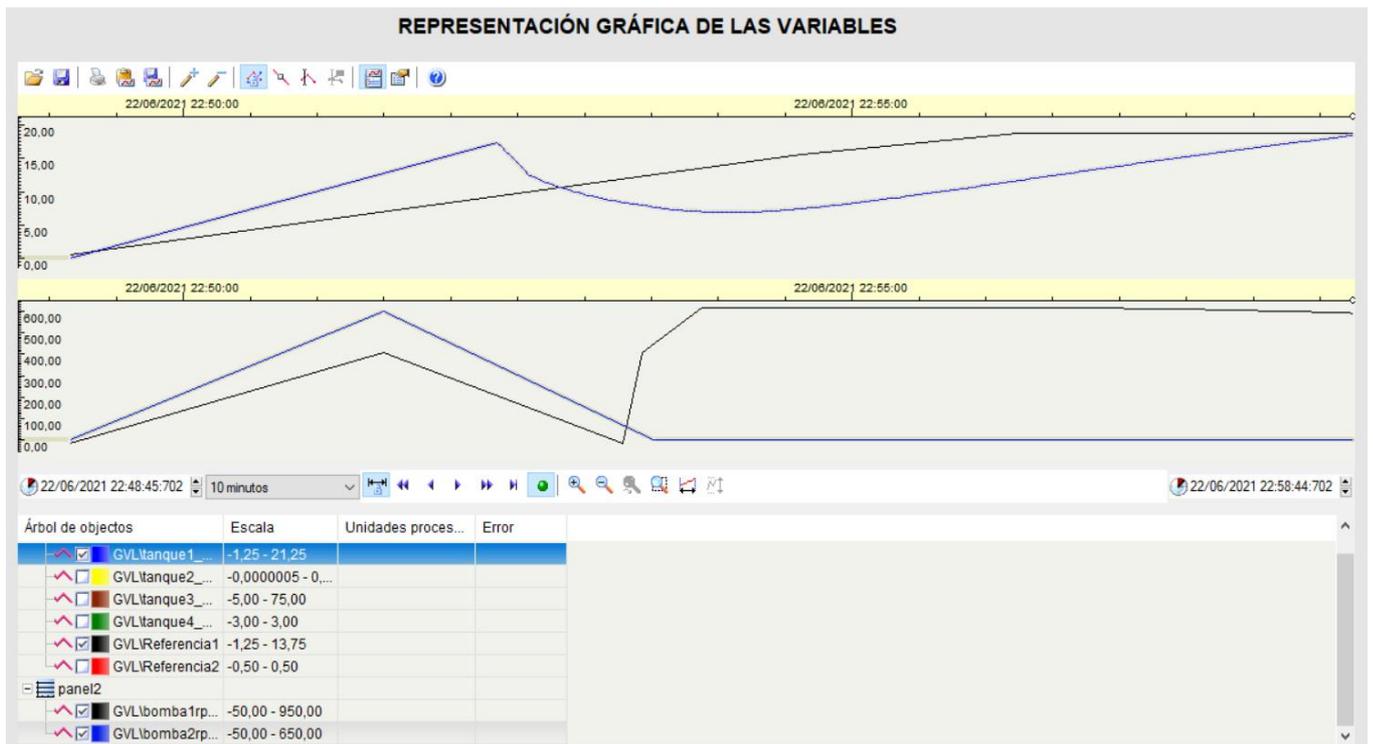


Figura 5-6. Representación de la evolución de variables

Se ha utilizado una variedad de paneles y colores como se observa en el árbol de objetos con el objetivo de diferenciar claramente la evolución de cada una de las variables.

En cualquier caso, los colores, las curvas visibles y no visibles, las escalas de valores, el tiempo de muestro y el resto de las propiedades se pueden actualizar en tiempo real si se cree necesario. Para que estas características se conserven a lo largo del tiempo se deben guardar como se ha explicado anteriormente.

La segunda pantalla que se desea explicar sería la pantalla principal, que incluye todos los objetos básicos definidos anteriormente junto con las funciones Cicode de manera indirecta. En la figura 5-8 se define cómo quedaría el modelo final.

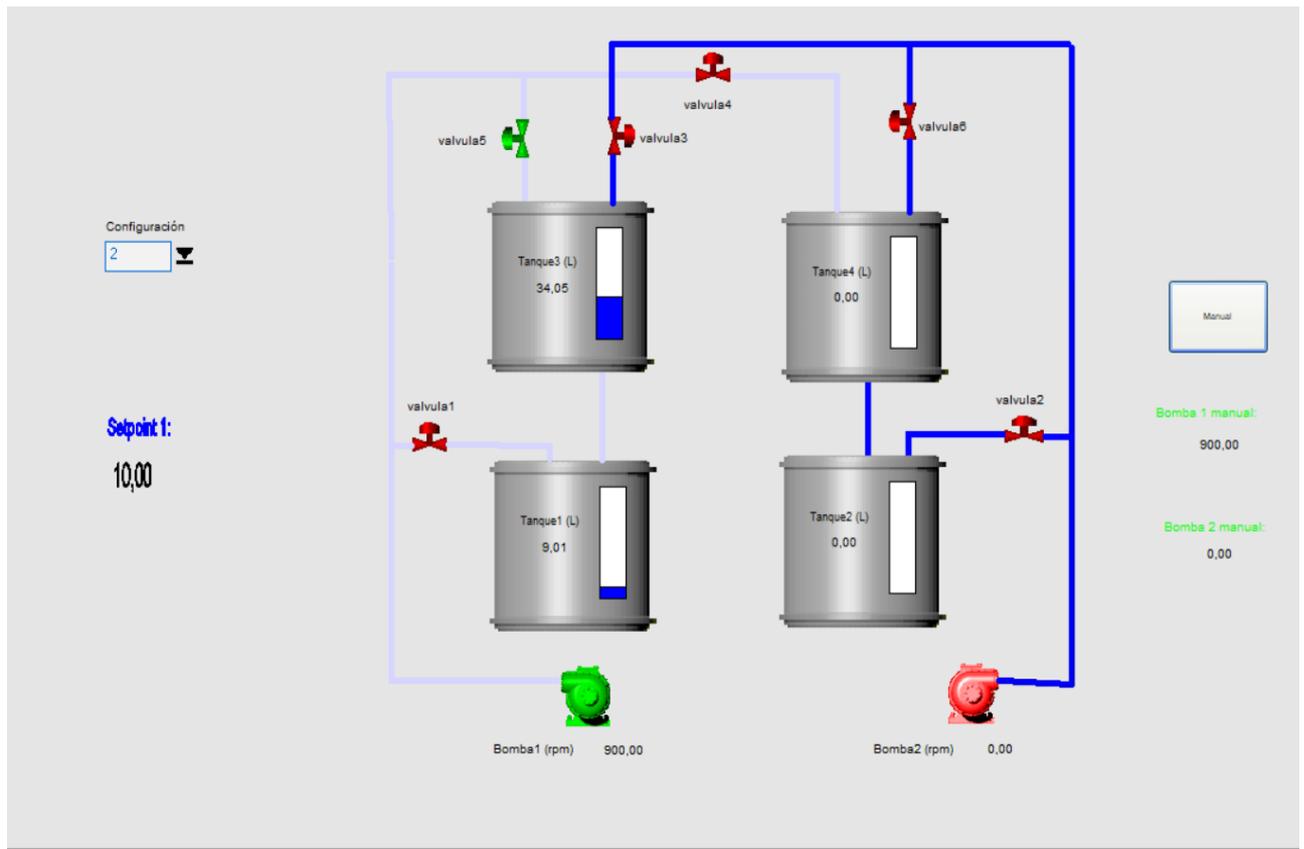


Figura 5-7. Pantalla principal de explotación

Con esto quedan explicadas las dos pantallas finales que aparecen en la ejecución del SCADA.

6. RESULTADOS Y CONCLUSIONES

Si buscas resultados distintos no hagas siempre lo mismo

Albert Einstein

En este capítulo se mostrarán los resultados obtenidos en cada una de las simulaciones realizadas hasta el momento para cada una de las tres configuraciones en las que se ha trabajado. Finalmente se añadirán unas conclusiones generales sobre el desarrollo del trabajo.

6.1. Resultados de las simulaciones

Todas las pruebas y simulaciones que se han realizado y que se mostrarán a partir de ahora conllevan la ejecución simultánea de los tres programas (Codesys, Citect y Matlab), de la cual se han obtenido resultados en tiempo real a través de la herramienta “Scope” en Matlab & Simulink.

Cabe destacar que cada vez que se ha deseado ejecutar los tres programas simultáneamente, el PLC virtual correspondiente a Codesys ha tenido que estar activo (dura activo 120 minutos sin interrupciones), mientras que por su parte el modelo de Matlab se ha ejecutado mediante el botón “run” con “Stop time” infinito.



Figura 6-1. Iniciar PLC virtual



Figura 6-2. Iniciar el modelo de Matlab&Simulink

Por su parte, el proyecto de Citect SCADA se ha ejecutado desde la página principal del proyecto en modo demostración (permanece activo durante 15 min) y una vez en ejecución, se ha elegido el menú y la “PageList” que aparece en la figura 6-3. Dentro de esta, se selecciona la página de explotación que sea conveniente.

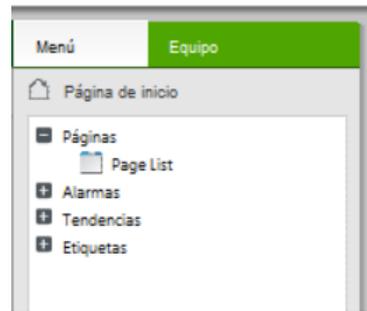


Figura 6-3. PageList en Citect SCADA

Para la recogida de datos se podría haber utilizado la herramienta de base de datos de Citect SCADA 2018, para la cual se hubiera necesitado crear un usuario (y en consecuencia una licencia). En este trabajo, se ha utilizado el modo Demo por lo que se ha decidido no obtener los resultados por este método.

Una vez aclarado el método de recogida de resultados, se procede a analizarlos en el caso de cada una de las configuraciones.

6.1.1. Resultados de la primera configuración

Para controlar el sistema en la primera configuración, se debe diseñar el modelo tal y como quedó explicado en capítulos anteriores. Se recuerda que en esta configuración solo queda abierta la válvula 1 y solo entra agua en el tanque 1.

En primer lugar, se establece un setpoint en régimen permanente, que en este caso será con la bomba 1 a 600 rpm y la bomba 2 apagada (por definición). En este setpoint, el tanque 1 alcanzará un valor permanente de 17.33L, mientras que el resto de los tanques permanecerán completamente vacíos (salvo el acumulador).

A partir de este momento, se establece un escalón en bucle abierto en la salida, desde 600 hasta 800 rpm en la bomba 1. La respuesta del volumen del tanque 1 será la que aparece en la figura 6-4.

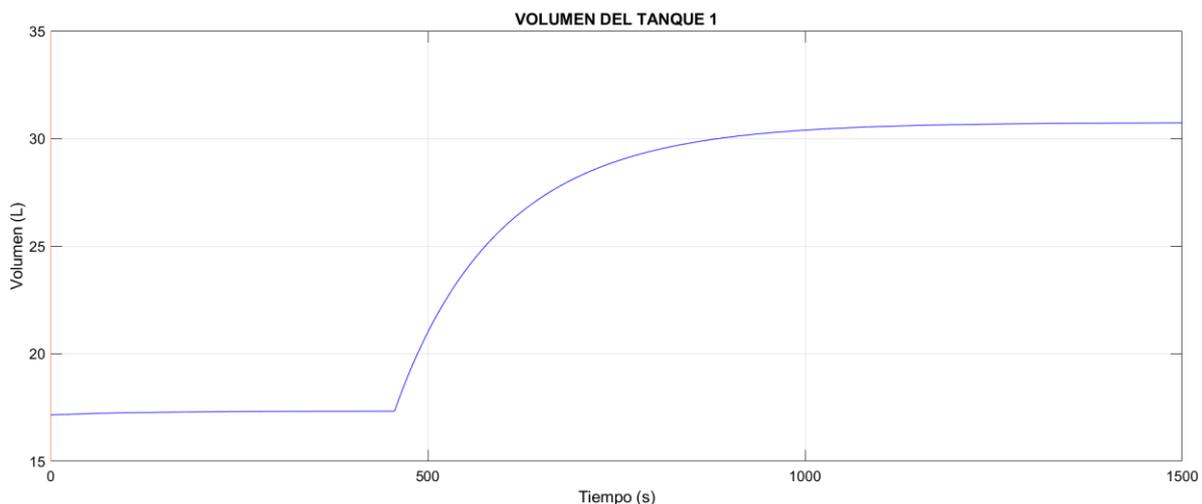


Figura 6-4. Respuesta ante escalón en la configuración 1

De esta respuesta se obtienen los valores del modelo de primer orden creado ya mencionados anteriormente:

- $K=0.067$
- $\tau = 143$

Una vez que se define el controlador, ya se puede exponer el sistema a un control en bucle cerrado con varios cambios de referencia en el volumen del tanque 1. La respuesta del sistema frente a varios cambios de referencia en ambos sentidos se observa en la figura 6-5.

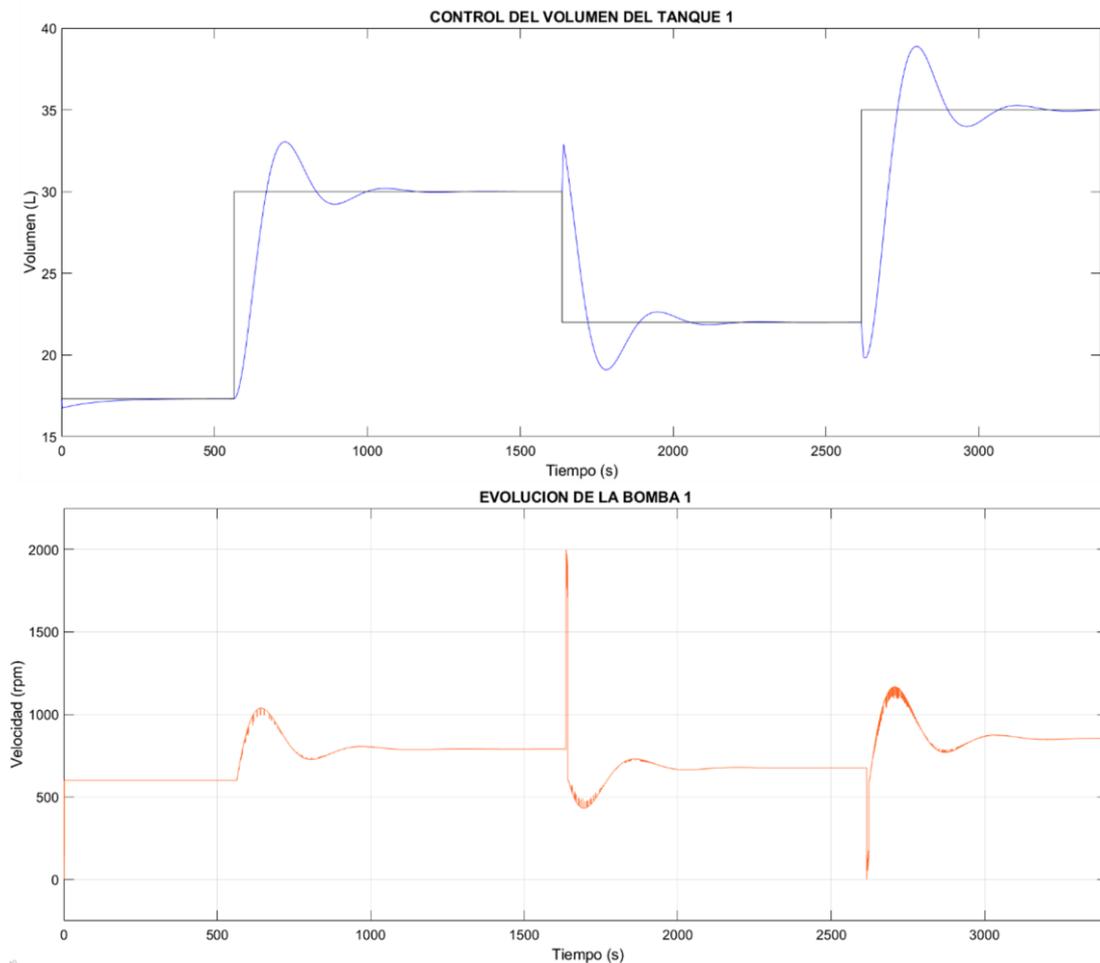


Figura 6-5. Respuesta del sistema en bucle cerrado en la configuración 1

Como resultado, se observa una respuesta satisfactoria al cambio de referencia, donde se disminuye el tiempo de establecimiento considerablemente respecto al régimen permanente (queda entorno a 400s) y la sobreoscilación de la salida no supera el 25% del incremento de la referencia.

6.1.2. Resultados de la segunda configuración

Para la controlar la segunda configuración, se recuerda que solo quedaría abierta la válvula 5 por lo que el agua solo circularía por los tanques 1 y 3. De esta manera el sistema quedaría de segundo orden.

De igual modo que en la configuración anterior, se lleva el sistema a régimen permanente con la bomba 1 a 600 rpm. En este setpoint, los volúmenes de los tanques quedarían como:

- $V_{tanque\ 1} = 17.19L$
- $V_{tanque2} = 45L$

De nuevo, se está en condiciones de estudiar la respuesta frente a un escalón en la bomba 1 desde 600 hasta 800 rpm como aparece en la figura 6-6.

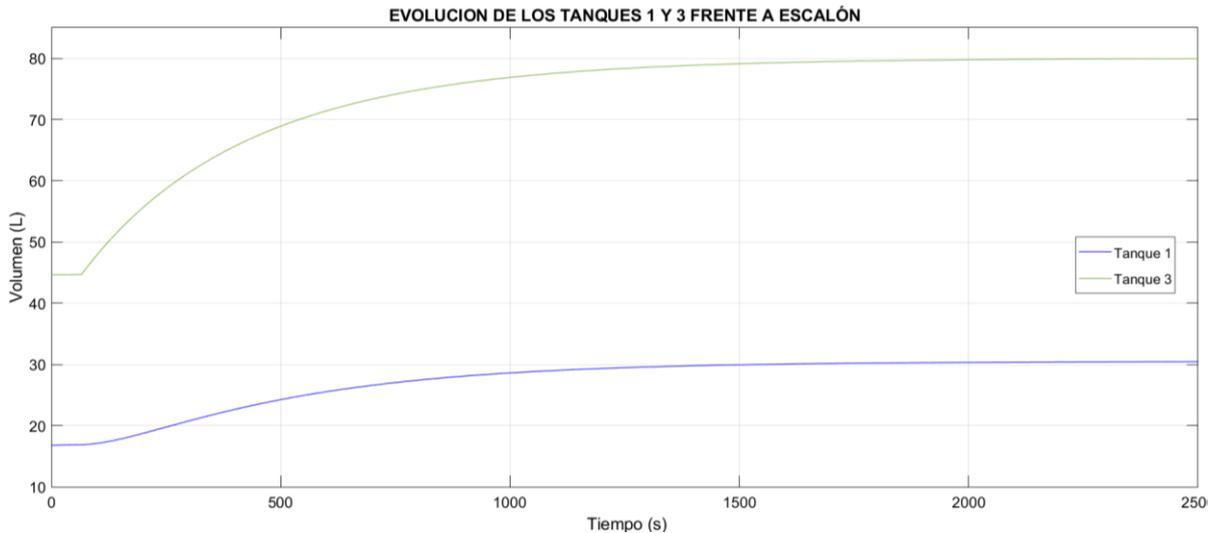


Figura 6-6. Respuesta ante escalón en la configuración 2

Aunque en la figura anterior aparece la respuesta de los dos volúmenes (tanque 1 y 2), el estudio se hará sobre la variable a controlar, que en este caso sería el tanque 1. Por lo tanto, como ya se mencionó en el capítulo 4, estudiando la respuesta el modelo queda definido por:

- $K=0.06655$
- $\sigma_1 = 2.19 \times 10^{-3}$
- $\sigma_2 = 0.03285$

Una vez que queda definido el modelo y el controlador explicado en el capítulo 4, se procede a la obtención de resultados en bucle cerrado frente a variaciones en la referencia.

Este modelo de segundo orden tiene una especial dificultad de control, debido a que al retraso que existe por la comunicación entre el controlador y el modelo en Matlab&Simulink se le suma el hecho de que es un modelo especialmente lento con un tiempo de establecimiento superior a 2000s en bucle abierto.

A pesar de ello se ha conseguido un control estable, con un error en régimen permanente que tiende a 0 como se observa en la figura 6-7, aunque el control sigue siendo lento.

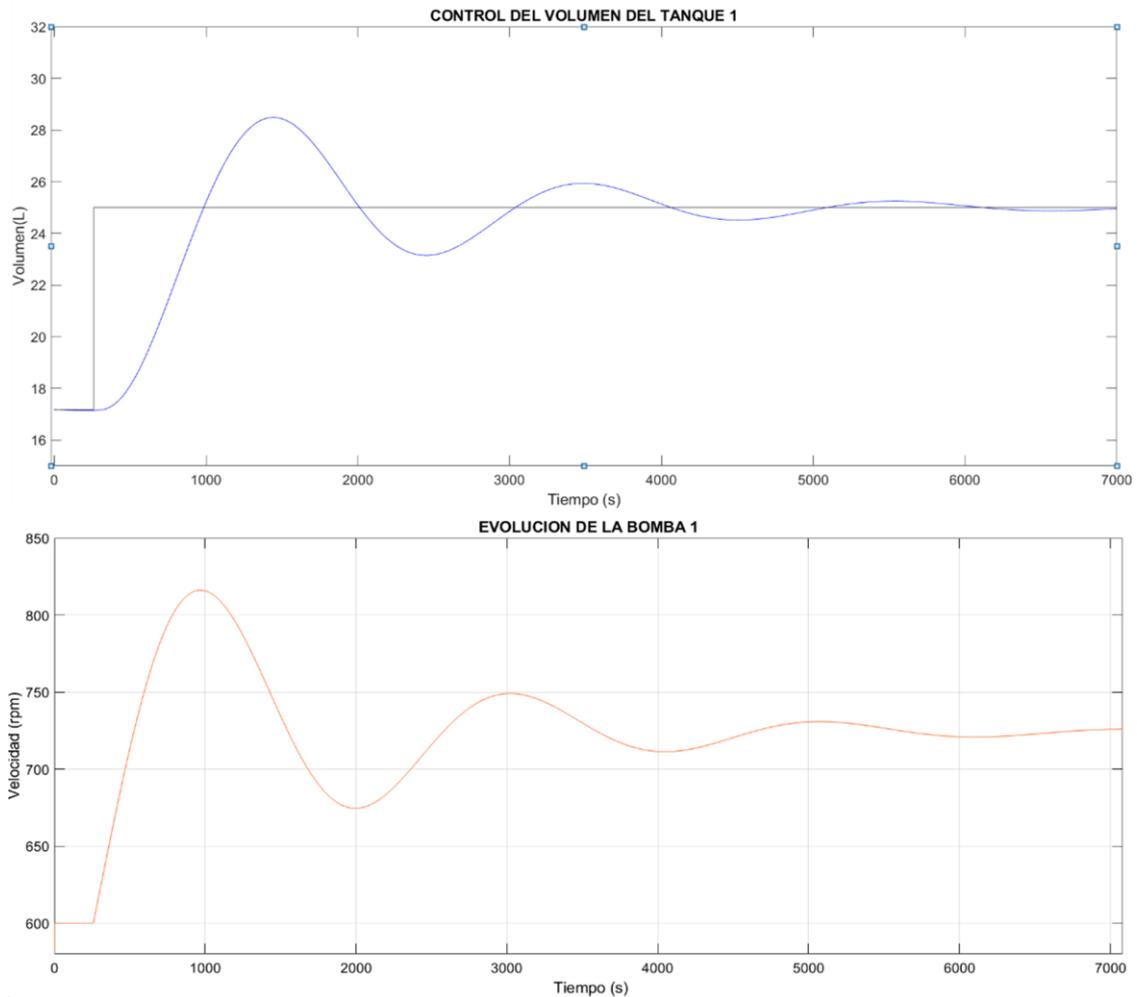


Figura 6-7. Respuesta del sistema en bucle cerrado en la configuración 2

En la respuesta mostrada se observa como la variable alcanza una sobreoscilación del 50% y un tiempo de establecimiento (98%) ligeramente inferior a 5000s.

6.1.3. Resultados de la tercera configuración

Esta configuración, como se ha explicado previamente, es algo distinta respecto a las anteriores, por lo que el modo de obtener resultados será ligeramente diferente. Se recuerda que en esta configuración las válvulas 1, 2, 3 y 4 estarán abiertas mientras que la 5 y 6 permanecerán cerradas, por lo que la bomba 1 alimenta los tanques 1 y 4, mientras que la bomba 2 hace lo propio con los tanques 2 y 3.

De nuevo se lleva el sistema a un setpoint en régimen permanente, pero en este caso será con ambas bombas a 600 rpm, por lo que nos encontraremos frente a un modelo simétrico tanque por tanque. Los volúmenes de los cuatro tanques en régimen permanente serán los siguientes:

- $V_{tanque1} = 43.44L$
- $V_{tanque2} = 43.44L$
- $V_{tanque3} = 0L$
- $V_{tanque4} = 0L$

A pesar de que existe un flujo de agua constante sobre los tanques 3 y 4, el valor en régimen permanente a 600 rpm será 0 debido a que el caudal de salida es mayor que el caudal de entrada de agua.

A continuación, se procede, a representar la respuesta de los tanques frente a entrada en escalón. En este caso se introduce un escalón en la bomba 1 que varía desde 600 hasta 700 rpm y posteriormente otro escalón hacia abajo en la bomba 2 que varía desde 600 hasta 400 rpm.

La respuesta de los tanques 1 y 2 se encuentra en la figura 6-8.

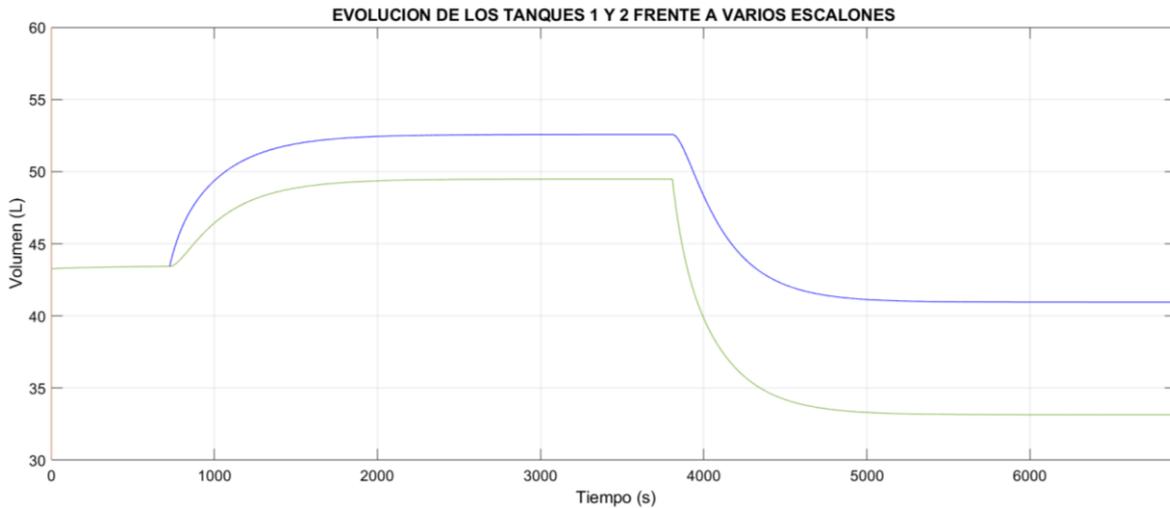


Figura 6-8. Respuesta antes escalón en la configuración 3

De la respuesta frente a escalón que se observa en la figura anterior, se obtienen los parámetros que definen el modelo de primer orden:

- $K=0.09$
- $\tau = 240$

Con estos modelos se crean dos controladores simétricos para la acción sobre las bombas 1 y 2 que controlen los tanques 1 y 2 adecuadamente.

Para la respuesta en bucle cerrado, se introducen varios cambios de referencia de subida y bajada frente a ambos volúmenes de los tanques, como se observa en la figura 6-9.

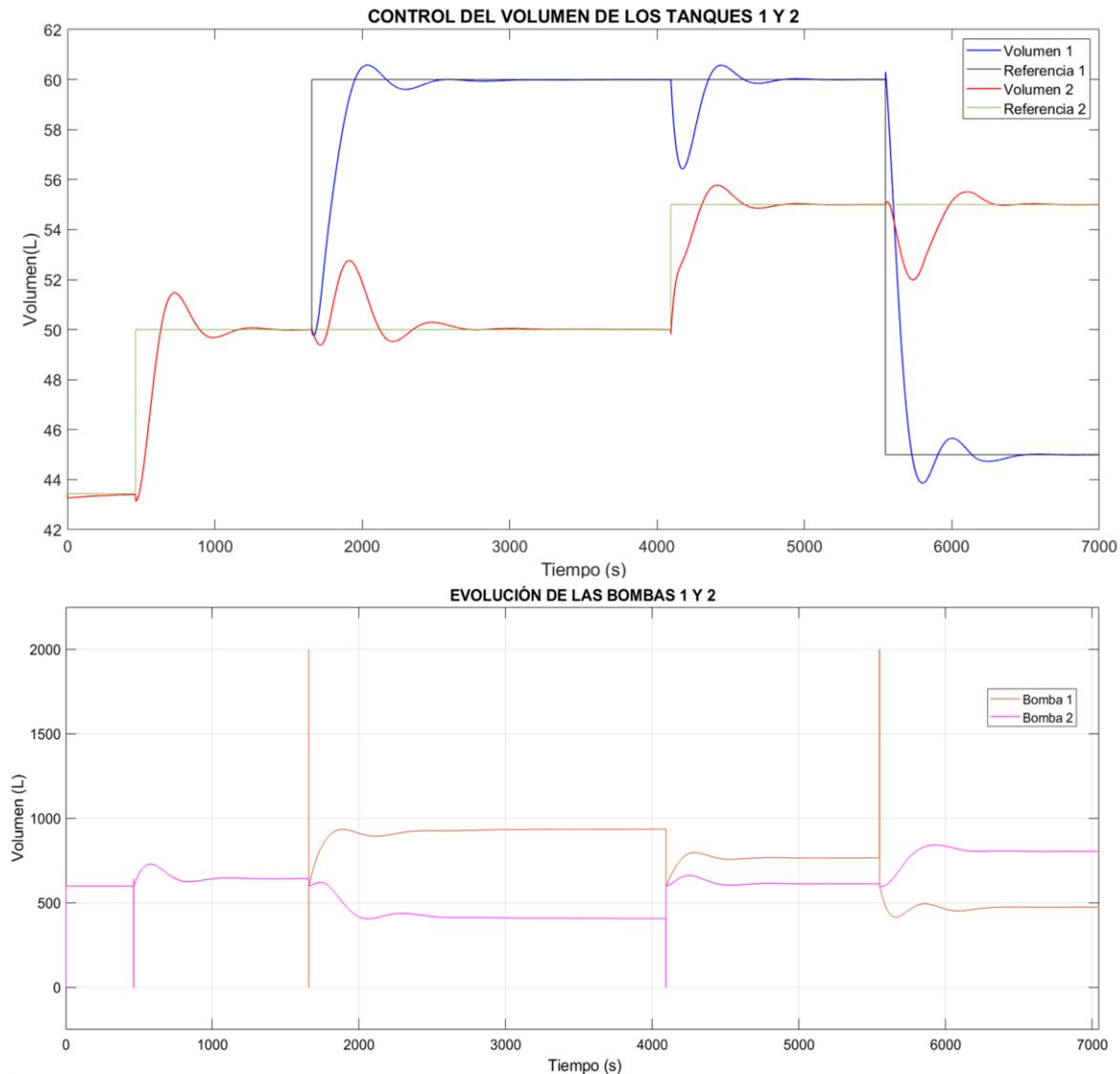


Figura 6-9. Respuesta del sistema en bucle cerrado en la configuración 3

En la respuesta se observa claramente como los valores de los volúmenes se adaptan a los cambios de referencia obteniendo un valor 0 en régimen permanente.

Por otra parte, el tiempo de establecimiento en bucle cerrado quedaría entorno a 600s, y la sobreoscilación menor a 25%.

6.2. Conclusiones y propuestas finales del trabajo

Se debe recordar que el objetivo principal de este trabajo era controlar el modelo creado en Matlab&Simulink a partir de un controlador en Codesys. Este objetivo ha sido alcanzado incluso en varias configuraciones del sistema como se ha mostrado en los resultados anteriores.

Cabe destacar la complejidad de la comunicación entre tres programas distintos mediante el único servidor OPC-UA. Se ha tenido que aprender a configurar la comunicación en cada programa por separado, ya sea como cliente o como servidor. En particular, Matlab&Simulink no incluye un bloque para este tipo de servidor como en el caso de OPC-DA, por lo que se ha tenido que crear la conexión manualmente a través de una función como se explicará más adelante en el anexo.

Con respecto al modelo, se ha intentado reproducir el sistema real del laboratorio, incorporando una cantidad considerable de elementos de la librería Simscape, que, unidos a la comunicación, establece un modelo complejo de ejecutar. Para poder simular en tiempo real, se ha incorporado un Solver sencillo Ode1 con paso fijo 0.3 segundos.

A continuación, detallando sobre el control realizado en Codesys, se ha conseguido un resultado muy satisfactorio en las configuraciones 1 y 3, obteniendo una respuesta en bucle cerrado con baja sobreoscilación además de bastante rápida, ya que se ha reducido el tiempo de subida y de establecimiento respecto al bucle abierto.

En la configuración 2, debido a la complejidad explicada anteriormente, ha sido necesario un examen analítico más exhausto para controlar el sistema de segundo orden. A pesar de que la respuesta ha sido sensiblemente lenta, se ha conseguido un control estable en bucle cerrado con una sobreoscilación aceptable, por lo que de nuevo el resultado vuelve a ser satisfactorio.

Por último, cabe destacar la incorporación del SCADA que facilita una interfaz entre el usuario y el controlador, que permite manejar con gran facilidad las variables principales del sistema, además de representar en tiempo real la respuesta del sistema en el modo y configuración que se estime oportuno.

Como proyecto futuro, se propone la posibilidad de incorporar el control y el SCADA creado en el modelo real del laboratorio, incorporando también la comunicación con el modelo real como cliente, para observar las diferencias con respecto a las simulaciones que se han llevado a cabo en este trabajo.

Por otra parte, se plantea la posibilidad de adaptar el sistema a un servidor que permita la comunicación a distancia, de manera que el control sea remoto, por lo que se realizaría desde una dirección IP distinta a la del modelo de Simscape creado en este proyecto. Esto facilitaría el desarrollo de las clases y prácticas online, sin necesidad de asistir presencialmente a los laboratorios de la Escuela de Ingenieros.

BIBLIOGRAFÍA

- [1] Tamás Kardos & Dénes Nimród Kutasi, «Hybrid modelling and model based control of the four tanke process», 2017.
- [2] Schneider Electric, «StruxureWare SCADA Expert Vijeo Citect v7.40 SP1» 2014.
- [3] Ignacio Alvarado aldea, «Proceso de los cuatro tanques»
- [4] Departamento Ingeniería Sistemas y automática «Control descentralizado de sistemas multivariables»

GLOSARIO

SCADA	Supervised Control and Data acquisition
OPC	Open Protocol Communication
POU	Program organization Unit
PLC	Programmable logic Controller (autómata programable)
UA	Unified Architecture
DA	Defined Architecture

ANEXO-A

Se ha añadido este anexo como guía para explicar detalladamente cómo configurar la comunicación mediante el servidor OPC-UA en los tres programas utilizados durante el trabajo: Codesys, Matlab&Simulink y Citect SCADA.

A.1. Configuración del servidor OPC-UA en Codesys

En primer lugar, recordemos que será el autómatas programable virtual de Codesys quien sea el servidor de la comunicación, mientras que los otros dos clientes permanecerán como clientes.

Para comenzar la configuración se crea un proyecto Standard en el Software de Codesys v3.5 (a partir de esta versión no hay problemas con el servidor OPC-UA). A continuación, se crea una lista de variables globales que serán las que se comuniquen a través del servidor OPC-UA siguiendo la siguiente secuencia:

-Application (botón derecho) /Agregar objeto/ Lista de variables globales.

Se adjunta un ejemplo por claridad en la figura A-1:

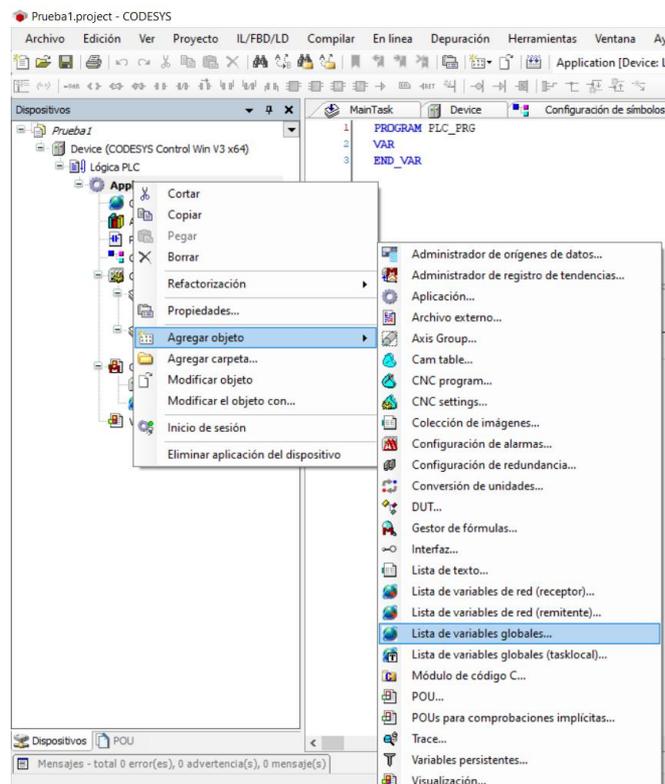


Figura A-1. Lista de variables globales

-Una vez creada la lista de variables que se comunicarán, se está en condiciones de incorporar la lógica del programa usando las variables globales que se comunicarán. En este caso, como se ha explicado previamente se crean 3 POU's auxiliares y una principal que llamará a cada una de las auxiliares cuando sea necesario.

Cuando la lógica funcione correctamente, se debe añadir una lista de configuración de símbolos siguiendo la misma ruta que se ha utilizado anteriormente:

-Application (Botón derecho) / Agregar objeto /Configuración de símbolos

Dentro de la lista de configuración de símbolos, añadimos las variables globales tal y como aparece en la figura A-2 y compilamos mediante el botón crear:

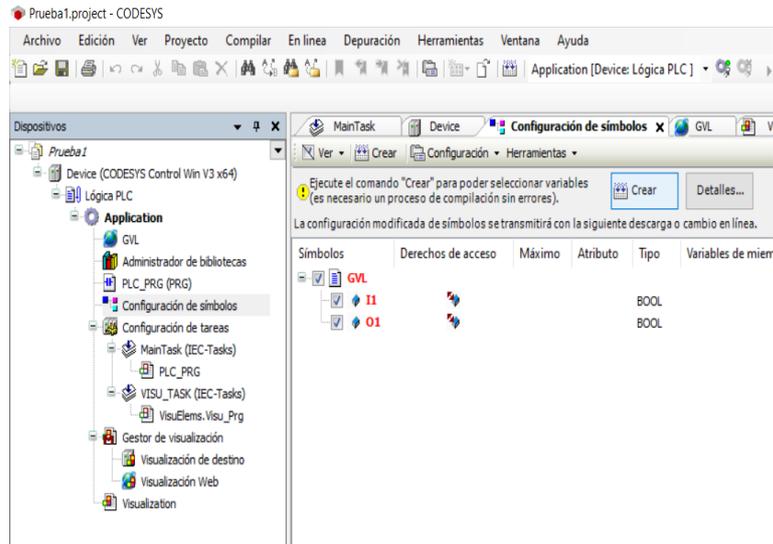


Figura A-2. Lista de configuración de símbolos

En este caso, se observa en la imagen que los derechos de acceso permiten a los clientes leer y escribir sobre las variables utilizadas sin ningún tipo de restricción.

Antes de continuar, es importante asegurarse de que el proyecto no se encuentra en modo simulación, es decir, la casilla debe estar desmarcada tal y como aparece en la figura 6-3:

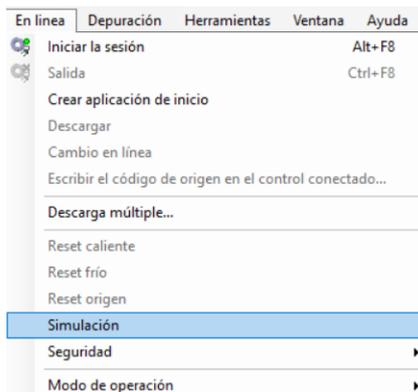


Figura A-3. Modo simulación del proyecto

A continuación, se ejecutan dos herramientas que deben aparecer en los iconos ocultos de la figura A-4, la puerta Gateway y el PLC:

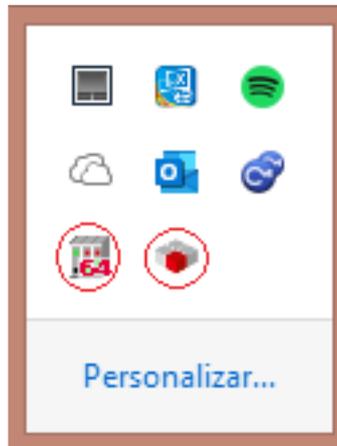


Figura A-4. Iconos ocultos en la pantalla

Estos dos iconos deben estar encendidos cada vez que se ejecute el proyecto, siendo un motivo constante de error al ejecutar ya que se desactivan cada 120 minutos.

Si no aparecen en el panel de iconos ocultos del ordenador se pueden buscar en el siguiente directorio:

C:\Program Files\CODESYS\3.5.16.40\GatewayPLC o alternativamente en el explorador del ordenador.

En este caso, con un sistema operativo de 64 bits, el nombre de estas herramientas es:

-Codesys Gateway SysTray -x64

-Codesys Control Win SysTray -x64

Una vez que ya aparezcan en los iconos ocultos asegurarse de activarlos (Start Gateway y Start PLC respectivamente).

A.1.1. Ejecución del proyecto con el servidor de comunicación

A partir de este momento, si se han realizado todos los pasos anteriores ya estamos en disposición de ejecutar el proyecto siguiendo la ruta:

En línea\Iniciar la sesión\ Botón Inicio

Se adjunta en la figura A-5 un ejemplo de cómo iniciar la ejecución:

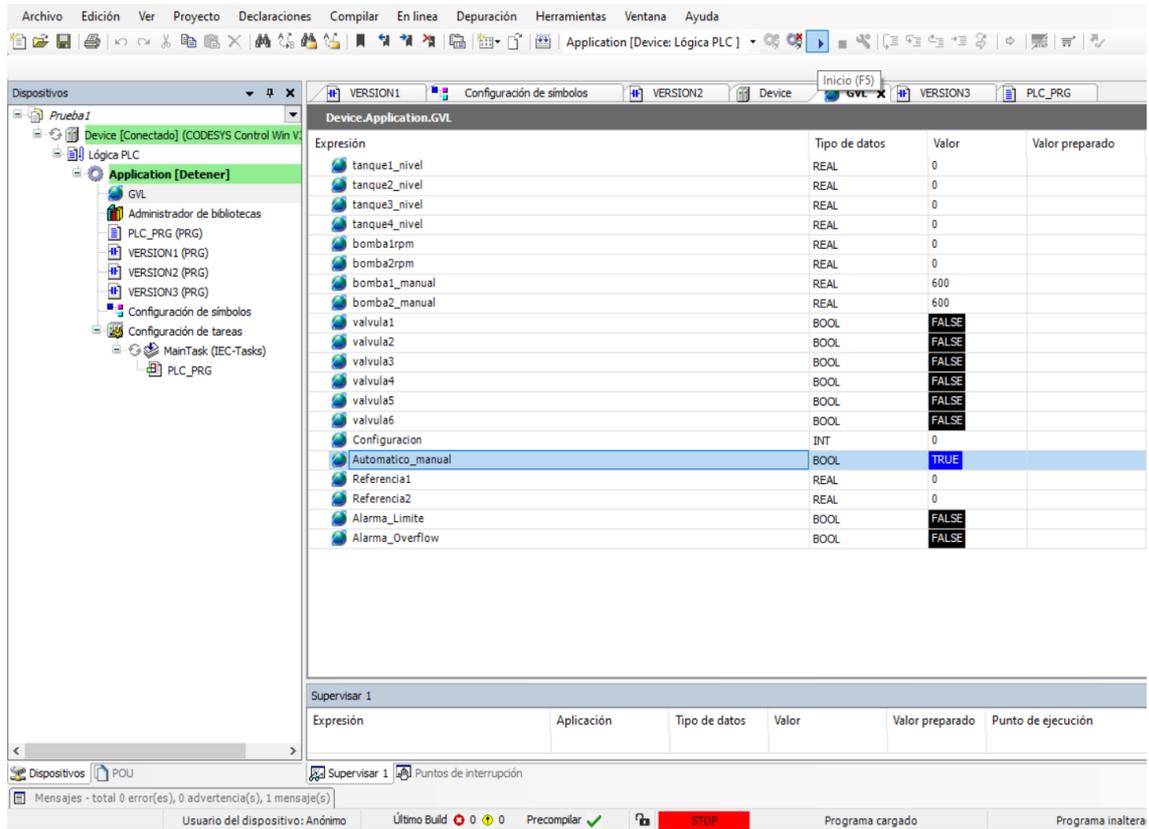


Figura A-5. Inicio de la ejecución del proyecto

En el caso de que ocurra algún error en la configuración, se procede a asegurarse de que el servidor está configurado correctamente, dirigiéndose hasta la ruta:

Device\ Configuración de comunicación:

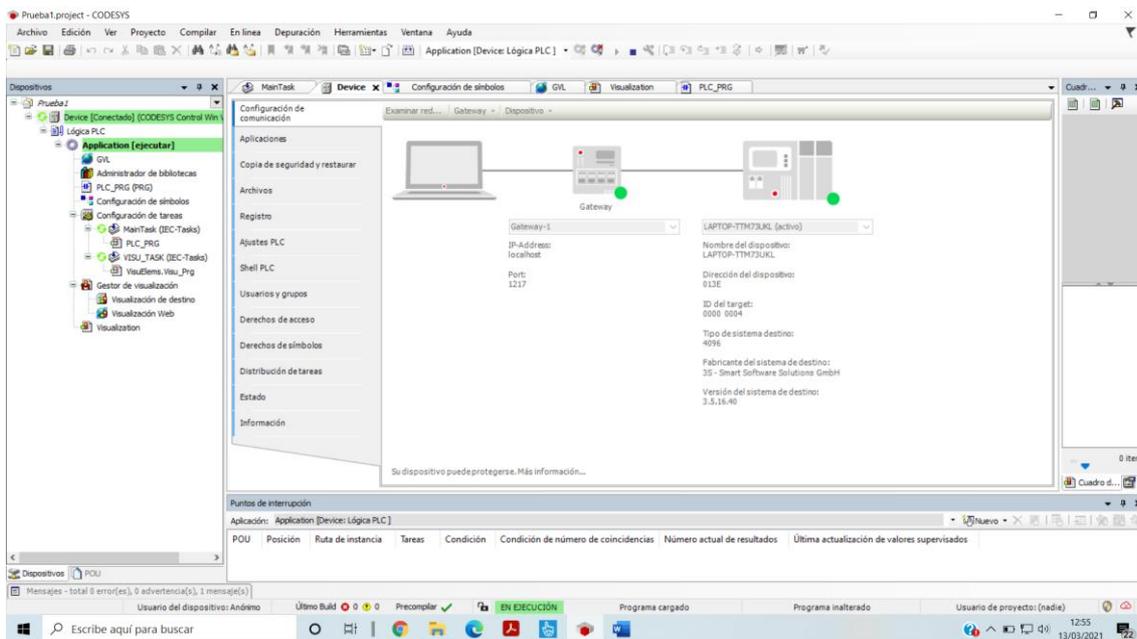


Figura A-6. Configuración de la comunicación

Comprobar que las dos luces de la figura A-6 están en verde y el proyecto está en ejecución tal y como aparece en la imagen. En el caso de que no sea así y no se haya conectado automáticamente el dispositivo al servidor se podrá hacer de modo manual.

Se pulsa el botón de examinar red que aparece en la figura A-7 y se selecciona nuestro positivo como conexión a la puerta “Gateway” correspondiente. A partir de este momento, ya deberían aparecer las dos luces verdes mencionadas anteriormente:

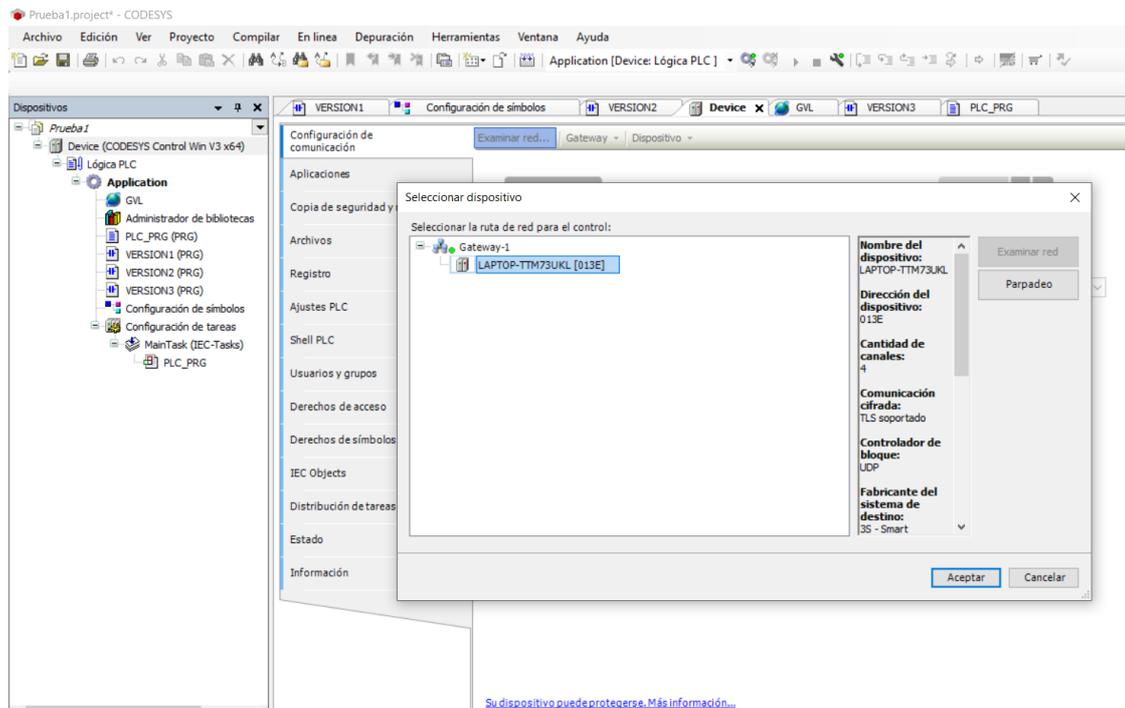


Figura A-7. Conexión manual del dispositivo a la puerta de enlace

Una vez que la ejecución sea correcta, este servidor se encuentra en condiciones de conectar todos los clientes que lo soliciten, por lo tanto, siempre será el primer programa en ejecutarse.

A-2. Configuración de un cliente en el Software Citect SCADA 2018

Una vez que se ha creado el servidor, se procede a conectar el primero de los clientes mediante configuración OPC.

En primer lugar, se crea un nuevo proyecto en la pantalla principal del programa tal y como aparece en la figura A-8 y elegir un nombre:

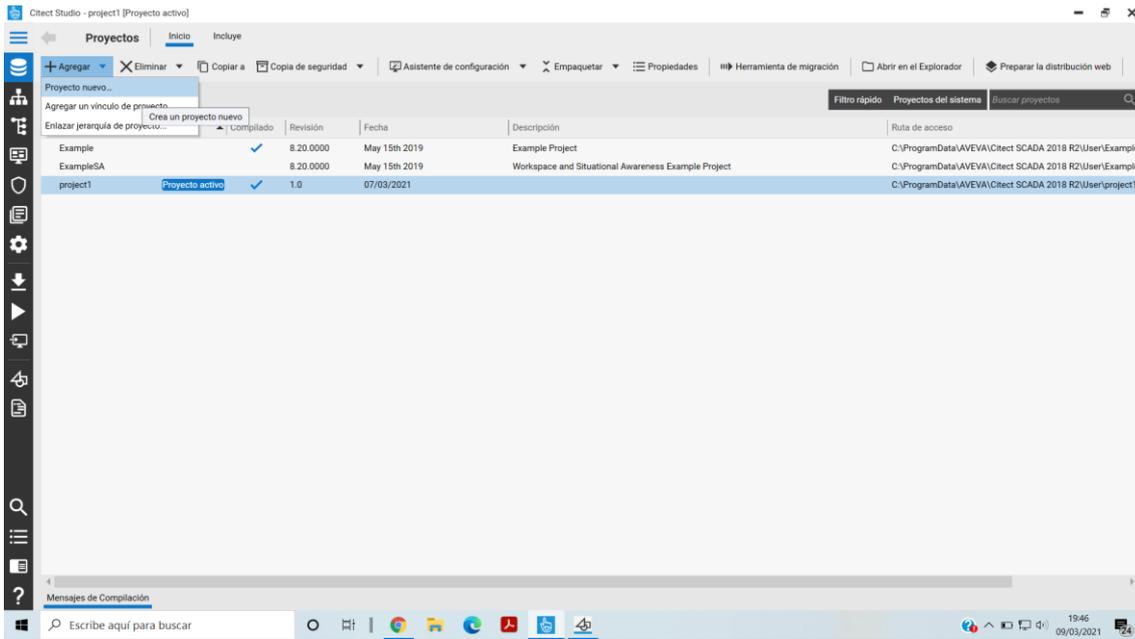


Figura A-8. Creación de un nuevo proyecto Citect SCADA

El proyecto se debe guardar en el directorio que viene por defecto para evitar posteriores problemas de ejecución. A continuación, se pulsa el botón ‘Activar’ sobre el proyecto que se acaba de crear de manera que se convierta en el único proyecto activo.

Una vez que se encuentre activo tal y como aparece en la figura A-8, se añade un nuevo dispositivo en el que se configura la comunicación OPC-UA a través de la siguiente ruta:

-Tipología (herramientas en el panel izquierdo) / Dispositivos de E/S / Dispositivo nuevo

En este momento, se abre un asistente de comunicaciones tal y como aparece en la figura A-9. El dispositivo se debe asociar al proyecto activo que acabamos de crear.

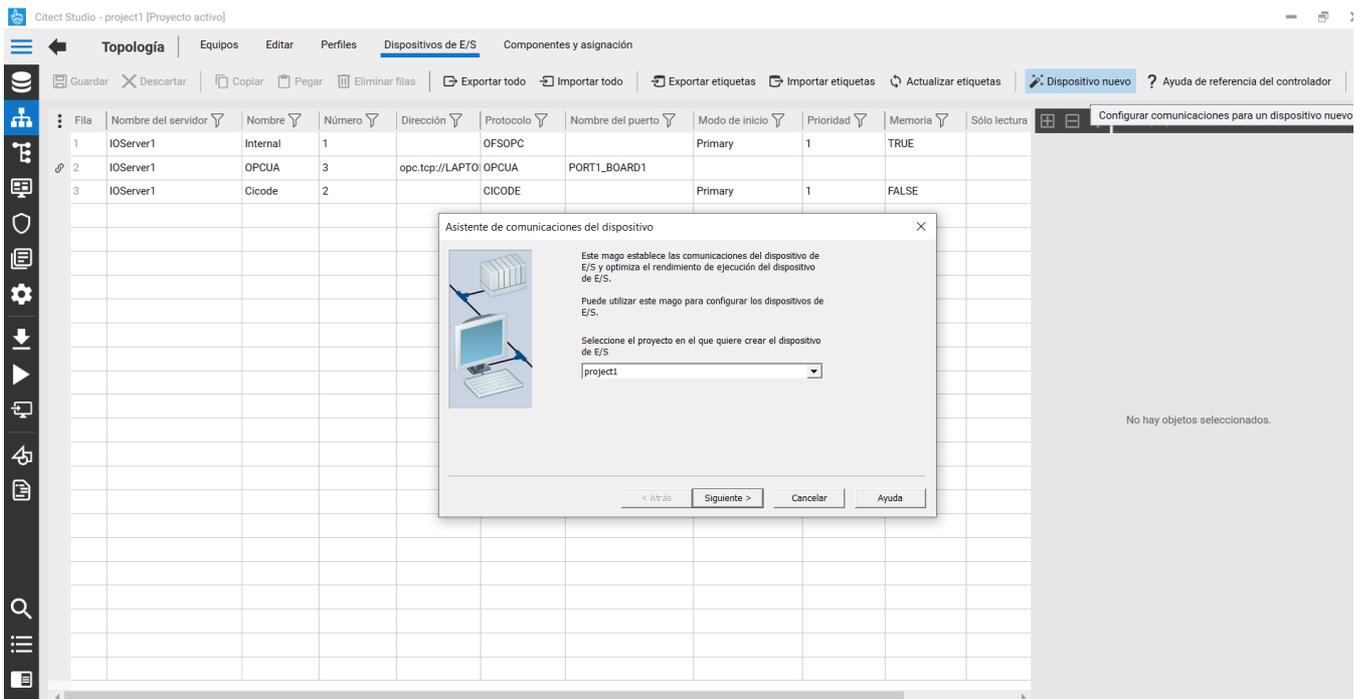


Figura A-9. Asistente de configuración de nuevo dispositivo

En este asistente de configuración, se continúa del siguiente modo:

-Siguiente/ Siguiente... hasta la pantalla de crear un dispositivo E/S que aparece en la figura A-10, donde se nombra el dispositivo en función de la utilidad que se le vaya a dar. En este caso nuestro dispositivo se nombra acorde a la comunicación OPC-UA que se va a realizar

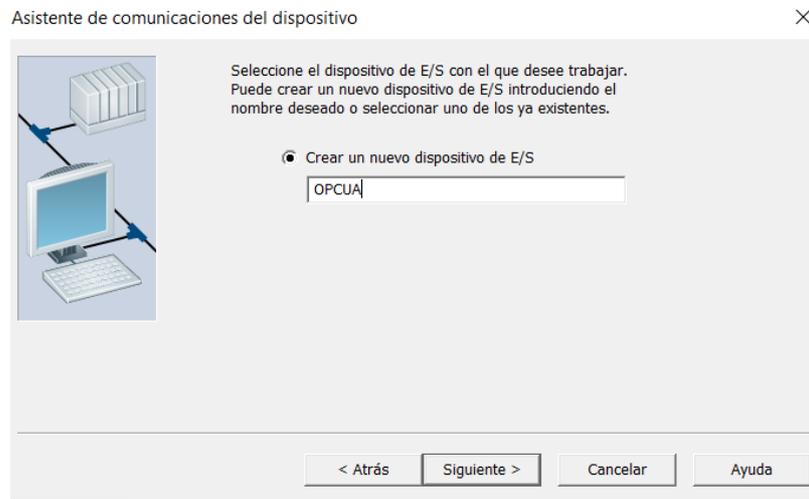


Figura A-10. Pantalla de crear un nuevo dispositivo

Se prosigue aceptando los siguientes paneles hasta elegir el método de comunicación del dispositivo OPCTCP que aparece en la figura A-11:

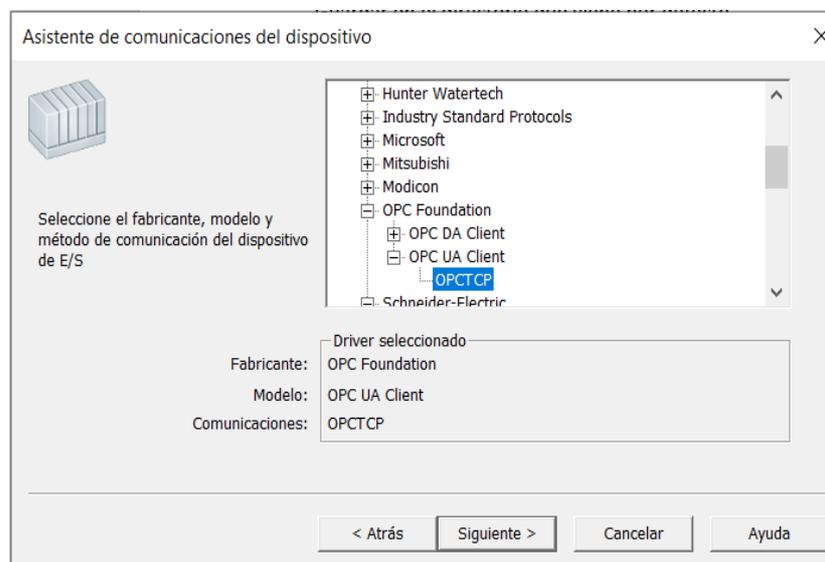


Figura A-11: Método de comunicación del dispositivo

En la siguiente pantalla se debe introducir la dirección del servidor OPC. La dirección está basada en la siguiente estructura:

opc.tcp://” Nombre del servidor OPC”: “Numero del puerto de comunicación”

En nuestro caso la dirección será la que aparece en la figura B-4.

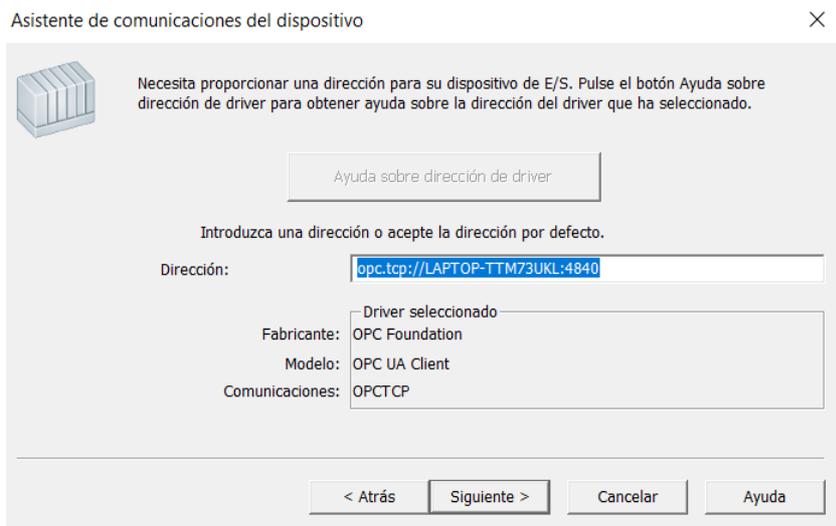


Figura A-12: Dirección del servidor OPC

En el siguiente panel se vincula el cliente con el servidor creado en Codesys siguiendo las instrucciones que aparecen en la figura A-13:

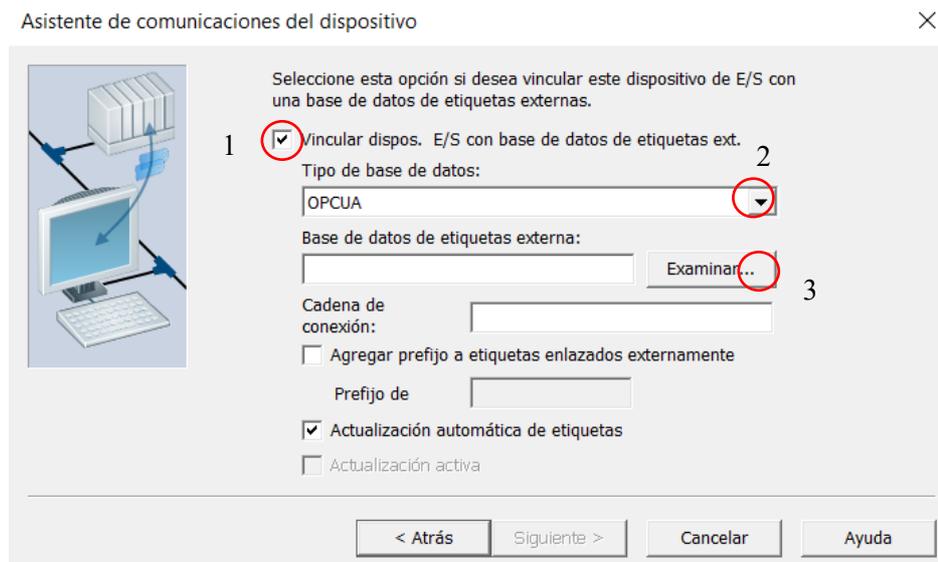


Figura A-13. Vinculación del dispositivo E/S

A continuación, se intenta seleccionar la rama adecuada para cargar solo las variables que interesan. En nuestro caso se eligen las variables globales tal y como aparece en la figura A-14:

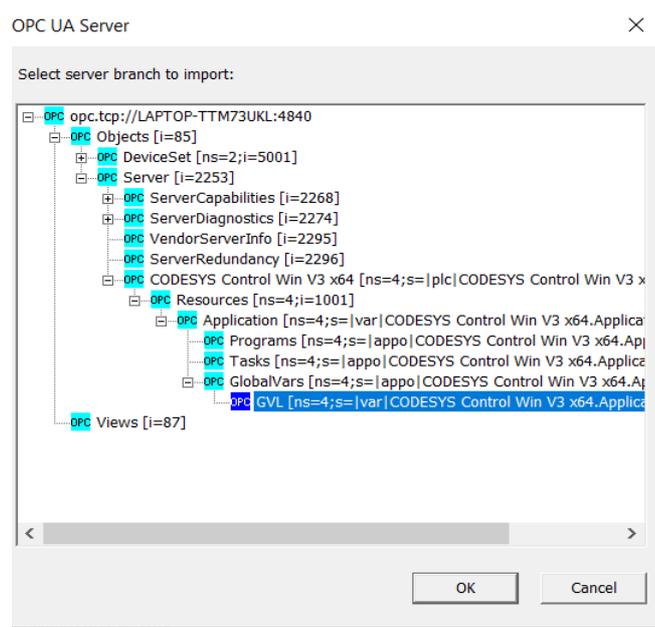


Figura A-14: Selección de las variables comunicadas

Finalmente se pulsa finalizar: Se incorporarán el número de variables que se hayan configurado en el servidor. Para estar seguros de que se han cargado correctamente las variables deseadas, ir a la siguiente ruta:

Modelo del sistema (panel izquierdo) /variables.

En el caso del proyecto, se muestran las variables que aparecen en la figura A-15:

Fila	Equipo	Nombre del ítem	Nombre de etiqueta	Nombre del cluster	Dispositivo de E/S	Tipo de datos	Dirección	Comentario	Escala cc
1			GVLAlarma_Limite		OPCUA	DIGITAL	ns=4;s= var CODI		
2			GVLAlarma_Overflow		OPCUA	DIGITAL	ns=4;s= var CODI		
3			GVLAutomatico_manual		OPCUA	DIGITAL	ns=4;s= var CODI		
4			GVLbomba1_manual		OPCUA	REAL	ns=4;s= var CODI		
5			GVLbomba1rpm		OPCUA	REAL	ns=4;s= var CODI		
6			GVLbomba2_manual		OPCUA	REAL	ns=4;s= var CODI		
7			GVLbomba2rpm		OPCUA	REAL	ns=4;s= var CODI		
8			GVLConfiguracion		OPCUA	INT	ns=4;s= var CODI		
9			GVLReferencia1		OPCUA	REAL	ns=4;s= var CODI		
10			GVLReferencia2		OPCUA	REAL	ns=4;s= var CODI		
11			GVLtanque1_nivel		OPCUA	REAL	ns=4;s= var CODI		
12			GVLtanque2_nivel		OPCUA	REAL	ns=4;s= var CODI		
13			GVLtanque3_nivel		OPCUA	REAL	ns=4;s= var CODI		
14			GVLtanque4_nivel		OPCUA	REAL	ns=4;s= var CODI		
15			GVLvalvula1		OPCUA	DIGITAL	ns=4;s= var CODI		
16			GVLvalvula2		OPCUA	DIGITAL	ns=4;s= var CODI		
17			GVLvalvula3		OPCUA	DIGITAL	ns=4;s= var CODI		
18			GVLvalvula4		OPCUA	DIGITAL	ns=4;s= var CODI		
19			GVLvalvula5		OPCUA	DIGITAL	ns=4;s= var CODI		
20			GVLvalvula6		OPCUA	DIGITAL	ns=4;s= var CODI		

Figura A-15. Variables cargadas en el servidor OPC-UA

En este momento ya se ha incorporado al software un nuevo dispositivo virtual que se encarga de la

comunicación OPC-UA. El siguiente paso será buscar (en el explorador del sistema) el Configurador del programa y ejecutarlo (figura A-16).

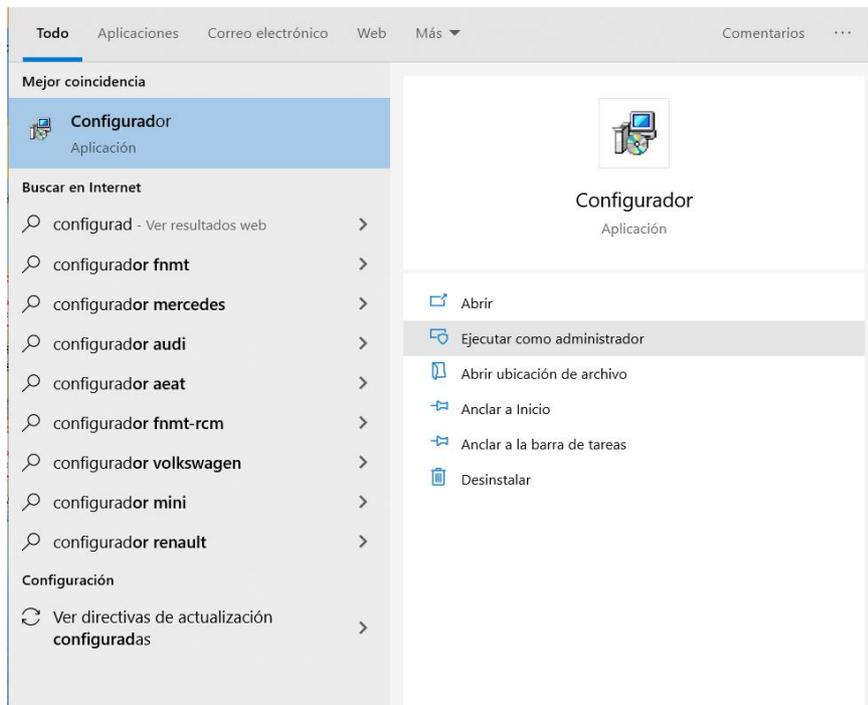


Figura A-16. Búsqueda del configurador Citect SCADA

Posteriormente, seleccionar OPC UA Client driver y crear una nueva configuración (si no estaba previamente creada) como aparece en la figura A-17

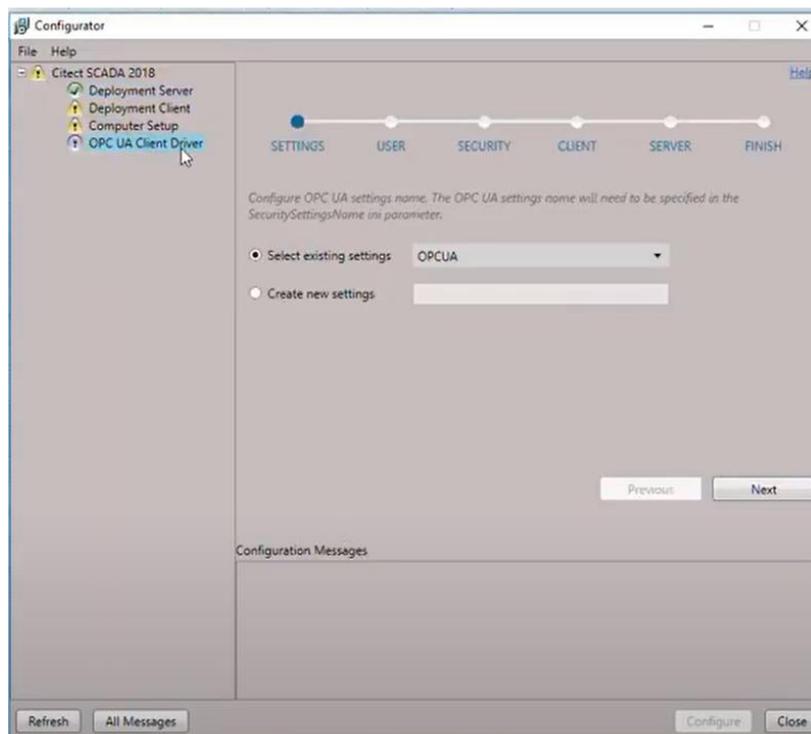


Figura A-17. Configurar un nuevo cliente OPC-UA

Seguidamente se puede crear un modo de encriptado (no se recomienda para este proyecto). En el último paso del panel, se pulsa el botón ‘Configurar’ (parte inferior del panel) y después probar la conexión introduciendo de nuevo la dirección del servidor OPC. Una vez realizado quedaría configurado el cliente.

Es importante no adentrarse en configurar otros parámetros en este panel de los que han sido explicados aquí, ya que el hecho de estar en modo demostración puede generar algún problema en la compilación y posterior ejecución.

Volviendo al programa Citect SCADA 2018, una vez que se haya creado la pantalla de explotación (se realiza en el editor de gráficos), antes de compilar y ejecutar el proyecto se debe añadir el asistente de configuración (o Setup Wizard) tal y como aparece en la figura A-18:

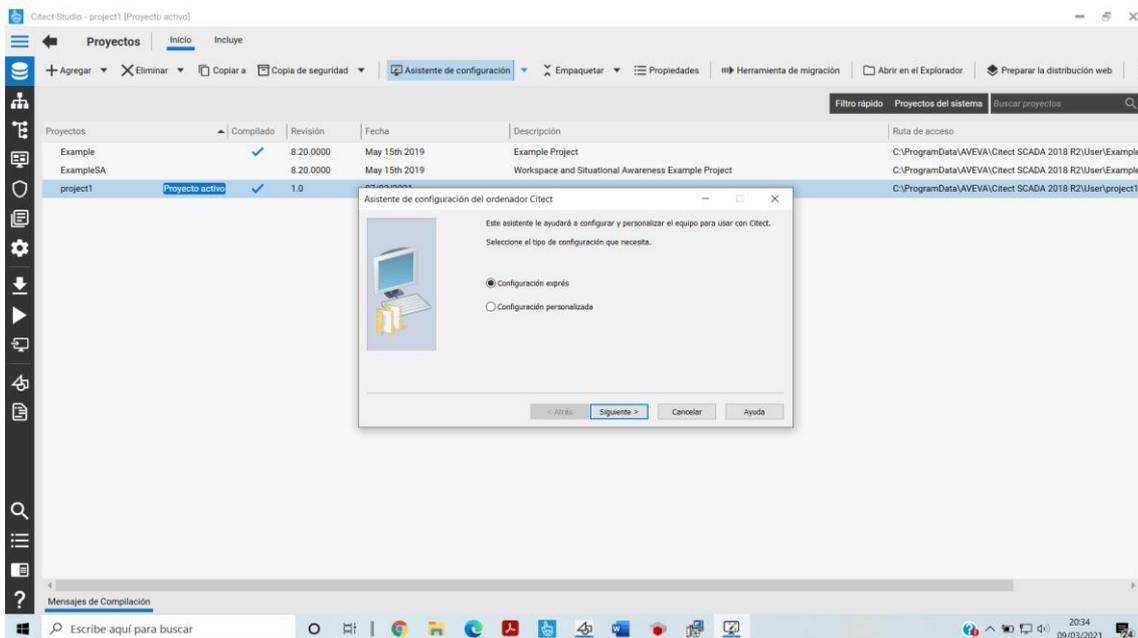
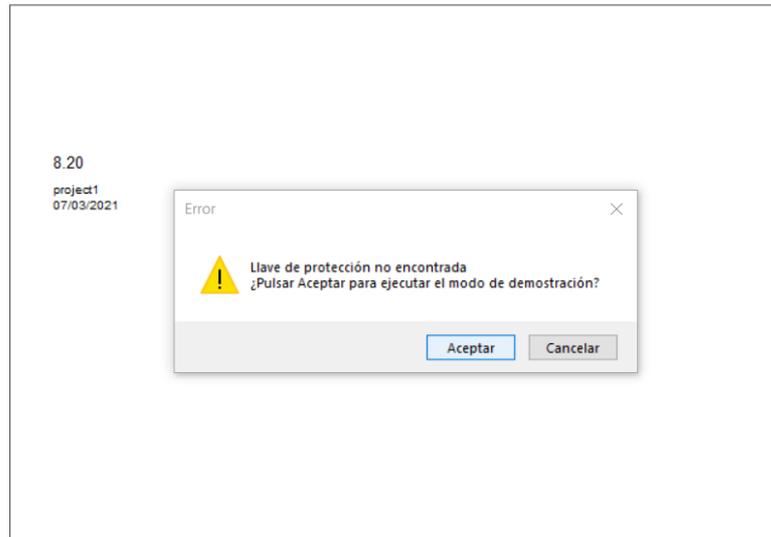


Figura A-18. Asistente de configuración

Se recomienda configurar todo como viene por defecto. No obstante, se pueden ajustar ciertos parámetros sobre la pantalla de explotación inicial o lo que se estime conveniente.

Una vez finalizado se podrá compilar y ejecutar el programa. Una vez que no haya errores y se haya podido ejecutar correctamente, debe aparecer un mensaje de aviso sobre la falta de licencia y la posibilidad de ejecutar en modo demostración cómo aparece en la figura A-19.



Use el árbol de menú para navegar a otras páginas.

Figura A-19. Ejecución del modo demostración

Cuando se hayan creado las páginas pertinentes en el editor de gráficos explicadas anteriormente en el trabajo, se podrán leer y escribir en las variables del servidor sin ningún tipo de error ni aviso.

A.3. Conexión del cliente en Matlab&Simulink

En este último apartado, se explica la conexión del segundo cliente al servidor, la cual ha sido más laboriosa que las anteriores debido a que Matlab&Simulink, como hemos mencionado anteriormente, no incluye un bloque de configuración como si lo hace para el caso del Servidor OPC-DA.

La comunicación se basa en crear una función en Matlab que incluya la conexión al servidor, elección de variables que se desea estudiar y configuración de la lectura y escritura constante en tiempo real de esas variables elegidas.

En la figura A-20 se adjunta la función donde se han incluido todos los comandos necesarios para la comunicación:

```

1  function [y]=Comunicacion OPCUA(tanque1,tanque2,tanque3,tanque4)
2  -   clic
3  -   persistent init_server
4  -   persistent init_nodes
5  -   persistent uaClient
6  -   persistent input1
7  -   persistent input2
8  -   persistent input3
9  -   persistent input4
10 -   persistent output1
11 -   persistent output2
12 -   persistent output3
13 -   persistent output4
14 -   persistent output5
15 -   persistent output6
16 -   persistent output7
17 -   persistent output8
18   %Se comprueba la primera vez que no está inicializado
19   if (isempty(init_server))
20       valor=zeros(1,8);
21       init_server=0;
22       init_nodes=0;
23   end
24   %Identificación del servidor y conexión del cliente (Simulink)
25   if init_server==0
26       uaClient=opcua('LAPTOP-TTM73UKL',4840);
27       connect(uaClient)
28       init_server=1;
29   end
30   %Define las variables a estudiar en nodos de cliente a través del servidor
31   if uaClient.isConnected==1 && init_nodes==0
32       init_nodes=1;
33       input1=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.tanque1_nivel',uaClient);
34       input2=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.tanque2_nivel',uaClient);
35       input3=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.tanque3_nivel',uaClient);
36       input4=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.tanque4_nivel',uaClient);
37       output1=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.bomba1rpm',uaClient);
38       output2=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.bomba2rpm',uaClient);
39       output3=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.valvula1',uaClient);
40       output4=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.valvula2',uaClient);
41       output5=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.valvula3',uaClient);
42       output6=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.valvula4',uaClient);
43       output7=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.valvula5',uaClient);
44       output8=opcuanode(4,'|var|CODESYS Control Win V3 x64.Application.GVL.valvula6',uaClient);
45
46   end
47
48   %Una vez que ya he creado los nodos leo y escribo el valor de la variable
49   if uaClient.isConnected==1 && init_nodes==1
50       [valor(1),~,~]=readValue(uaClient, output1);
51       [valor(2),~,~]=readValue(uaClient, output2);
52       [valor(3),~,~]=readValue(uaClient, output3);
53       [valor(4),~,~]=readValue(uaClient, output4);
54       [valor(5),~,~]=readValue(uaClient, output5);
55       [valor(6),~,~]=readValue(uaClient, output6);
56       [valor(7),~,~]=readValue(uaClient, output7);
57       [valor(8),~,~]=readValue(uaClient, output8);
58
59       writeValue(uaClient,input1,tanque1)
60       writeValue(uaClient,input2,tanque2)
61       writeValue(uaClient,input3,tanque3)
62       writeValue(uaClient,input4,tanque4)
63   end
64
65   %Asigno finalmente el valor a la salida
66   [y]=double(valor);
67   end

```

Figura A-20. Función de comunicación OPC-UA en Matlab

A partir de este momento, pasamos a explicar cómo se ha diseñado la función detalladamente:

- Los cuatro valores a la entrada (tanque1, tanque2, tanque3, tanque4) son los volúmenes de los tanques en litros en el momento en que se llama a la función. Estos valores se extraerán directamente del modelo de Simulink mostrado en capítulos anteriores.
- El segundo paso será inicializar todas las variables necesarias en esta función. Se definen todas como variables persistentes, lo que significa que mantienen su valor de una llamada de la función a la siguiente. Esto es recomendable para trabajar en tiempo real, especialmente en el caso de las variables con las que estamos trabajando, ya que en el caso de que no se actualice la variable en alguna de las llamadas mantiene el valor anterior en lugar de ir a cero.

Se usará una variable que define cuando el servidor se encuentra inicializado (init_server), otra análoga para los nodos, que irán asociados a las variables (init_nodos) y otra que define al cliente que se desea conectar al servidor. Finalmente se añaden doce nodos que se asociarán a las variables de entrada (input1...input4) y salida (output1...output8).

- A continuación, se inicializa un servidor (la primera vez que se llama a la función) que será el equivalente al servidor ya existente creado en Codesys.
- El siguiente paso sería crear un cliente y conectarlo al servidor existente. Al igual que en el caso anterior, esto solo se realiza en la primera llamada a la función, por lo que a partir de la segunda llamada se omiten estos dos pasos.
- Una vez que la conexión del cliente es correcta, se deben asociar una por una todas las variables del servidor a los nodos dentro del cliente. Por ejemplo: la variable del volumen del tanque 1 “tanque1_nivel”, se busca en el servidor OPC-UA dentro de la ruta que aparece en la función y se asocia al nodo “input1” dentro del cliente “uaClient”.

Si no se encuentra la ruta de la variable que se desea, un modo alternativo sería buscar variable por variable gracias al comando de Matlab “browseNamespace” y añadirlas al nodo correspondiente.

- Una vez que se han definido los cuatro nodos con variables de entrada y los ocho con variables de salida, se procede a leer y escribir los valores. En la función, se ha optado por leer el valor de los nodos de salida del cliente y copiarlos en cada una de las componentes del vector “valor” definido previamente.

A continuación, se escriben los valores de las entradas definidas al inicio de la función (tanque1...tanque4) en los valores de los nodos asociados a variables de entrada (input1...input4). Estos valores se verán actualizados automáticamente en el servidor OPC.

- Finalmente, el vector de ocho componentes “valor” se pasa a tipo “double” para poder pasar todos estos valores al modelo Matlab&Simulink. Con este último paso se da por concluida la función.

A.3.1 Incorporación de la comunicación en el modelo

Como hemos dicho anteriormente, no existe un bloque concreto de comunicación OPC-UA, por lo que la única solución es incorporar un bloque que integre la función que se ha creado en el apartado anterior. Esto es exactamente lo que se ha realizado gracias a la ayuda del bloque “Interpreted MATLAB Fcn” tal y como aparece en la figura A-21.

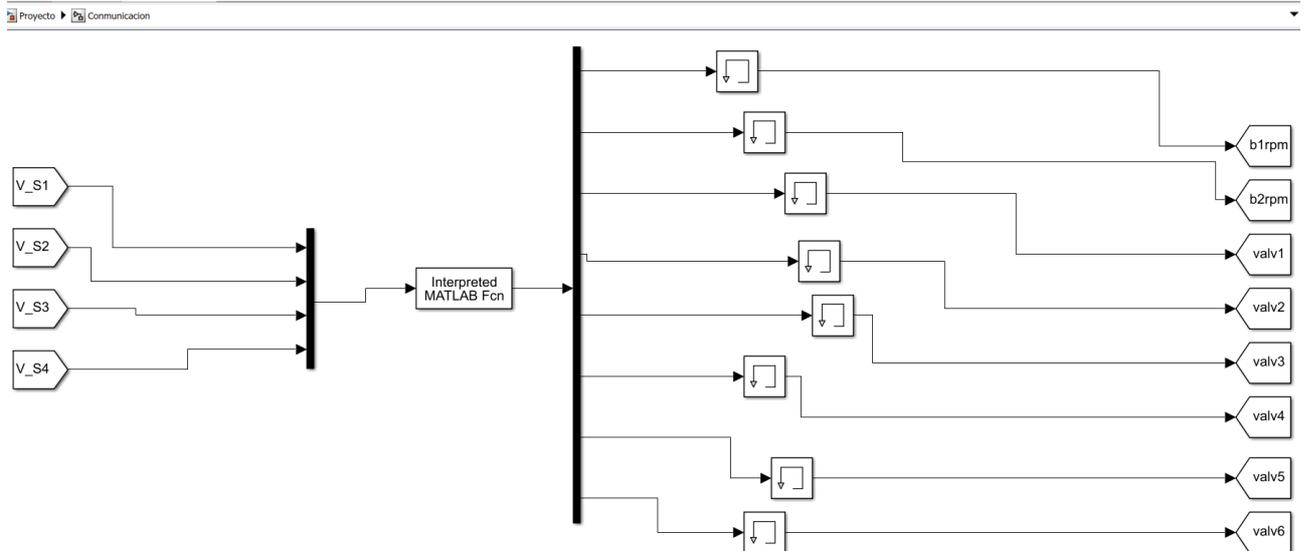


Figura A-21. Incorporación de la comunicación en el modelo Matlab&Simulink

Siendo coherente con la función, se han introducido cuatro variables de entrada a la función con la ayuda de un bloque “Mux” y ocho variables de salida con la ayuda de un bloque “Demux”.

Por otra parte, cabe destacar que la configuración del bloque “Interpreted MATLAB Fcn” requiere una nomenclatura especial en función del número de variables de entrada que aparece en la figura A-22.

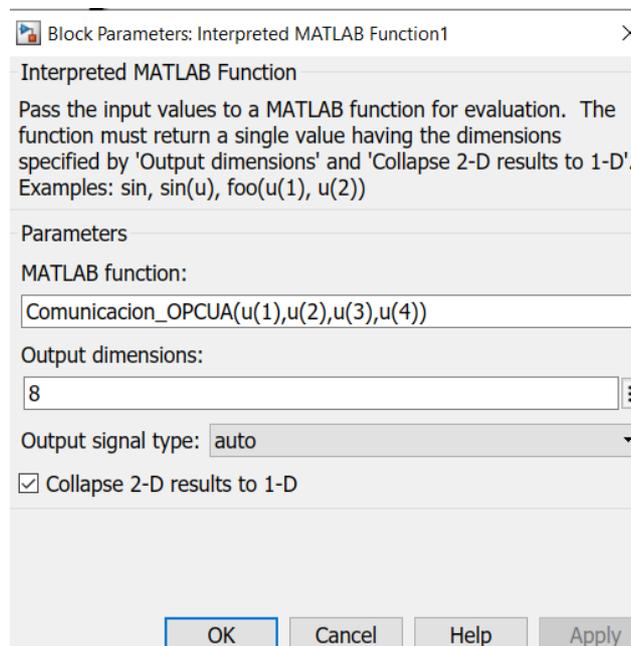


Figura A-22. Nomenclatura del bloque “Interpreted Matlab Fcn”

No sería válido introducir el nombre real de la variable, el nombre que se use en la función o cualquier otra nomenclatura.

Por último, se aconseja revisar dos motivos de error frecuentes en la ejecución del modelo debido a la comunicación:

- En primer lugar, la función de comunicación previamente explicada debe encontrarse en la misma carpeta que el modelo creado, y ambas deben estar en la carpeta que se encuentre abierta en el momento del inicio de la ejecución.
- Por otra parte, cada vez que se detenga la comunicación por parte del servidor en Codesys, se debe reiniciar el proceso también en Matlab para borrar el cliente, por lo que se debe añadir un “Clear all” en el Command Window.

Una vez que todo ha sido configurado correctamente, la comunicación entre los tres programas se podrá realizar en tiempo real sin ningún tipo de restricción, ya que todos podrán leer y escribir sobre el servidor en tiempo real.