



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería en Tecnologías Industriales**

**LABORATORIO VIRTUAL DE UNA PLANTA  
PILOTO DE DOS TANQUES COMUNICANTES**

**Autor:**

**Ríos Vicente, Juan Carlos**

**Tutor:**

**Zamarreño Cosme, Jesús María  
Ingeniería de Sistemas y  
Automática**

**Valladolid, Julio 2023**



# Agradecimientos

A la Universidad de Valladolid, que tantas oportunidades me ha dado.

A mi tutor, Jesús María Zamarreño, por acompañarme en el proceso y ser paciente conmigo.

A mi familia, que siempre ha estado a mi lado.

A mis amigos, que tantos buenos momentos me han dado y que hacen de la universidad una etapa difícil de olvidar.



# Resumen

Se ha desarrollado un laboratorio virtual ejecutable en un navegador web que simula el comportamiento de dos tanques comunicados mediante una tubería en su parte inferior. Su comportamiento se ha ajustado para que sea similar a las plantas reales del laboratorio de Control de Procesos. El primer depósito recibe un caudal de agua proveniente de una bomba y el segundo desaloja un cierto caudal mediante una válvula manual de salida. Tanto la potencia de la bomba como la apertura de la válvula manual de salida son regulables. Se puede realizar control manual, control PID y control Todo/Nada con histéresis. La evolución del sistema se aprecia visualmente en un sinóptico de la planta y se ve reflejada en gráficas temporales. Asimismo, los valores obtenidos se pueden exportar a un archivo de texto para realizar análisis posteriores en programas externos.

## Palabras clave

Laboratorio virtual, simulación de sistemas, tanques comunicantes, control de procesos, fundamentos de automática.

## Abstract

A virtual laboratory has been developed that can be run in a web browser to simulate the behaviour of two tanks connected by a pipe at their bottom. Their behaviour has been adjusted to be similar to the real plants in the Process Control laboratory. The first tank receives a flow of water from a pump and the second tank discharges a certain flow through a manual outlet valve. Both the power of the pump and the opening of the manual outlet valve are adjustable. Manual control, PID control and On/Off control with hysteresis are possible. The evolution of the system can be seen visually in a synoptic of the plant and is reflected in time graphs. The values obtained can also be exported to a text file for subsequent analysis in external programs.

## Keywords

Virtual laboratory, system simulation, communicating tanks, process control, fundamentals of automation.



# Índice de contenidos

<b>CAPÍTULO 1: Introducción y Objetivos .....</b>	<b>17</b>
1.1 Introducción.....	17
1.2 Objetivos .....	18
<b>CAPÍTULO 2: Easy Java/JavaScript Simulations .....</b>	<b>21</b>
2.1 Introducción.....	21
2.2 Instalación .....	23
2.3 Desarrollo de aplicaciones en EJS .....	24
2.3.1 Descripción .....	25
2.3.2 Modelo .....	26
2.3.3 HtmlView .....	27
2.4 Empaquetamiento de la simulación .....	31
<b>CAPÍTULO 3: Fundamento teórico .....</b>	<b>35</b>
3.1 Modelado de dos tanques comunicantes .....	35
3.2 Resolución mediante el método numérico Runge - Kutta .....	40
3.3 Estructura de control .....	44
3.3.1 Operación manual .....	45
3.3.2 Controlador PID .....	45
3.3.2.1 Acciones del controlador PID .....	47
3.3.2.2 Sintonización de un controlador PID .....	49
3.3.2.3 Sintonización manual de un controlador PID .....	49
3.3.2.4 Efectos de saturación y límites del actuador:.....	50
3.3.2.5 Controlador PID con filtro en el término derivativo .....	51
3.3.2.6 Controlador PID de JavaRegula. ....	53

3.3.3	Controlador Todo/Nada con histéresis .....	55
<b>CAPÍTULO 4:</b>	<b>Desarrollo del laboratorio virtual .....</b>	<b>59</b>
4.1	Interfaz gráfica .....	59
4.1.1	Barra de herramientas .....	59
4.1.2	Ilustración de la simulación .....	62
4.1.3	Gráficas .....	66
4.1.4	Configuración Simulación .....	68
4.1.5	Opciones Operación Manual.....	71
4.1.6	Opciones Controlador PID .....	72
4.1.7	Opciones Controlador Todo/Nada.....	74
4.2	Modelo.....	75
4.2.1	Variables.....	75
4.2.2	Inicialización.....	82
4.2.3	Evolución .....	83
4.2.4	Relaciones fijas.....	88
4.2.5	Propio .....	89
4.3	Créditos y Descripción.....	91
4.4	Empaquetamiento de la simulación y detalles finales .....	91
<b>CAPÍTULO 5:</b>	<b>Validación del laboratorio virtual .....</b>	<b>95</b>
5.1	Primer experimento .....	95
5.2	Segundo experimento .....	99
5.3	Tercer experimento.....	101
5.4	Conclusión.....	104
<b>CAPÍTULO 6:</b>	<b>Utilización del laboratorio virtual .....</b>	<b>105</b>
6.1	Apertura del laboratorio virtual.....	105



6.2 Barra de herramientas.....	106
6.3 Visualización gráfica .....	111
6.4 Ajustes de la simulación y los controladores .....	113
<b>CAPÍTULO 7: Casos prácticos .....</b>	<b>119</b>
7.1 Operación manual.....	119
7.2 Controlador PID (Sintonización) .....	119
7.2.1 Obtención del sistema de primer orden aproximado .....	120
7.2.2 Sintonía Lambda Tuning PID sin filtro .....	121
7.2.3 Sintonía Lambda Tuning PID con filtro .....	123
7.2.4 Sintonía Lambda Tuning PI.....	126
7.2.5 . Sintonía López PID .....	128
7.3 Controlador Todo/Nada con histéresis .....	131
<b>CAPÍTULO 8: Conclusiones y trabajo futuro.....</b>	<b>133</b>
8.1 Conclusiones .....	133
8.2 Trabajo futuro.....	134
<b>CAPÍTULO 9: Bibliografía.....</b>	<b>137</b>
<b>CAPÍTULO 10: Anexos .....</b>	<b>141</b>



# Índice de figuras

Figura 1 - Fotografía del montaje real .....	18
Figura 2 - Documentación Easy Java/JavaScript Simulations .....	23
Figura 3 - Contenido de la carpeta del programa .....	23
Figura 4 - Contenido de la carpeta "workspace" .....	24
Figura 5 - Consola de EJS .....	24
Figura 6 - Pestaña "Descripción" de EJS .....	25
Figura 7 - Pestaña "Modelo" de EJS .....	26
Figura 8 - Pestaña "HtmlView" de EJS .....	27
Figura 9 - Menu de propiedades del elemento Tanque del conjunto de elementos de dibujo 2D .....	28
Figura 10 - Interfaz gráfica creada con estructura basada en bloques .....	28
Figura 11 - Elementos insertables en la interfaz .....	29
Figura 12 - Vista previa de la interfaz .....	30
Figura 13 - Ejecución de la simulación .....	30
Figura 14 - Archivos resultantes de empaquetar una simulación .....	31
Figura 15 - Página principal tras exportar la simulación (índice) .....	32
Figura 16 - Página de la simulación .....	32
Figura 17 - Información del autor de la simulación .....	33
Figura 18 - Esquema del montaje .....	36
Figura 19 - Tablero de Butcher .....	41

Figura 20 - Esquema de estructura de control en lazo cerrado.....	45
Figura 21 - Estructura de control con controlador PID paralelo.....	46
Figura 22 - Controlador PID ideal .....	50
Figura 23 - Esquema de señales al controlador PID ideal con filtro.....	53
Figura 24 - Señal de control de un controlador Todo/Nada.....	55
Figura 25 - Variable controlada en Control Todo/Nada con histéresis .....	56
Figura 26 - Variable controlada en Control Todo/Nada sin histéresis .....	56
Figura 27 - Paneles principales de la interfaz.....	59
Figura 28 - Barra de herramientas.....	59
Figura 29 - Estructura barra de herramientas.....	60
Figura 30 - Estructura sinóptico e ilustración.....	62
Figura 31 - Elementos de dibujo empleados en el sinóptico .....	63
Figura 32 - Grupo de dibujo del depósito 1.....	63
Figura 33 - Estructura de la representación de la bomba .....	64
Figura 34 - Aspecto visual de la bomba .....	64
Figura 35 - Estructura de instrumento.....	65
Figura 36 - Aspecto visual del caudalímetro .....	65
Figura 37 - Aspecto visual del montaje .....	65
Figura 38 - Estructura panel de gráficas .....	66
Figura 39 - Leyenda de la gráfica “Altura depósito” .....	66
Figura 40 - Aspecto visual gráficas .....	68

Figura 41 - Estructura panel de opciones .....	68
Figura 42 - Estructura panel de configuración de la simulación .....	69
Figura 43 - Estructura panel operación manual .....	71
Figura 44 - Interfaz operación manual .....	72
Figura 45 - Estructura panel controlador PID .....	72
Figura 46 - Interfaz controlador PID.....	73
Figura 47 - Estructura panel controlador Todo/Nada .....	74
Figura 48 - Interfaz controlador Todo/Nada .....	75
Figura 49 - Ecuaciones diferenciales ordinarias.....	84
Figura 50 - Distribución de números normales mediante Box Muller.....	90
Figura 51 - Microsoft Visual Studio.....	91
Figura 52 - Resultado del empaquetamiento de la simulación.....	92
Figura 53 - Sección "metadata" .....	93
Figura 54 - Archivos del programa final.....	93
Figura 55 - Gráfica altura deposito izquierdo del experimento 1 (validación).98	
Figura 56 - Gráfica señal de control del experimento 1 (validación) .....	98
Figura 57 - Gráfica señales de control del experimento 2 (validación) .....	100
Figura 58 - Gráfica altura deposito izquierdo del experimento 2 (validación) .....	100
Figura 59 - Gráfica señales de control del experimento 3 (validación) .....	103
Figura 60 - Gráfica altura deposito izquierdo del experimento 3 (validación) .....	103

Figura 61 - Archivos que componen el laboratorio virtual.....	105
Figura 62 - Interfaz del laboratorio virtual .....	106
Figura 63 - Página de descripción del laboratorio virtual.....	106
Figura 64 - Menú desplegable de tipología de control .....	107
Figura 65 - Archivo guardado datos .....	108
Figura 66 - Exportación de datos a Excel.....	109
Figura 67 - Cuadro de selección de archivo de datos para exportación de datos a Excel .....	109
Figura 68 - Cuadro de importación de datos a Matlab.....	110
Figura 69 - Datos importados a Matlab .....	110
Figura 70 - Página de créditos.....	111
Figura 71 - Sinóptico de la simulación.....	111
Figura 72 - Representación del desbordamiento del tanque .....	112
Figura 73 - Gráficas del laboratorio virtual.....	113
Figura 74 - Ajustes de la simulación .....	114
Figura 75 - Parámetros de Operación manual .....	115
Figura 76 - Parámetros de control PID.....	116
Figura 77 - Parámetros del controlador Todo/Nada con histéresis .....	116
Figura 78 - Evolución de la altura del depósito izquierdo .....	119
Figura 79 - Evolución temporal de la altura del depósito.....	120
Figura 80 - Ajuste por mínimos cuadrados.....	121

Figura 81 - Evolución altura depósito ante salto en la referencia con controlador PID sin filtro (Lambda Tuning).....	122
Figura 82 - Evolución señal de control ante salto en la referencia con controlador PID sin filtro (Lambda Tuning).....	122
Figura 83 - Evolución altura depósito ante perturbación con controlador PID sin filtro (Lambda Tuning).....	123
Figura 84 - Evolución señal de control ante perturbación con controlador PID sin filtro (Lambda Tuning).....	123
Figura 85 - Evolución altura depósito ante salto en la referencia con controlador PID con filtro (Lambda Tuning).....	124
Figura 86 - Evolución señal de control ante salto en la referencia con controlador PID con filtro (Lambda Tuning) .....	124
Figura 87 - Evolución altura depósito ante perturbación con controlador PID con filtro (Lambda Tuning).....	125
Figura 88 - Evolución señal de control ante perturbación con controlador PID con filtro (Lambda Tuning).....	125
Figura 89 - Evolución altura depósito ante salto en la referencia con controlador PI (Lambda Tuning) .....	126
Figura 90 - Evolución señal de control ante salto en la referencia con controlador PI (Lambda Tuning).....	127
Figura 91 Evolución altura depósito ante perturbación con controlador PI (Lambda Tuning) .....	127
Figura 92 - Evolución señal de control ante perturbación con controlador PI (Lambda Tuning) .....	128
Figura 93 - Tabla de sintonía de Lopez .....	128
Figura 94 - Evolución altura depósito ante salto en la referencia con controlador PID (López).....	129
Figura 95 - Evolución señal de control ante salto en la referencia con controlador PID (López) .....	130

Figura 96 - Evolución altura depósito ante perturbación con controlador PID (López).....	130
Figura 97 - Evolución señal de control ante perturbación con controlador PID (López).....	131
Figura 98 - Evolución altura del depósito con control On/Off .....	132
Figura 99 - Evolución señal de control con control On/Off .....	132
Figura 100 - Ejemplo laboratorio remoto.....	134
Figura 101 - Esquema comunicaciones servidor RIP .....	135
Figura 102 - Repositorio de UNILabs .....	135
Figura 103 - Objetivos de desarrollo sostenible.....	136



# CAPÍTULO 1: Introducción y Objetivos

---

## 1.1 Introducción

La elaboración del laboratorio virtual de un experimento físico permite a los usuarios realizar una experimentación práctica con una mayor flexibilidad. Su carácter informático permite su utilización desde cualquier ordenador con conexión a internet, eliminando la necesidad de acudir físicamente y en una franja horaria habilitada a las instalaciones reales.

Este trabajo se inspira en el proyecto UNILabs (UNED, s.f.), una iniciativa de la UNED (Universidad Nacional de Educación a Distancia), por la cual diversas universidades del mundo desarrollan diferentes laboratorios tanto virtuales como remotos y los comparten en un repositorio público. En pos de una formación flexible y universal, gran cantidad de simulaciones han sido ya compartidas en la red para cualquier usuario con interés en la materia.

El experimento a modelar pretende lograr facilitar la experimentación con los sistemas de control, tanto controladores PID como controladores Todo/Nada con histéresis. Adicionalmente, existe la opción de realizar un control manual de la planta mediante el cual alcanzar las condiciones de inicio deseadas o bien conocer el comportamiento del sistema.

El experimento consta de dos tanques comunicados en su parte inferior por una tubería (Figura 1). El depósito derecho (también denominado depósito 2) recibe un caudal de agua de una bomba, la cual es succionada de la bandeja. El depósito izquierdo (también denominado depósito 1) posee una válvula de salida, cuya apertura se puede regular de forma manual.

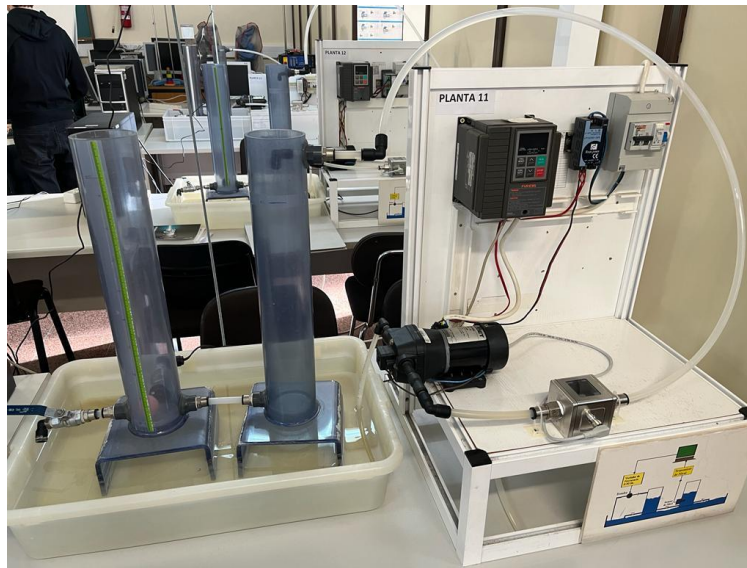


Figura 1 - Fotografía del montaje real

Además de los elementos que permiten el funcionamiento del sistema, han sido colocados ciertos instrumentos que permiten conocer su estado y evolución temporal. Tales son un caudalímetro, situado a la salida de la bomba, que indica el caudal de agua introducido en el sistema; y un medidor del nivel de agua del depósito izquierdo. La potencia de la bomba se regula mediante un ordenador situado en las inmediaciones que realiza las veces de controlador. La apertura de la válvula de salida, en cambio, no se realiza por instrumentación de precisión y su modificación se realiza girando manualmente la manilla de actuación.

En cuanto a las dimensiones relevantes del montaje, los depósitos poseen un diámetro de 8 cm y una altura de 50 cm. La tubería que los comunica posee una longitud de 15 cm y un diámetro de 1 cm. La válvula de salida, al encontrarse abierta en su totalidad, permite el paso del fluido por una sección de también 1 cm de diámetro. La bomba es capaz de proporcionar una caudal máximo de 11 L/min.

## 1.2 Objetivos

Se busca la elaboración de un programa informático desarrollado mediante EjsS (Easy Java/Javascript Simulations) que modele con suficiente precisión el experimento descrito. Dicho programa permitirá visualizar la evolución temporal del sistema y controlarlo mediante operación manual, controladores PID o bien controladores todo/nada con histéresis. Los parámetros de los

### *Laboratorio virtual de una planta piloto de dos tanques comunicantes*

controladores deberán poder modificarse, siendo el objetivo docente del programa que futuros usuarios puedan probar la sintonización de controladores PID, y conocer el uso de controladores Todo/Nada. Otro objetivo complementario sería que los estudiantes puedan practicar con el sistema en simulación antes de enfrentarse al sistema físico real. Adicionalmente, los resultados de la simulación realizada se deberán poder almacenar en un archivo de texto plano para posteriores análisis. El programa incluirá también una descripción del experimento realizado y de sus propias funcionalidades.



# CAPÍTULO 2: Easy Java/JavaScript Simulations

---

## 2.1 Introducción

Easy Java/JavaScript Simulations (Esquembre, 2013), también conocido como EJS o EJSs, es un entorno de desarrollo creado por Francisco Esquembre, catedrático de Análisis Matemático en el departamento de Matemáticas de la Universidad de Murcia, y distribuido bajo una licencia tipo GNU GLP, la cual garantiza el libre acceso al software, así como el derecho a modificarlo, siempre y cuando se reconozca la autoría de su creador (intención de este breve epígrafe).

Este entorno de desarrollo, creado principalmente con intención pedagógica, busca facilitar la creación de programas informáticos que lleven a cabo simulaciones de fenómenos reales de interés. Una simulación no es más que “la acción y el efecto de representar algo, fingiendo o imitando lo que no es” (RAE, s.f.). Definiéndolo de una forma más apropiada para el contexto en el que nos encontramos, una simulación viene a ser “el proceso de diseñar y desarrollar un modelo computarizado de un sistema o proceso y conducir experimentos con este modelo con el propósito de entender el comportamiento del sistema o evaluar varias estrategias con las cuales se puede operar” (Shannon, 2012).

El beneficio que aporta EJS radica en permitir al usuario inexperto en lenguajes de programación crear la simulación deseada sin ahondar en cuestiones técnicas de desarrollo informático. Esto se logra facilitando la creación de la interfaz gráfica sin más que arrastrar los elementos dentro de la zona de trabajo y asignándoles sus propiedades. Además, permite describir el fenómeno a estudio introduciendo las ecuaciones diferenciales del modelo y seleccionando el método numérico de resolución. El programa se encarga de traducir la interfaz indicada por el usuario al lenguaje de programación y calcular la evolución temporal del sistema mediante la iteración de un método numérico sobre las ecuaciones diferenciales. Esto se podría traducir en que el mérito del programa se encuentra en lograr que el usuario realice una programación de más alto nivel que la habitual en el desarrollo de este tipo de aplicaciones. El creador puede redactar también una página de descripción que

se mostrará al futuro usuario de la simulación con la cual explicar su uso y su fundamento teórico.

Gracias al cumplimiento de estos objetivos, el programa ha recibido numerosos reconocimientos y galardones de gran prestigio tales como el “Premio Global Online del Consorcio de Laboratorios” (premio GOLC), el “Premio de la Sociedad Física Americana a la Excelencia en la Educación de Física” o el “Premio al contenido multimedia en Enseñanza y Aprendizaje de Física” entre otros. Habiendo recibido también dos premios de la Unesco: uno primero por parte de la oficina de educación de Asia y la región del Pacífico y otro segundo, premio rey Hamad Bin Isa Al-Khalifa por el uso de nuevas tecnologías en la educación.

Ahondando ahora en el aspecto técnico del programa, existen dos versiones del mismo. La primera, basada en Java, permite la creación de programas de ejecución local gracias a la máquina virtual de Java, propiedad de Oracle. Esta versión ha quedado discontinuada debido a las ventajas que proporciona su alternativa. La segunda versión, basada en JavaScript, HTML y CSS, permite la creación de simulaciones ejecutables sobre navegador, permitiendo de esta manera ejecutar el programa accediendo a un directorio web, sin necesidad de descargar ningún archivo, o bien descargando los archivos correspondientes y ejecutándola de manera local, de forma análoga a la versión basada en Java. No obstante, la versión basada en JavaScript sigue presentando la ventaja de permitir ambos tipos de acceso sin requerir modificaciones adicionales.

Otras de las novedades debidas a la proliferación de la versión de JavaScript es la compatibilización en últimas versiones con los lectores de los dispositivos móviles. Esto supone que tabletas y móviles pueden ahora ejecutar también la simulación gracias a su compatibilidad con las aplicaciones web. Como comentario adicional indicar que la librería que ha permitido este avance ha sido posible por la colaboración de Félix J. García Clemente, doctor en Informática y Catedrático de la Universidad de Murcia en el área de Arquitectura y Tecnología de Computadores.

Existen diferentes versiones de EJS, siendo la última versión estable de JavaScript la 6.01, aunque existe también una versión beta 6.02. De forma análoga, la última versión estable de la versión Java es la 6.02, la cual a diferencia de en JavaScript ya ha sido depurada.

Toda explicación y referencia hecha a partir de este punto se realizará sobre la versión 6.01 de JavaScript, la cual es la elegida para la realización del proyecto.

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

Este entorno de desarrollo cuenta con su propia documentación debido a incorporar métodos y funciones no nativas de JavaScript, CSS y HTML (Figura 2). Así como librerías que es posible utilizar. Es posible también consultar la documentación propia de cada lenguaje dado que sobre ellos se construye la simulación y por tanto hereda todas sus funcionalidades.

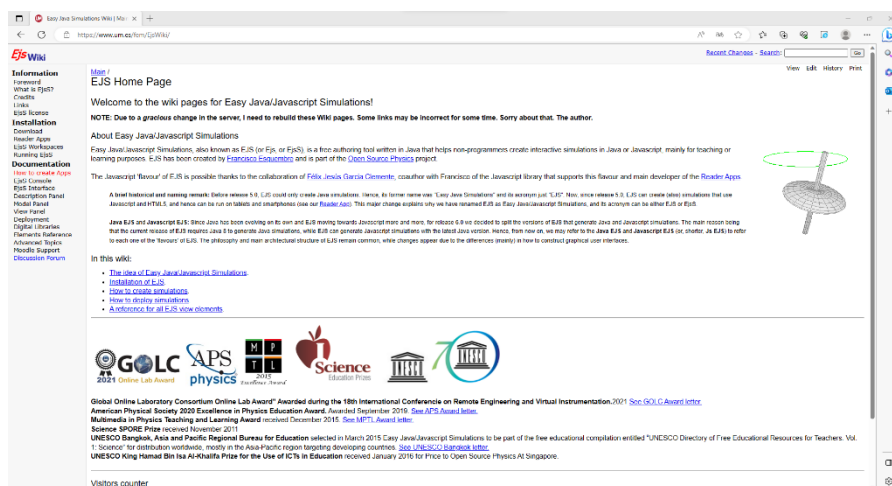


Figura 2 - Documentación Easy Java/JavaScript Simulations

## 2.2 Instalación

Para instalar el entorno EJS basta con descargar del repositorio online el archivo comprimido correspondiente a la versión deseada. Posteriormente, se extrae la carpeta con los archivos del programa (Figura 3).

Nombre	Fecha de modificación	Tipo	Tamaño
bin	28/01/2023 13:41	Carpeta de archivos	
doc	28/01/2023 13:41	Carpeta de archivos	
workspace	27/06/2023 18:06	Carpeta de archivos	
EjsConsole.jar	13/10/2020 21:46	Executable Jar File	3.060 KB

Figura 3 - Contenido de la carpeta del programa

La carpeta “bin” incluye archivos internos del programa y la carpeta “doc” documentación y ejemplos. En “workspace” se almacenan los archivos de las simulaciones creadas y “EjsConsole.jar” abre el entorno de desarrollo propiamente dicho. No es necesaria su instalación por lo que se trata de un

programa portable. Dentro de “workspace” se encuentran a su vez una serie de carpetas (Figura 4).

La carpeta “config” incluye un fichero de texto plano con la configuración del programa. La carpeta “export” incluye los archivos generados cuando se empaqueta la simulación (se da por terminado el desarrollo y se exporta). La carpeta “output” es en la cual se generan los archivos tipo “xhtml” para poder visualizar la simulación en la etapa de desarrollo y la carpeta “source” es en la cual se almacenan los archivos con el código fuente tipo “ejs”.

Nombre	Fecha de modificación	Tipo	Tamaño
config	28/01/2023 13:41	Carpeta de archivos	
export	26/06/2023 0:15	Carpeta de archivos	
output	25/06/2023 0:13	Carpeta de archivos	
source	25/06/2023 17:31	Carpeta de archivos	

Figura 4 - Contenido de la carpeta "workspace"

## 2.3 Desarrollo de aplicaciones en EJS

Una vez conocido el sistema de archivos del programa, pasemos a comentar el procedimiento de creación de una simulación. Tras ejecutar el archivo “EjsConsole.jar” se abren dos ventanas. La primera es la consola (Figura 5), la cual permite cambiar tanto configuración básica y avanzada del entorno como visualizar el estado del programa por medio de los mensajes que aparecen en el Área de mensajes. Esta ventana será escasamente utilizada.

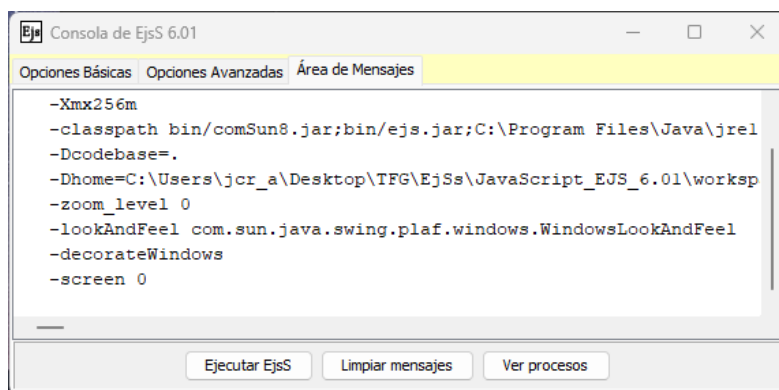
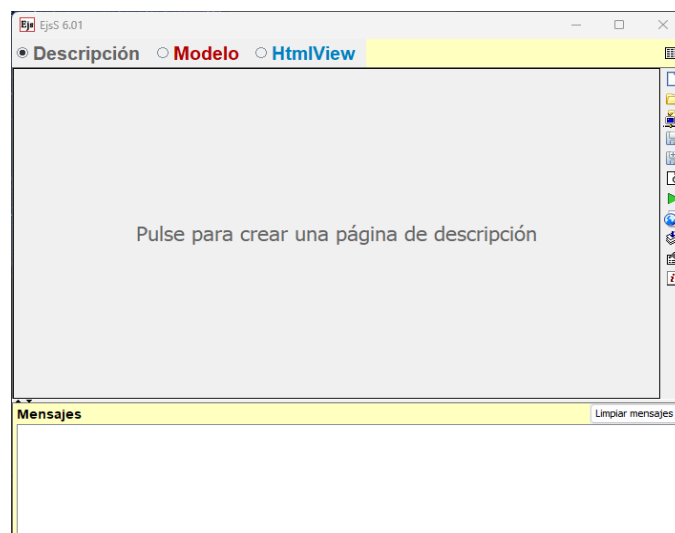


Figura 5 - Consola de EJS



La otra ventana (Figura 6) es la interfaz del programa, donde se realiza mayoritariamente el trabajo. Consta de tres pestañas las cuales son “Descripción, Modelo y HtmlView”. Adicionalmente, consta de una barra de herramientas vertical a la derecha de la interfaz y de un área de mensajes en la parte inferior. Siendo estos dos últimos elementos comunes a todas las pestañas. En la barra de herramientas se encuentran opciones tales como abrir archivo, guardar, buscar en el código, ejecutar, empaquetar, etc.

El área de mensajes realiza una función similar a la consola. En ella aparece el estado del entorno el cual puede ser: compilando, guardando, exportando, etc. Mediante un botón situado en la esquina superior derecha se puede indicar información global a la simulación, tal como autoría y tipo de licencia, opciones de ejecución y de edición, entre otros.



*Figura 6 - Pestaña “Descripción” de EJS*

### **2.3.1 Descripción**

En la pestaña en la que nos encontramos (Figura 6), se crea la página que explica al futuro usuario la simulación creada y su fundamento teórico. Su formato es HTML y se abrirá como una ventana emergente. Es posible redactarla en texto plano como si de un editor de textos se tratase sin necesidad de programación. No obstante, si se desea una página más pulida es posible trabajar directamente sobre el código para asegurar el resultado final.

## 2.3.2 Modelo

La pestaña Modelo (Figura 7), en cambio, se utiliza para expresar el comportamiento del sistema de forma entendible para la máquina con el fin de que calcule su evolución temporal. Consta a su vez de seis subpestañas. Las cuales son “Variables, Inicialización, Evolución, Relaciones fijas, Propio y Elementos”.

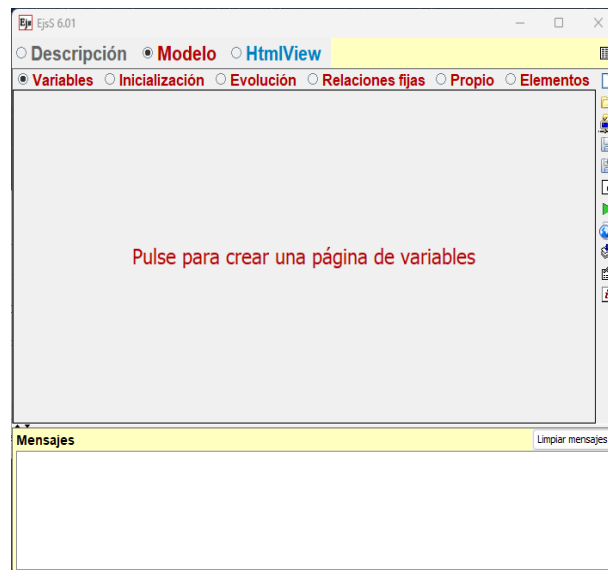


Figura 7 - Pestaña "Modelo" de EJS

El apartado “Variables” permite, como su nombre indica, declarar e instanciar las variables necesarias para el funcionamiento del sistema. En él se indica el nombre de la variable, su valor inicial, su tipo y su dimensión (para el caso de los arrays).

El apartado “Inicialización” incluye el código que es necesario ejecutar de forma inmediata tras ejecutar la simulación.

En el apartado “Evolución” se indican las ecuaciones diferenciales que describen el sistema, así como el método numérico empleado para su resolución. Es posible también expresar la evolución del sistema en forma de código para indicar cambios bruscos o valores condicionales (entre otros). En este apartado se realizan ciertos ajustes sobre el cálculo de la evolución temporal del sistema tales como modificar el periodo, la tolerancia del método numérico, el comportamiento ante errores o discontinuidades, eventos, el máximo número de pasos por iteración, etc.

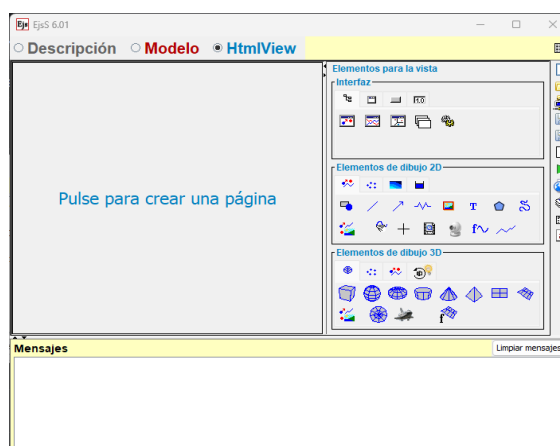
En el apartado “Relaciones fijas” se incluye código que se ejecuta tras cada cálculo del método numérico. Típicamente se describen relaciones entre variables, pero pueden incluirse más funcionalidades como cambios en la interfaz.

En el apartado “Propio” pueden definirse métodos o funciones del usuario a las cuales llamar desde otras partes del código. Permite sustituir líneas repetitivas en el programa por una llamada a la función con el código a ejecutar.

El apartado “Elementos” permite añadir librerías preinstaladas en el entorno EJS a la simulación a desarrollar. Esto implica incluir funciones ya programadas por ser habituales en la generación de simulaciones. Este es el caso de las librerías “Hardware, InputOutput, Numerics y SoftwareLinks”, estando “Numerics” enfocada en el cálculo matemático avanzado y el resto en facilitar la conexión con laboratorios remotos o con el hardware del dispositivo o bien en exportar datos de la simulación al dispositivo local.

### 2.3.3 HtmlView

La pestaña “HtmlView” permite la creación de la interfaz (Figura 8). Es posible crear más de una página para conmutar entre ellas.



*Figura 8 - Pestaña "HtmlView" de EJS*

Para incluir elementos en la interfaz de la simulación basta con arrastrar el elemento deseado al árbol de elementos. Para modificar propiedades tales como su posición, su estilo, o su función, basta con hacer doble clic sobre él. Se abre un menú que resume las principales características modificables (Figura 9).

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

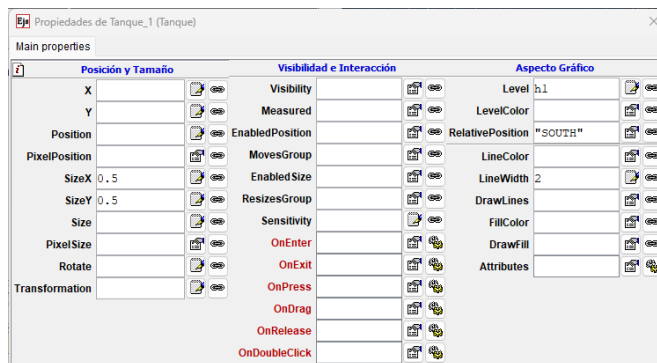


Figura 9 - Menu de propiedades del elemento Tanque del conjunto de elementos de dibujo 2D

En el caso de no encontrar la propiedad de estilo deseada puede añadirse en el recuadro “CSS” el código de estilo CSS correspondiente. Análogamente, en el caso de no encontrar el ajuste de evento deseado se añadiría el código JavaScript pertinente en la pestaña “Modelo” haciendo referencia a la instancia del elemento HTML pertinente. Cabe recordar que el entorno EJS es una simplificación del proceso de creación de una página web usando HTML, CSS y JavaScript. Se han incluido las funcionalidades habituales en el programa, pero no su totalidad. Para acceder a una función nativa de uno de los lenguajes no disponible en EJS, basta con utilizar la programación en el lenguaje original en forma de código.

La organización de los elementos en la página sigue una estructura de etiquetas basada en HTML, centrado en el uso de bloques (Figura 10). De ser necesario también se puede incluir código HTML con el cual lograr el resultado esperado.

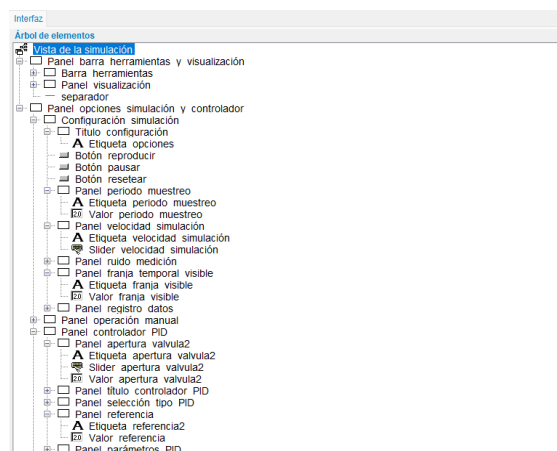
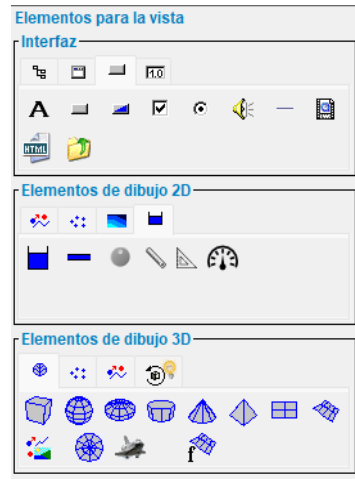


Figura 10 - Interfaz gráfica creada con estructura basada en bloques

Todos los elementos deben colocarse sobre paneles, los cuales son la base en la creación de la interfaz representando un bloque. Además de los elementos habituales en el diseño web (Figura 11) como pueden ser paneles, etiquetas de texto, botones, menús desplegables, etc.



*Figura 11 - Elementos insertables en la interfaz*

Se pueden incorporar elementos destinados especialmente para su uso en simulaciones no nativos del lenguaje HTML. Tal es el caso de gráficas, elementos de vídeo y audio, selectores de archivos con los cuales cargar datos o lienzos de dibujo 2D o 3D. Para los lienzos de dibujo 2D y 3D existe una colección de elementos precargados.

Tal es el caso de, para un lienzo 2D, formas, segmentos, flechas, textos o elementos ya dibujados como depósitos, tuberías, reglas, cartabones o medidores entre otros. Existe también la posibilidad de incrustar imágenes, videos o una retransmisión de imagen en directo desde una webcam.

En el caso de un lienzo 3D, es posible incluir Figuras ya dibujadas tales como prismas, esferas, tetraedros, conos o modelos 3D de archivos externos, entre otros. Al igual que en el modelo 2D, es posible incluir texto, segmentos, rayas o incrustar imágenes.

Es posible observar los cambios realizados en la interfaz mediante la vista previa, la cual se actualiza con cada cambio en el código. Esta consiste en una ventana abierta automáticamente en el navegador predeterminado del usuario (Figura 12).

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

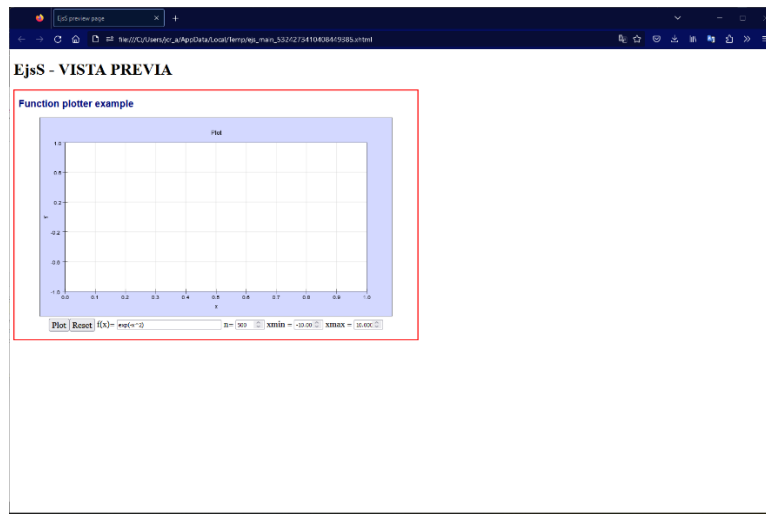


Figura 12 - Vista previa de la interfaz

En la propia página se indica que se trata de la vista previa y es importante notar que, aunque se muestre la interfaz del programa no es la simulación propiamente dicha y por tanto no funciona.

Para ejecutar la simulación se usa el botón “Ejecutar la simulación” situado sobre la barra vertical de herramientas de la parte derecha. De esta manera sí se compila el código generando una aplicación completamente funcional, que se almacena en la carpeta “Output”. Esta se abrirá también en el navegador predeterminado (Figura 13).

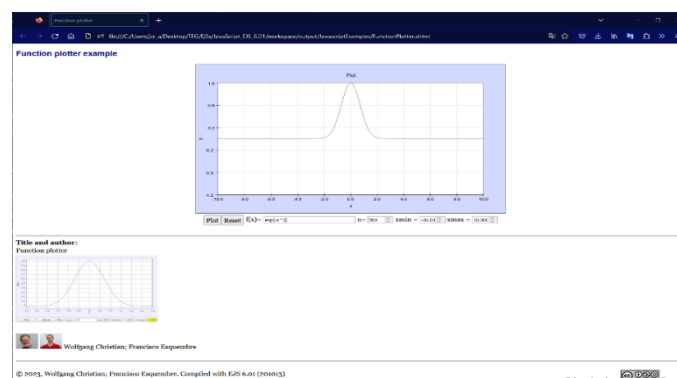
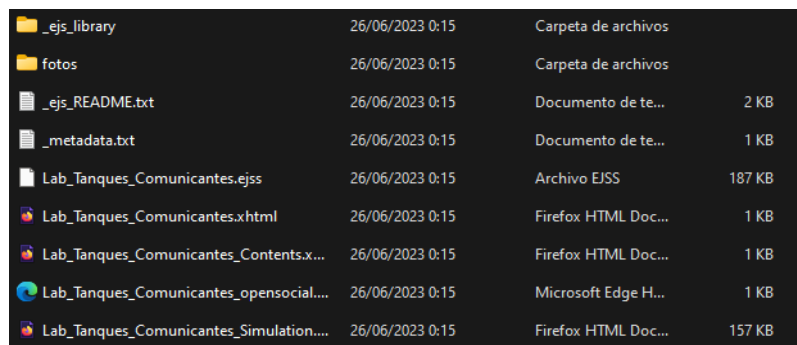


Figura 13 - Ejecución de la simulación

## 2.4 Empaquetamiento de la simulación

Una vez terminado el desarrollo del programa se hace necesario poder ejecutarlo sin necesidad de tener instalado el entorno Easy Java/JavaScript Simulations. Para ello, se hace uso del botón “Empaquetar”, y el programa genera un archivo comprimido (Figura 14).



Nombre	Fecha	Descripción	Tamaño
_ejs_library	26/06/2023 0:15	Carpeta de archivos	
fotos	26/06/2023 0:15	Carpeta de archivos	
_ejs_README.txt	26/06/2023 0:15	Documento de te...	2 KB
_metadata.txt	26/06/2023 0:15	Documento de te...	1 KB
Lab_Tanques_Comunicantes.ejss	26/06/2023 0:15	Archivo EJSS	187 KB
Lab_Tanques_Comunicantes.xhtml	26/06/2023 0:15	Firefox HTML Doc...	1 KB
Lab_Tanques_Comunicantes_Contents.x...	26/06/2023 0:15	Firefox HTML Doc...	1 KB
Lab_Tanques_Comunicantes_opensocial....	26/06/2023 0:15	Microsoft Edge H...	1 KB
Lab_Tanques_Comunicantes_Simulation....	26/06/2023 0:15	Firefox HTML Doc...	157 KB

Figura 14 - Archivos resultantes de empaquetar una simulación

Estudie como ejemplo el caso de este proyecto: Se genera una carpeta denominada “\_ejs\_library” con las librerías no nativas de HTML, CSS y JavaScript necesarias para la ejecución de la simulación. Adicionalmente, se crea una carpeta con los recursos multimedia que emplea el programa. En este caso se llama fotos dado que en su creación esa ha sido la elección.

Aparece también un fichero de tipo texto plano denominado “\_ejs\_README.txt” que describe la función de cada archivo del comprimido, y otro fichero de la misma tipología llamado “\_metadata.txt”, el cual expresa la autoría del entorno de desarrollo EJS así como la información de la cual el creador de la simulación haya querido dejar constancia (Universidad de origen, nombre, foto, licencia, palabras clave, comentarios, etc).

El fichero “Lab\_Tanques\_Comunicantes.ejss” contiene el código propiamente dicho y en caso de querer modificar el programa este es el cual se debe editar.

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

Quedan también 3 ficheros tipo “xhtml” y un fichero “xml”. Ejecutando el fichero principal denominado “Lab\_Tanques\_Comunicantes.xhtml”, se abre una página de navegador (Figura 15) cuyo contenido es un índice situado a la izquierda de las páginas de descripción e interfaz del programa.

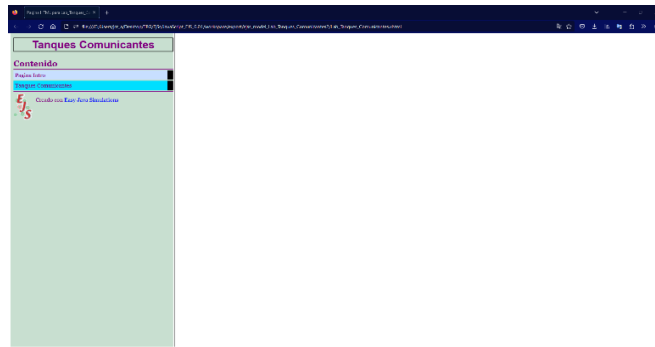


Figura 15 - Página principal tras exportar la simulación (índice)

Posee también el icono del entorno de desarrollo Easy Java/JavaScript Simulations junto a un enlace a su página oficial. Haciendo clic sobre el elemento deseado en el índice, se abre este en una ventana emergente (Figura 16).

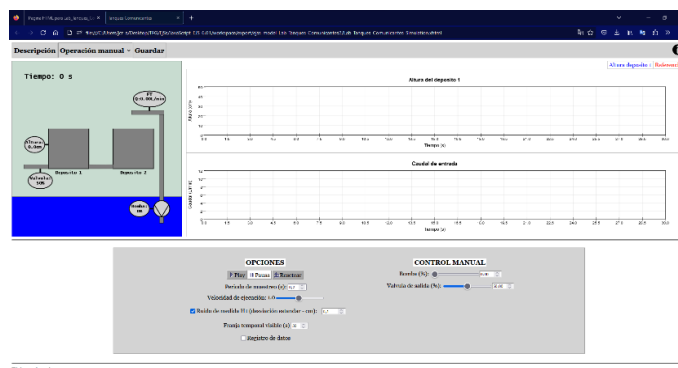


Figura 16 - Página de la simulación



## Laboratorio virtual de una planta piloto de dos tanques comunicantes

Cabe también destacar que el entorno EJS crea automáticamente una sección en la parte inferior de la página web en la que se expresa información relevante a la autoría de la simulación. Ver ejemplo en la simulación de la caída libre de un cuerpo (Figura 17).

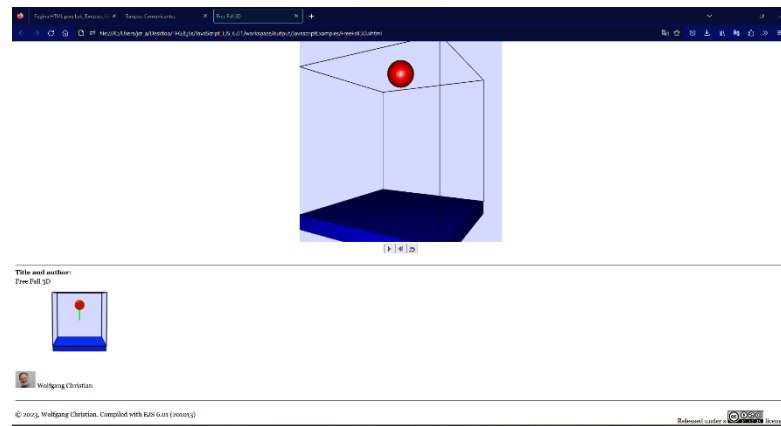


Figura 17 - Información del autor de la simulación

Esta sección no se considerará de interés y se eliminará. El reconocimiento de la autoría del proyecto se realizará por otros medios.



# CAPÍTULO 3: Fundamento teórico

---

## 3.1 Modelado de dos tanques comunicantes

Para realizar una simulación del sistema es necesario obtener un modelo del fenómeno descrito. Un modelo es “una representación precisa de la dinámica de un sistema que se utiliza para responder preguntas a través del análisis y la simulación” (Åström & Murray, 2021).

El primer paso en la elaboración del mismo es indicar las hipótesis de modelado aplicadas. En el caso de dos tanques comunicantes, las hipótesis tomadas son las siguientes:

- Condiciones de experimentación constantes de 1 atm y 25°C (condiciones estándar).
- Efectos de pérdidas de agua por evaporización u otros efectos despreciables.
- Efecto de la tubería que une los depósitos sobre la dinámica del sistema despreciable. Se supone longitud de tubería nula. La tubería implicaría un pequeño retardo en la evolución del sistema, pero no es tenido en cuenta.
- Tubería llena en toda su sección independientemente del nivel de agua
- Bomba isoentrópica que provoca cambios de temperatura despreciables en el caudal de agua de entrada.
- El fluido de trabajo es agua pura con una densidad constante e igual a 1 Kg/L.

Una vez conocidos los axiomas de partida se busca la obtención de las ecuaciones que describen la evolución temporal del sistema mediante los conocimientos obtenidos en la materia de Fluidomecánica. Realizamos un esquema representativo del montaje con las variables relevantes en el estudio (Figura 18).

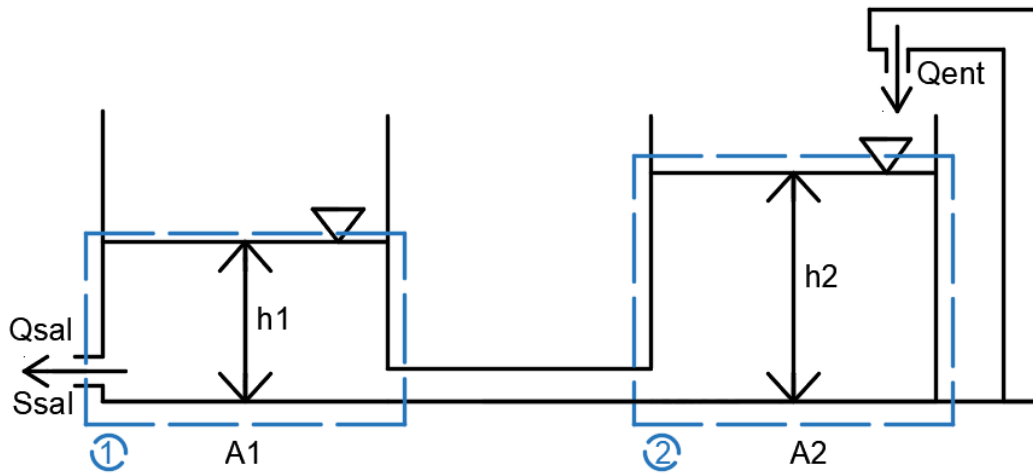


Figura 18 - Esquema del montaje

La tabla a continuación explica el significado de cada abreviatura.

$h_1$	Altura del depósito 1 (izquierdo)
$h_2$	Altura del depósito 2 (derecho)
$A_1$	Diámetro del depósito 1 (izquierdo)
$A_2$	Diámetro del depósito 2 (derecho)
$Q_{ent}$	Caudal de entrada
$Q_{tub}$	Caudal que atraviesa la tubería
$Q_{sal}$	Caudal de salida
$S_{tub}$	Sección de la tubería
$S_{sal}$	Sección de la válvula de salida
$v_{tub}$	Velocidad del agua en la tubería
$v_{sal}$	Velocidad del agua en la salida
$\rho$	Densidad del agua
$g$	Aceleración de la gravedad
$\vec{v}_r$	Velocidad relativa a la superficie de control

$\vec{n}$	Normal exterior a la superficie de control
SC	Superficie de control
VC	Volumen de control

A continuación, se aplica la ley de Conservación de la materia aplicada al caso de fluidos estudiando volúmenes de control variables definidos como la masa de agua alojada dentro de cada uno de los depósitos (ambos con forma cilíndrica).

Siendo:

Aplicando la ecuación de conservación de la masa al volumen de control 2:

$$0 = \frac{d}{dt} \int_{VC} \rho \, dV + \int_{SC} \rho \cdot (\vec{v}_R \cdot \vec{n}) \, dA \quad (3.1)$$

$$0 = \frac{d}{dt} (\rho \cdot A_2 \cdot h_2) + \rho \cdot (Q_{tub} - Q_{ent}) \quad (3.2)$$

$$\frac{d}{dt} h_2 = \frac{(Q_{ent} - Q_{tub})}{(A_2)} \quad (3.3)$$

Aplicando la ecuación de conservación de la masa al volumen de control 1:

$$0 = \frac{d}{dt} \int_{VC} \rho \, dV + \int_{SC} \rho \cdot (\vec{v}_R \cdot \vec{n}) \, dA \quad (3.4)$$

$$0 = \frac{d}{dt} (\rho \cdot A_1 \cdot h_1) + \rho \cdot (Q_{sal} - Q_{tub}) \quad (3.5)$$

$$\frac{d}{dt} h_1 = \frac{(Q_{\text{tub}} - Q_{\text{sal}})}{(A_1)} \quad (3.6)$$

El valor de  $Q_{\text{ent}}$  viene dado por la bomba por lo que es un valor conocido. El valor de  $Q_{\text{tubería}}$  y  $Q_{\text{sal}}$  dependen de las alturas de los depósitos. Buscamos su relación.

El teorema de Torricelli afirma que “La velocidad de un líquido en una vasija abierta, por un orificio, es la que tendría un cuerpo cualquiera, cayendo libremente en el vacío desde el nivel del líquido hasta el centro de gravedad del orificio” (Teorema de Torricelli, s.f.). Lo cual se expresa mediante la siguiente relación matemática:

$$v = \sqrt{2gh} \quad (3.7)$$

Siendo  $h$  la distancia entre el orificio y la superficie libre del fluido.

Si se tienen dos depósitos comunicados, se puede describir el comportamiento de los tanques comunicantes suponiendo que la velocidad del fluido en el interior de la tubería es proporcional a la raíz cuadrada de la diferencia de distancias de cada superficie libre al orificio. Como se supone que la altura del depósito 2 va a ser mayor que la del 1, es posible afirmar que el fluido se desplazará en la dirección que va desde el depósito derecho al izquierdo.

$$v_{\text{tub}} = \sqrt{2g(h_2 - h_1)} \quad (3.8)$$

El caudal que atraviesa la tubería se puede calcular sin más que relacionar la velocidad del fluido con su sección.

$$Q_{\text{tub}} = S_{\text{tub}} \cdot v_{\text{tub}} \quad (3.9)$$

$$Q_{\text{tub}} = S_{\text{tub}} \cdot \sqrt{2g(h_2 - h_1)} \quad (3.10)$$

Siendo  $S_t$  la sección de la tubería y  $v_{\text{tubería}}$  la velocidad del fluido que atraviesa la tubería.

Aplicando de nuevo el teorema de Torricelli se puede expresar la velocidad del agua que abandona el depósito 1 como:

$$v_{\text{sal}} = \sqrt{2gh_1} \quad (3.11)$$

El caudal de salida, el cual es el caudal de agua que abandona el depósito 1, puede calcularse por tanto como el producto de la velocidad de la corriente de agua de salida por la sección de la válvula de salida.

$$Q_{\text{sal}} = S_{\text{sal}} \cdot v_{\text{sal}} \quad (3.12)$$

$$Q_{\text{sal}} = S_{\text{sal}} \cdot \sqrt{2gh_1} \quad (3.13)$$

Sustituyendo la ecuación (3.10) en (3.3) se obtiene la ecuación que describe la variación de la altura del depósito 2 en función de la altura del depósito 1 y de la altura del depósito 2.

$$\frac{dh_2}{dt} = \frac{Q_{\text{ent}} - S_{\text{tub}} \cdot \sqrt{2g(h_2 - h_1)}}{A_2} \quad (3.14)$$

De igual manera, sustituyendo la ecuación (3.10) y (3.13) en (3.6) se obtiene la variación de la altura del depósito 1 en función de las alturas de los depósitos 1 y 2.

$$\frac{dh_1}{dt} = \frac{S_{\text{tub}} \cdot \sqrt{2g(h_2 - h_1)} - S_{\text{sal}} \cdot \sqrt{2gh_1}}{A_1} \quad (3.15)$$

Las funciones (3.14) y (3.15) describen la evolución temporal del sistema para un  $Q_{\text{ent}}$  conocido.

### 3.2 Resolución mediante el método numérico Runge – Kutta

No se conoce la solución analítica de todas las ecuaciones diferenciales y en muchos casos es necesario recurrir al uso de métodos numéricos. Los métodos numéricos para ecuaciones diferenciales ordinarias son procedimientos utilizados para encontrar aproximaciones numéricas a las soluciones de ecuaciones diferenciales ordinarias (EDO).

Para la resolución de las ecuaciones diferenciales planteadas se va a utilizar el método Runge–Kutta 4 (RK4) (Método de Runge-Kutta, 2023).

Los métodos Runge-Kutta son un conjunto de métodos genéricos iterativos, explícitos e implícitos para la resolución numérica de ecuaciones diferenciales. Fueron desarrollados en torno al año 1900 por los matemáticos Carl David Tomé Runge y Martin Wilhelm Kutta y permiten la resolución de problemas de valor inicial. Su razonamiento matemático es el siguiente:

Considerando el intervalo  $[a, b]$  y con tamaño de paso  $h$ . Se trata de calcular aproximaciones a la solución exacta en los puntos  $t_j = a + jh, j = 0, \dots, M$ , siendo  $M = (b - a)/h$ , que denotaremos por  $y_j, j = 0, \dots, M$ .

Dado  $y_0$ , un método de Runge-Kutta general de  $s$  etapas, está definido por:

$$\left\{ \begin{array}{l} K_1 = f(t_j + c_1 h, y_j) + h \sum_{k=1}^s a_{1k} K_k \\ \vdots \\ K_i = f(t_j + c_i h, y_j) + h \sum_{k=1}^s a_{ik} K_k \\ \vdots \\ K_s = f(t_j + c_s h, y_j) + h \sum_{k=1}^s a_{sk} K_k \end{array} \right. \quad (3.16)$$



$$\left\{ \begin{array}{l} y_{j+1} = y_j + h \sum_{i=1}^s b_i K_i \end{array} \right.$$

Donde se asumen las siguientes condiciones.

$$c_i = \sum_{k=1}^s a_{ik}, \quad i = 1, 2, \dots, s \quad (3.17)$$

Es posible representar esta familia de métodos mediante su tablero de Butcher (Figura 19):

$c_1$	$a_{11}$	$a_{12}$	$\cdots$	$a_{1s}$
$c_2$	$a_{21}$	$a_{22}$	$\cdots$	$a_{2s}$
$\vdots$	$\vdots$			$\vdots$
$c_s$	$a_{s1}$	$a_{s2}$	$\cdots$	$a_{ss}$
	$b_1$	$b_2$	$\cdots$	$b_s$

Figura 19 - Tablero de Butcher

Clasificándose el método en:

El método es explícito si  $a_{ik} = 0$  para  $k \geq i$ ,  $k = 1, 2, \dots, s$ .

El método es semi-implícito si  $a_{ik} = 0$  para  $k \geq i$ ,  $k = 1, 2, \dots, s$ .

El método es implícito si  $a_{ik} \neq 0$  para algún  $k > i$ .

En cuanto a la convergencia y el orden. Los métodos Runge-Kutta son métodos de un paso

$$y_{j+1} = y_j + h\Phi_f(y_j, t_j; h) \quad (3.18)$$

Siendo

$$\Phi_f(y_j, t_j; h) = \sum_{i=1}^s b_i K_i \quad (3.19)$$

Todo método debe verificar que en el límite cuando  $h$  tiende a 0, la solución numérica  $\{y_j : j = 0, 1, \dots, M = (b - a)/h\}$  calculada mediante el método numérico converja hacia la solución exacta  $y(t)$ ,  $t \in [a, b]$ . Un método es convergente cuando verifica esta propiedad. Un método Runge-Kutta es convergente si y sólo si:

$$\sum_{i=1}^s b_i = 1 \quad (3.20)$$

En cuanto al error, sea el método siguiente

$$y_0 \text{ dado} \quad (3.21)$$
$$y_{j+1} = y_j + h\Phi_f(y_j, t_j; h), j = 0, \dots, M - 1$$

se definen los errores de truncación local del método en los nodos  $t_j = a + jh$ ,  $j = 1, \dots, M$  como

$$\rho_j = \frac{y(t_j) - y(t_{j-1})}{h} - \Phi_f(y(t_{j-1}), t_{j-1}; h) \quad (3.22)$$

Para  $j=1, \dots, M$ , donde  $y(t)$  es la solución exacta del problema de valores iniciales.

Los errores de truncación global del método en los nodos  $t_j = a + jh$ ,  $j = 1, \dots, M$  se definen como

$$e_j = y(t_j) - y_j \quad (3.23)$$

Se dice que el método tiene orden  $p$  si este es el mayor entero para el cual

$$y(t+h) - y(t) - h\Phi_f(y(t), t_j; h) = O(h^{p+1}) \quad (3.24)$$

Donde  $y(t)$  es la solución exacta del problema de valor inicial.

Se deduce inmediatamente a partir de las definiciones de error de truncación local y de orden que si el método (3.18) tiene orden  $p$  entonces los errores de truncación local verifican

$$\rho_j = O(h^p), \quad j = 1, \dots, M \quad (3.25)$$

Y por tanto los errores de truncación local convergen a cero más deprisa cuanto mayor sea el orden del método. Además, si  $\rho_j = O(h^p)$ , entonces los errores globales  $e_j = O(h^p)$ .

Respecto del método Runge-Kutta 4 (RK4) utilizado como método numérico en el cálculo de la simulación, se trata de un método de orden 4, de 4 etapas, que trabaja bien en la mayoría de las situaciones. Es seguramente el más popular de los métodos Runge-Kutta debido a la facilidad de cálculo de sus coeficientes antes de la aparición de la computación y es comúnmente conocido como RK4.

Se define como:

$$\left\{ \begin{array}{l} K_1 = f(t_j, y_j), \\ K_2 = f(t_j + (h/2), y_j + (h/2) K_1), \\ K_3 = f(t_j + (h/2), y_j + (h/2) K_2), \\ K_4 = f(t_j + (h/2), y_j + (h/2) K_3), \\ y_{j+1} = y_j + (h/6) \cdot (K_1 + 2K_2 + 2K_3 + K_4) \end{array} \right. \quad (3.26)$$

Así, el siguiente valor  $y_{n+1}$  es determinado por el presente valor  $y_n$  más el producto del tamaño del intervalo  $h$  por una pendiente estimada. La pendiente es un promedio ponderado de pendientes, donde  $K_1$  es la pendiente al principio del intervalo,  $K_2$  es la pendiente en el punto medio del intervalo, usando  $K_1$  para determinar el valor de  $y$  en el punto  $x_{n+h/2}$  usando el método

de Euler.  $K_3$  es de nuevo la pendiente del punto medio, pero ahora usando  $K_2$  para determinar el valor de  $y$ ;  $K_4$  es la pendiente al final del intervalo, con el valor de  $y$  determinado por  $K_3$ .

$$\text{pendiente} = \frac{K_1 + 2K_2 + 2K_3 + K_4}{6} \quad (3.27)$$

Esta tipología de la familia Runge-Kutta es un método de cuarto orden, lo cual implica que el error por paso es del orden de  $O(h^5)$ , mientras que el error total acumulado tiene el orden  $O(h^4)$ . Es por ello, que la convergencia del método es del orden de  $O(h^4)$ , razón de peso por la cual es usado en los métodos computacionales.

### 3.3 Estructura de control

La estructura de control es el conjunto de elementos de un sistema, y sus parámetros, destinados a la determinación de la señal de control aplicada en el actuador. Ello con el fin de alcanzar un valor en la variable controlada lo más cercano posible a la referencia. La estructura control puede ser de lazo abierto o de lazo cerrado, según esté realimentada o no.

Si el controlador es automático, será él quien determine el valor de la señal de control en función del valor de la variable controlada en cada instante. Este es el caso del controlador PID y del controlador todo/nada con histéresis. Siendo ambos casos de controladores en lazo cerrado por estar realimentados.

Debe tenerse en cuenta que los objetivos de control vienen determinados por las especificaciones de diseño, las cuales pueden ser:

- Asegurar la estabilidad absoluta y mejorar la relativa.
- Lograr un buen seguimiento de la referencia.
- Rechazo de las perturbaciones.
- Mejora del transitorio.
- Eliminar las oscilaciones de la respuesta.
- Eliminar/Reducir el error estacionario.
- Reducir la energía necesaria, es decir, la magnitud de la señal de control.
- Un cierto grado de robustez frente a cambios en el punto de operación o proceso.

- Etc.

Un ejemplo de estructura de control en lazo cerrado puede observarse en el siguiente esquema (Figura 20).

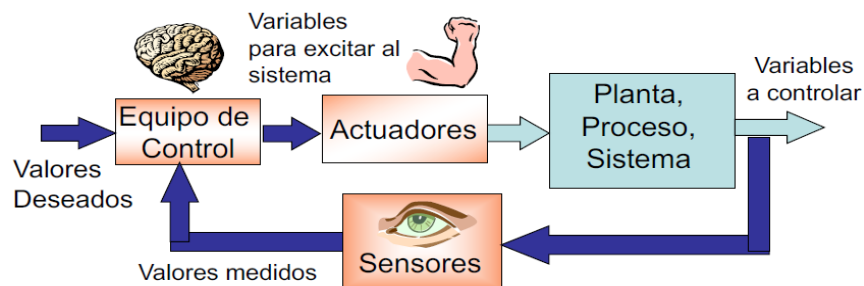


Figura 20 - Esquema de estructura de control en lazo cerrado

El equipo de control lee el valor de la referencia deseada y lo compara con el valor actual de la variable controlada. En función de su magnitud, calcula el valor de la señal de control. Los actuadores intervienen sobre el sistema en función de la señal recibida del equipo de control y los sensores determinan el valor de la variable controlada en cada instante.

En el laboratorio virtual a crear se plantean las tres siguientes tipologías de control:

### 3.3.1 Operación manual

En este escenario es el usuario quien realiza las veces de controlador seleccionando él mismo la señal de control a aplicar. Su uso puede deberse a la intención de conocer la situación de equilibrio de un cierto experimento o de conocer el comportamiento del sistema como paso previo a la sintonización de un controlador PID.

### 3.3.2 Controlador PID

Un controlador o regulador PID calcula la señal de control adecuada para que la salida alcance el valor de referencia deseado basándose en la señal de error, diferencia entre el valor de referencia y la variable controlada o de proceso. El controlador PID se compone de tres términos que proporcionan respectivamente una acción Proporcional, una acción Integral y una acción Derivativa. Debido a estas tres acciones es que el controlador PID recibe su

nombre. Un controlador PID puede encontrarse implementado en serie o en paralelo.

El controlador empleado en la simulación se encuentra implementado en paralelo por lo que es el que se estudiará (Figura 21).

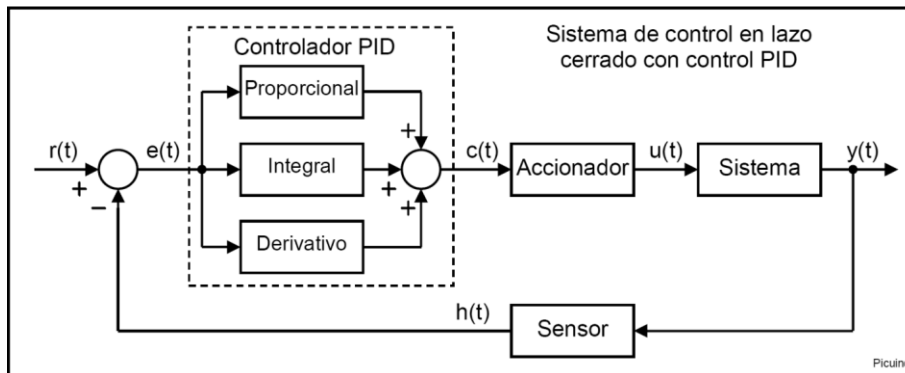


Figura 21 - Estructura de control con controlador PID paralelo

La señal de referencia recibe el nombre de  $r(t)$  y es el valor que se busca alcanzar en la salida del sistema  $y(t)$ .

Tanto la salida como la referencia tienen una evolución temporal, dependiendo la primera respecto de la segunda. En el sistema a estudio la referencia es la altura del depósito izquierdo deseada por el usuario, y la salida, el valor real de dicha altura.

No obstante, como se aprecia en el esquema de la Figura 20, no es la referencia, sino el error de la referencia  $e(t)$  la que entra al equipo de control. Esta señal es la diferencia entre la señal de referencia y la señal de salida del sistema. Indica la diferencia entre el estado buscado y el estado real. Un valor grande del error implica un sistema muy alejado de la referencia y por tanto la necesidad de una señal de control grande. En caso contrario, el sistema ha alcanzado el valor de referencia y debe conservarlo (el error estacionario aceptable depende de las especificaciones buscadas). Uno de los beneficios del controlador PID es no requerir conocer el proceso a regular de forma explícita. Sin embargo, no es adecuado para casos de dinámica difícil, retardos grandes, inestabilidad, respuesta inversa, mínima varianza, etc.

### **3.3.2.1 Acciones del controlador PID**

#### **Acción de control Proporcional**

Este término es proporcional a la señal de error  $e(t)$ , la cual es multiplicada por una constante  $K_p$ .

Esta acción de control trata de reducir el error estacionario. Cuando el error es elevado, la acción de control es grande, tratando de contrarrestarlo.

Un aumento de la acción de control proporcional mediante su constante  $K_p$  conlleva los siguientes efectos:

- Un aumento de la velocidad de respuesta.
- Una disminución del error en régimen permanente.
- Un incremento de la inestabilidad.

Los dos primeros son positivos y deseables. El último es negativo y es necesario mitigarlo en la medida de lo posible. Aumentando la acción proporcional se alcanza cierto punto de equilibrio en el cual se consigue la suficiente rapidez del sistema con una reducción del error suficiente, sin desestabilizar excesivamente el sistema. Seguir aumentando la acción proporcional conllevaría una desestabilización indeseable del sistema. Así mismo, reducir esta acción reduciría la rapidez del sistema y aumentaría su error.

#### **Acción de control Derivativa**

Esta acción de control es proporcional a la derivada del error  $e(t)$  multiplicada por la constante  $K_d$ . Cuando la variable controlada se desplaza rápidamente hacia el punto de referencia, esta se excederá debido a su inercia. Se produce un sobrepico y oscilaciones en torno a la referencia. Para evitar este fenómeno, es necesario que el controlador pueda conocer la velocidad a la que la variable controlada se aproxima a la referencia para poder disminuirla a medida que se aproxime. De esta manera, se evita que la sobrepase excesivamente y el correspondiente oscilamiento.

Un aumento en la constante de la acción derivativa  $K_d$  conllevaría los siguientes cambios:

- Un aumento de la estabilidad.

- Una ligera disminución de la velocidad.
- El error estacionario se mantendrá igual.

A modo de resumen, la acción derivativa permite reducir el oscilamiento de un sistema en torno a la referencia.

Esta señal de control presenta también algunos inconvenientes como es el hecho de amplificar las señales que varían rápidamente, como por ejemplo el ruido de alta frecuencia. Para contrarrestar este fenómeno es necesario aplicar un filtro de paso bajo a la señal de error antes de aplicar la acción de control derivativa. Esta medida reduce el efecto de la acción derivativa, por lo que es aconsejable tratar de reducir el ruido por otros medios antes de recurrir a un filtro. Una vez establecida la acción proporcional y la acción derivativa el sistema es ya rápido y estable, pero conserva un cierto error en régimen permanente. Para solucionar dicho desajuste, se incorpora al controlador la acción de control integral.

### **Acción de control Integral**

Tal y como su nombre indica, esta acción de control calcula la integral de la señal de error  $e(t)$  y la multiplica por la constante  $K_i$ . Esta puede interpretarse como la suma o acumulación de la señal de error. Progresivamente, pequeños errores se van acumulando, haciendo que la acción integral sea cada vez mayor, logrando reducir el error estacionario del sistema. La desventaja de la aplicación de este término radica en aumentar la inestabilidad del sistema, dotándole de una cierta inercia.

Un aumento en la constante de la acción integral  $K_i$  conlleva los siguientes efectos:

- Una disminución del error estacionario.
- Un aumento de la inestabilidad.
- Un ligero aumento de la velocidad.

La función principal de esta acción de control es reducir el error estacionario y habitualmente se combina con el control derivativo para evitar el oscilamiento del sistema.



### **3.3.2.2 Sintonización de un controlador PID**

Existen diferentes maneras de sintonizar un controlador PID:

- Métodos de prueba y error
- Métodos basados en experimentos: estimando ciertas características dinámicas del proceso con un experimento y calculando los parámetros del regulador mediante tablas o fórmulas.
- Métodos analíticos basados en modelos.
- Métodos de Sintonía automática.

Se va a tratar brevemente el proceso de sintonización manual de un controlador PID. El método basado en tablas será tratado más adelante junto con dos ejemplos prácticos referidos al proyecto.

### **3.3.2.3 Sintonización manual de un controlador PID**

Es una de las maneras más sencillas de sintonizar un controlador PID, aunque no la óptima. Se van ajustando progresivamente los parámetros del regulador mediante un proceso iterativo de prueba y error. Partiendo de valores bajos de  $k_p$ , sin acción integral o derivativa, se ajustan los parámetros en el siguiente orden:

#### **1. Acción proporcional.**

Progresivamente se aumenta  $K_p$  hasta obtener una forma de respuesta aceptable sin valores excesivos de  $u(t)$ .

Si se logra una respuesta aceptable para las especificaciones buscadas, no es necesario incluir acción derivativa o integral.

Si el sistema se desestabiliza antes de alcanzar la respuesta deseada, es necesario aumentar la acción de control derivativa.

#### **2. Acción derivativa.**

Se aumenta ligeramente  $T_d$  ( $K_d = K_p * T_d$ ) para mejorar la respuesta transitoria y en caso necesario estabilizar el sistema.

#### **3. Acción integral.**

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

Si se desea reducir el error estacionario, se comienza con un valor alto de  $T_i$  ( $K_i = K_p/T_i$ ) y se va disminuyendo hasta alcanzar el error estacionario buscado. Si el sistema se inestabiliza antes de lograr las especificaciones necesarias, se debe aumentar el término derivativo.

De esta manera, mediante un proceso iterativo se logra ajustar los 3 parámetros para las especificaciones de control buscadas.

La ecuación de un controlador PID ideal (Figura 22) es la siguiente:

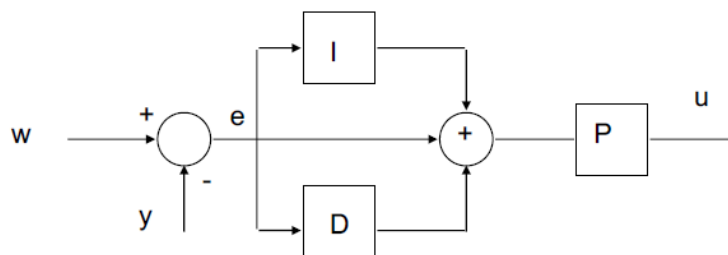


Figura 22 - Controlador PID ideal

$$u(t) = K_p \cdot (e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \cdot \frac{de(t)}{dt}) \quad (3.28)$$

Siendo:

- $u(t)$  = señal de control
- $e(t)$  = señal del error
- $K_p$  = constante proporcional
- $T_i$  = tiempo integral
- $T_d$  = tiempo derivativo

### 3.3.2.4 Efectos de saturación y límites del actuador:

Hay que tener en cuenta que es posible que se den efectos de saturación en el controlador debido al efecto de la acción de control integral. Esto implica que el término de la acción integral se haya hecho demasiado elevado debido a los errores estacionarios que se han ido sumando y que tras un cambio en la referencia el sistema no sea capaz de actuar por el elevado valor acumulado.

Otro fenómeno que se debe considerar son los límites del actuador. Es posible que la señal de control solicite al actuador una acción por encima de sus posibilidades. Sin importar el valor de la señal de control, la acción máxima que un actuador puede alcanzar es su 100%, y su mínimo el 0%, no pudiendo sobrepasarse dichos límites.

### 3.3.2.5 Controlador PID con filtro en el término derivativo

El primer tipo de controlador PID disponible es el controlador PID ideal con filtro en el término derivativo. Las ecuaciones que lo describen son las siguientes:

$$\left\{ \begin{array}{l} u(t) = K_p \cdot (e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de_f}{dt}) \\ \alpha T_d \frac{de_f}{dt} + e_f = e(t) \end{array} \right. \quad (3.29)$$

Dado que sus fórmulas (eq. (3.22)) se encuentran en el régimen temporal en el dominio continuo, es necesario discretizar.

Sustituyendo en las ecuaciones (3.22) la integral como integral de Riemann por el sumatorio equivalente y las derivadas como la variación por unidad de tiempo se obtiene

$$\left\{ \begin{array}{l} u(t) \approx K_p \cdot (e(t) + \frac{1}{T_i} \sum_{i=1}^t e(iT)T + T_d \frac{e_f(t) - e_f(t-T)}{T}) \\ \alpha T_d \frac{e_f(t) - e_f(t-T)}{T} + e_f \approx e(t) \end{array} \right. \quad (3.30)$$

Operando sobre la primera ecuación del sistema (3.23)

$$\begin{aligned} u(t-T) &\approx K_p \cdot (e(t-T) \\ &+ \frac{1}{T_i} \sum_{i=1}^{t-T} e(iT)T + T_d \frac{e_f(t-T) - e_f(t-2T)}{T}) \end{aligned} \quad (3.31)$$

$$u(t) - u(t - T) \approx K_p \cdot (e(t) - e(t - T)) + \frac{T}{T_i} e(t) + T_d \frac{e_f(t) - 2e_f(t - T) + e_f(t - 2T)}{T} \quad (3.32)$$

$$u(t) \approx u(t - T) + K_p \cdot \left(1 + \frac{T}{T_i}\right) e(t) - K_p \cdot e(t - T) + K_p \frac{T_d}{T} e_f(t) - 2 \frac{K_p \cdot T_d}{T} e_f(t - T) + \frac{K_p \cdot T_d}{T} e_f(t - 2T) \quad (3.33)$$

La ecuación (3.26) describe la señal de control que emite el controlador PID. Falta obtener la expresión de la señal del error filtrado. Operando sobre la segunda ecuación del sistema (3.32) se obtiene lo siguiente.

$$\frac{\alpha T_d}{T} e_f(t) + e_f(t) - \frac{\alpha T_d}{T} e_f(t - T) = e(t) \quad (3.34)$$

Haciendo factor común al error filtrado en la ecuación (3.27) queda.

$$\left(\frac{\alpha T_d}{T} + 1\right) e_f(t) = e(t) + \frac{\alpha T_d}{T} e_f(t - T) \quad (3.35)$$

Y finalmente despejando en (3.28) el error filtrado resulta.

$$e_f(t) = \frac{\frac{\alpha T_d}{T}}{\frac{\alpha T_d}{T} + 1} e_f(t - T) + \frac{1}{\frac{\alpha T_d}{T} + 1} e(t) \quad (3.36)$$

Ecuación que describe el valor del error filtrado. A modo de resumen, las dos ecuaciones que describen la señal de control del controlador PID con filtro en el término derivativo son las siguientes:

$$\left\{ \begin{array}{l} u(t) \approx u(t - T) + K_p \cdot \left(1 + \frac{T}{T_i}\right) e(t) - K_p \cdot e(t - T) + K_p \frac{T_d}{T} e_f(t) + \\ -2 \frac{K_p \cdot T_d}{T} e_f(t - T) + \frac{K_p \cdot T_d}{T} e_f(t - 2T) \quad \textcircled{A} \\ e_f(t) = \frac{\frac{\alpha T_d}{T}}{\frac{\alpha T_d}{T} + 1} e_f(t - T) + \frac{1}{\frac{\alpha T_d}{T} + 1} e(t) \quad \textcircled{B} \end{array} \right. \quad (3.37)$$

A continuación, se encuentra un esquema que describe las relaciones entre las señales y las ecuaciones del controlador PID (Figura 23).

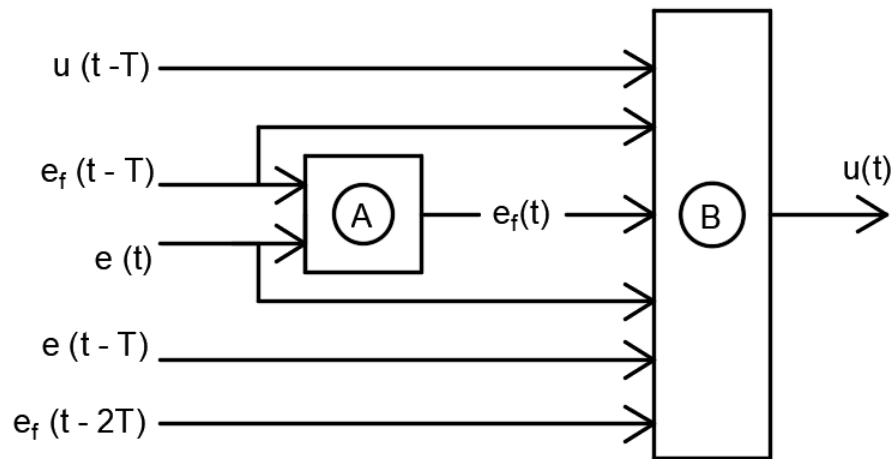


Figura 23 - Esquema de señales al controlador PID ideal con filtro

### 3.3.2.6 Controlador PID de JavaRegula.

La segunda modalidad de controlador PID disponible para realizar el control de la simulación es el controlador PID empleado por el software JavaRegula, el cual posee un filtro de paso bajo (Santos, 2020). La cual viene definida en el manual de usuario de dicho software y que emplea las siguientes ecuaciones:

La señal de control se calcula de la siguiente manera:

$$K = K_p \cdot \text{Valor máximo del control (en \%)} \div \text{Span de la medida}$$

de esta expresión resulta la constante proporcional (K) (3.38)

del PID

$$u(t) - u(t - 1) = K \cdot (e(t) - e(t - 1)) + \dots \quad (3.39)$$

JavaRegula calcula el término integral según la siguiente fórmula, siendo  $T_i$  el tiempo integral expresado en minutos.

$$u(t) - u(t - 1) = \dots + \frac{K \cdot T \cdot e(t - 1)}{T_i \cdot 60} + \dots \quad (3.40)$$

JavaRegula calcula el término derivativo no como la derivada del error, sino a través de las diferencias entre la derivada de la salida y la derivada de la salida filtrada con un filtro de primer orden. Siendo  $T_d$  el tiempo derivativo expresado en minutos.

$$u(t) - u(t - 1) = \dots + \frac{K \cdot T_d \cdot 60 \cdot \alpha \cdot (\dot{y}(t) - \dot{y}_f(t))}{T} \quad (3.41)$$

Siendo:

$$\text{con } \alpha = 0.27$$

$$\dot{y}(t) = y(t) - y(t-1) \quad (3.42)$$

$$\dot{y}_f(t) = y_f(t) - y_f(t-1) \quad (3.43)$$

$$y_f(t) = y_f(t-1) + \alpha \cdot (y(t-1) - y_f(t-1)) \quad (3.44)$$

Estas ecuaciones describen completamente el cálculo de la señal de control para el caso del controlador empleado en el software JavaRegula.

### 3.3.3 Controlador Todo/Nada con histéresis

Se trata de un tipo de controlador de sencilla implementación en el cual la señal de control puede alternar solamente entre dos valores, sin poder alcanzar valores intermedios. Un controlador tipo Todo/Nada estándar emite una señal

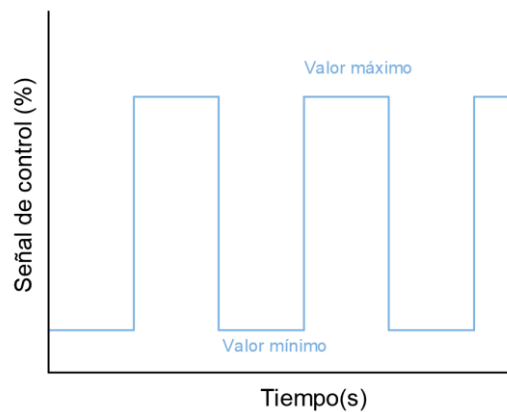


Figura 24 - Señal de control de un controlador Todo/Nada

de control máxima cuando la variable controlada se encuentra por debajo de la referencia establecida. Por el contrario, alterna a su valor mínimo cuando se encuentra por encima. Puede observarse la evolución de la señal de control (Figura 24).

Cabe destacar que los valores entre los que alterne la señal de control no tienen por qué ser necesariamente el 0% y el 100%. Como consecuencia de este algoritmo, se produce una oscilación elevada en la señal de control (Figura 26).

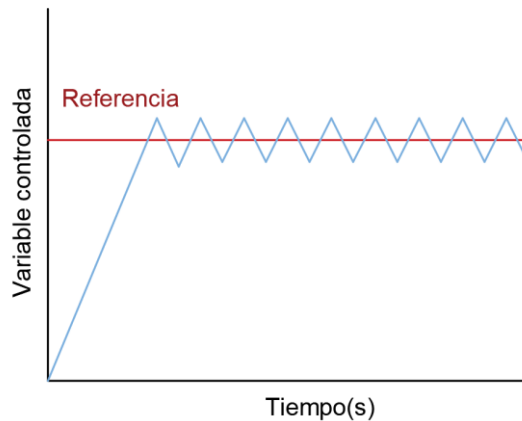


Figura 26 - Variable controlada en Control Todo/Nada sin histéresis

Este comportamiento tan oscilante puede no ser recomendable para el modelo físico en el cual se va a implantar el algoritmo de control o puede provocar ineficiencias o inconvenientes en el proceso controlado. Es por ello que se incorpora a la estructura de control una histéresis. Esto se traduce en que se añade una banda alrededor de la referencia definida su anchura (ancho de banda). En vez de que la señal de control oscile entre los valores máximos y mínimos cada vez que la variable de control cruce la referencia, adquirirá su valor máximo cuando supere el límite inferior de la banda y su valor mínimo cuando supere el superior (Figura 25).

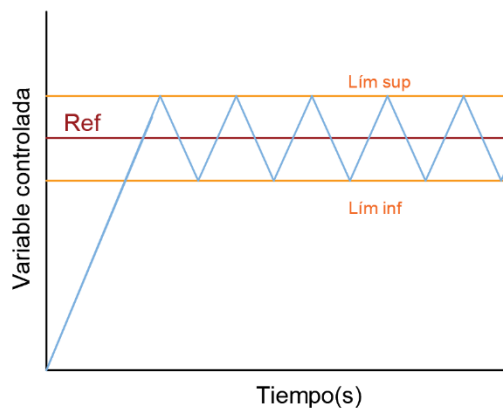


Figura 25 - Variable controlada en Control Todo/Nada con histéresis

Puede apreciarse como la presencia de la banda evita el exceso de conmutaciones en un corto periodo de tiempo. Si el sistema se encontrase cerca de la referencia y hubiera ruido de medida, la variable controlada estaría oscilando continuamente en torno a la referencia y el controlador conmutaría con elevada frecuencia. Lo cual afectaría de forma negativa al actuador.



El control Todo/Nada con histéresis presenta ciertas desventajas al proporcionar por lo general una respuesta pobre y no alcanzar un valor estacionario constante.



# CAPÍTULO 4: Desarrollo del laboratorio virtual

---

El código fuente del laboratorio virtual puede encontrarse en los anejos. Es por ello por lo que solo se comentarán aquellos aspectos de su desarrollo que posean especial interés técnico.

## 4.1 Interfaz gráfica

Para la creación de la interfaz gráfica, nos dirigimos a la pestaña “HtmlView”. Se crea una nueva página y dentro de ella dos paneles (Figura 27).

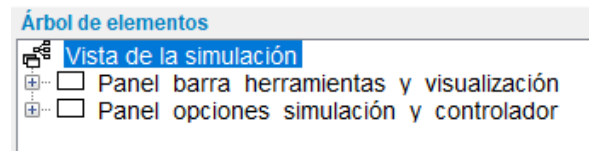


Figura 27 – Paneles principales de la interfaz

El primer panel alojará la barra de herramientas, la ilustración representativa de la simulación y las gráficas. El segundo panel alojará las opciones de la simulación y los ajustes de los parámetros de los controladores.

### 4.1.1 Barra de herramientas

Su función es la de permitir abrir la página de descripción, seleccionar el tipo de control a realizar, guardar los datos de la simulación o bien abrir la página de los créditos (Figura 28).



Figura 28 - Barra de herramientas

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

Para su creación, dentro del panel de la barra de herramientas y visualización se crea otro panel que la va a alojar llamado “Panel\_barra\_herramientas” (Figura 29).

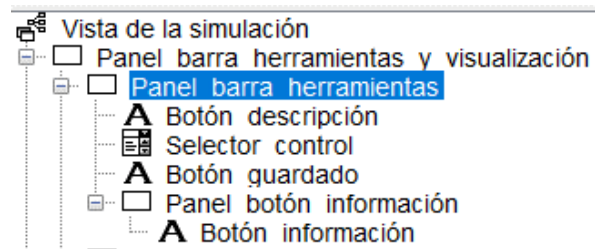


Figura 29 - Estructura barra de herramientas

En su interior, se crea una etiqueta de texto que realizará las veces de botón de descripción, un selector de control que permitirá seleccionar la tipología de control, otra etiqueta que cumplirá las funciones de botón de guardado y un panel. Dentro de este último panel, se encontrará la etiqueta que actuará como botón de créditos.

Se define el código a ejecutar en el caso de que el usuario realice un clic sobre la etiqueta “Botón\_descripción” mediante el evento “OnClick”. El cual abre la página de descripción como una ventana emergente.

```
window.open("../Descripción.html", "_blank");
```

Para el caso de la etiqueta “Botón\_guardado”, el código es el siguiente:

```
%guardar%
```

El cual ejecuta la función “Guardar” definida en el apartado “Modelo”.

En cuanto al menú desplegable, toma las opciones disponibles del array “opccontrol” y almacena la opción marcada en la variable “elcontrol”.

Adicionalmente, se especifica el código que se debe ejecutar al cambiar de opción, el cual es el siguiente:

```
if(controlSelection=="Operación manual"){
control = 1;
dispopManual = "inline-block";
dispControlador = "none";
dispTodoNada = "none";
}

else if(controlSelection=="Control PID"){
control = 2;
dispopManual = "none";
dispControlador = "inline-block";
dispTodoNada = "none";
}

else if(controlSelection=="Control On/Off"){
control = 3;
dispopManual = "none";
dispControlador = "none";
dispTodoNada= "inline-block";
    if (u>valON){
        u=valON;
    }else {
        u=valOFF;
    }
}
_resetSolvers;
```

Este código almacena en la variable “control” el tipo de controlador a utilizar mediante un valor numérico. Después, hace visible el panel con la configuración del controlador a utilizar e invisibles los demás cambiando el valor de la propiedad CSS “display”. Adicionalmente, se resetea el método numérico para evitar conflictos. En caso de cambiar al control Todo/Nada con histéresis, se cambia el valor de la señal de control al valor máximo o mínimo según cual sea más cercano al actual.

La etiqueta “Botón\_información” se encuentra en el interior de un panel para poder colocarla en el extremo derecho gracias a aplicar propiedades de bloque.

La etiqueta alojada en su interior posee la imagen de los créditos gracias a la propiedad:

- imageUrl: "./fotos/info.png" (establece el directorio en el que se encuentra la foto a usar).

Adicionalmente, ante el evento "OnClick" se ejecuta el siguiente código:

```
window.open("./Créditos.html", "_blank");
```

El cual abre la página de créditos en una ventana emergente.

#### 4.1.2 Ilustración de la simulación

También en el interior del panel llamado "Panel\_barra\_herramientas\_y\_visualización", se crea otro panel denominado "Panel visualización". Este panel, situado debajo de la barra de herramientas, contendrá tanto la ilustración de la simulación como las gráficas (Figura 30).

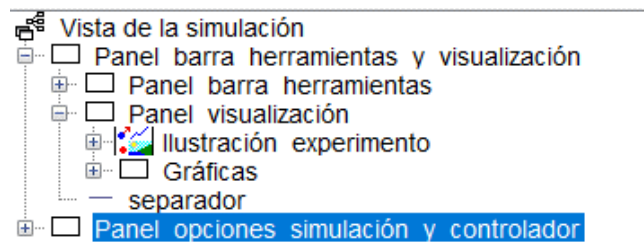


Figura 30 - Estructura sinóptico e ilustración

La ilustración de la simulación se crea mediante el elemento PanelDibuo creado con el nombre "Ilustración\_experimento". En su interior, se dibuja una representación del modelo mediante la adición de diversos elementos (Figura 31) colocados en respectivas coordenadas.

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

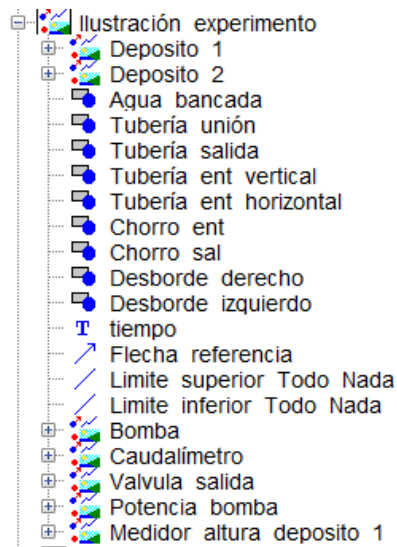


Figura 31 - Elementos de dibujo empleados en el sinóptico

Los depósitos se dibujan mediante un elemento de dibujo 2D ya definido cuya representación es un tanque. Adicionalmente, se incluye un rectángulo gris que representará el fondo del mismo y una etiqueta de texto especificando si se trata del depósito 1 o 2. Estos tres elementos se agrupan dentro de un grupo de dibujo con el objetivo de que estos se desplacen de forma solidaria (Figura 32).

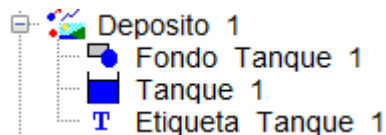


Figura 32 - Grupo de dibujo del depósito 1

El nivel de agua representado en el depósito se relaciona con la variable en la cual se almacena el valor de la altura obtenido mediante las ecuaciones diferenciales.

Mediante un rectángulo de color azul situado en la parte inferior se representa el agua de la bandeja. Mediante rectángulos también azules se representan los caudales de entrada y salida y el agua que desborda, siendo este únicamente visibles en el caso de producirse el desbordamiento.

## *Laboratorio virtual de una planta piloto de dos tanques comunicantes*

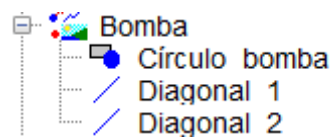
Los caudales de entrada y salida se dibujan con un tamaño proporcional a la relación existen entre su valor y su valor máximo por medio de una simple regla de tres.

Los rectángulos que representan el desbordamiento igualmente poseen un tamaño proporcional a la cantidad de agua desbordada. Este se calcula sin más que realizar una regla de tres al caudal que desborda. Considerando el caudal que desborda como el caudal de entrada menos el caudal que atraviesa la tubería.

En la esquina inferior izquierda aparece el Tiempo de simulación.

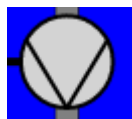
En el caso de estar realizando una tipología de control automática, se representa en el depósito izquierdo la referencia a alcanzar mediante una flecha. Si el control es Todo/Nada con histéresis, aparecen dos segmentos que representan el límite superior e inferior de la banda de histéresis.

El símbolo de la bomba se representa mediante un círculo con fondo gris y dos segmentos diagonales (Figura 33).



*Figura 33 – Estructura de la representación de la bomba*

Siendo el resultado final el siguiente (Figura 34):

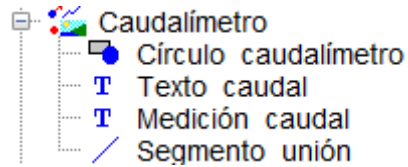


*Figura 34 – Aspecto visual de la bomba*



*Laboratorio virtual de una planta piloto de dos tanques comunicantes*

Los cuatro instrumentos del montaje: Caudalímetro, Válvula de Salida, Indicador de potencia de la bomba y medidor de la altura del depósito izquierdo se modelan de la misma manera (Figura 35).



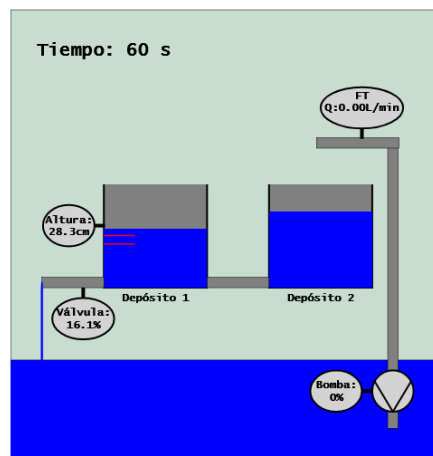
*Figura 35 - Estructura de instrumento*

Se utiliza un círculo con fondo gris para definir el cuerpo del instrumento, una etiqueta de texto que indica el instrumento del que se trata, otra etiqueta que muestra el valor medido y un segmento que une el instrumento de medida con el montaje. El resultado es el siguiente (Figura 36):



*Figura 36 - Aspecto visual del caudalímetro*

El resultado global logra representar tanto el montaje realizado como los instrumentos de medida y sus valores percibidos (Figura 37).



*Figura 37 - Aspecto visual del montaje*

### 4.1.3 Gráficas

Dentro también del panel llamado “Panel\_visualización” se encuentra otro panel con el nombre de “Gráficas”, el cual las aloja. En su interior, se encuentra otro panel llamado “Leyenda” con las etiquetas que cumplen dicha función. En la misma categoría que el panel “Leyenda” se encuentran la gráfica de la altura del depósito izquierdo y la gráfica del caudal de entrada (Figura 38).

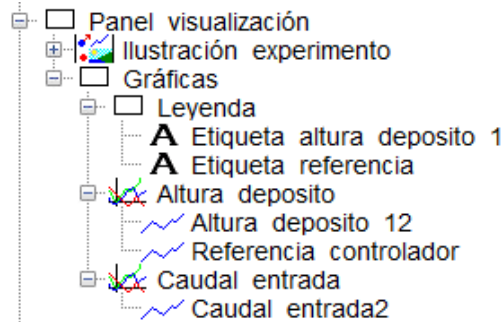


Figura 38 - Estructura panel de gráficas

Dentro del panel “Leyenda” se encuentran “Etiqueta\_altura\_deposito 1” y “Etiqueta\_referencia”. Ambas etiquetas componen la leyenda de la primera gráfica, la cual representa la altura del depósito izquierdo y la referencia deseada (cm) respecto del tiempo.

El objetivo de la leyenda es expresar a qué variable hace referencia cada trazo (Figura 39).



Figura 39 - Leyenda de la gráfica “Altura depósito”

A continuación, debajo de la leyenda, se encuentran las gráficas. Se trata de dos elementos tipo “Paneles con ejes”.

Dentro de la gráfica “Altura\_deposito” se representan dos trazos. El primero se llama “Altura\_deposito\_12” y representa la altura del depósito izquierdo. El segundo se llama “Referencia\_controlador” y representa la referencia. Dentro de la segunda gráfica se representa el trazo “Caudal\_entrada2”, el cual representa el caudal de entrada.

Los tres trazos poseen las siguientes propiedades en común:

- Maximum: npuntos (es el número máximo de puntos que se pueden representar en la gráfica. Una vez se alcanza el máximo los puntos más antiguos se pierden para alojar a los nuevos. Como valor se le ha dado la variable “npuntos”. El valor de la variable “npuntos” se determina en la sección “Modelo”. Este valor depende de la franja temporal visible y del periodo de muestreo.
- NoRepeat:true (omite la representación de puntos iguales).

Estas dos propiedades se establecen con el objetivo de mejorar el rendimiento del laboratorio virtual. Adicionalmente, se establece el color de cada trazo, siendo rojo para la referencia y azul tanto para la altura del depósito izquierdo como para el caudal de entrada al sistema.

Los valores de entrada del trazo “Altura\_deposito\_1” son:

- InputX: t (el tiempo).
- InputY:  $h1f*100$  (la altura del depósito izquierdo teniendo en cuenta el error de medida en metros, multiplicado por 100 para convertirlo a cm).

Los valores de entrada del trazo “Referencia\_controlador” son:

- InputX: refx (establecido en la pestaña Modelo).
- InputY: refy (establecido en la pestaña Modelo).

Los valores de entrada del trazo “Caudal\_entrada2” son:

- InputX: t (el tiempo).
- InputY:  $Qent*60*1000$  (el caudal de entrada al sistema en  $m^3/s$  multiplicado por un factor de conversión para pasarlo a L/min).

El aspecto de las gráficas es el siguiente (Figura 40):

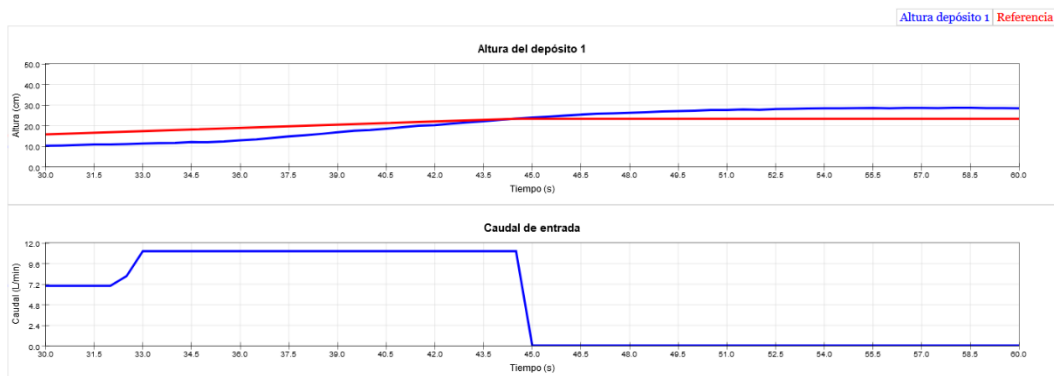


Figura 40 – Aspecto visual gráficas

Dentro del panel “Panel\_barra\_herramientas\_y\_visualización” se ha añadido también un separador, el cual aparece debajo de la representación del montaje y las gráficas.

#### 4.1.4 Configuración Simulación

Dentro del Panel llamado “Panel\_opciones\_simulación\_y\_controlador” se encuentran tanto los ajustes referidos a la configuración de la simulación como los ajustes de los controladores.

En su interior, se encuentran cuatro paneles los cuales alojan la configuración de la simulación, las opciones de la operación manual, las opciones del controlador PID y las opciones del controlador Todo/Nada con histéresis (Figura 41).

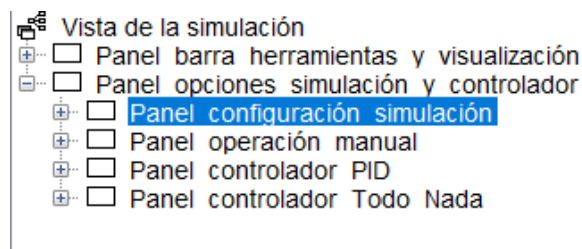


Figura 41 - Estructura panel de opciones

Dentro del panel “Panel\_configuración\_simulación”, se coloca un panel que aloja cada fila de opciones (Figura 42). Esto se realiza debido a que las etiquetas y los cuadros de texto son elementos de línea, y los paneles elementos de bloque. De esta manera se logra que en la interfaz aparezcan en filas diferentes gracias a las propiedades de bloque.

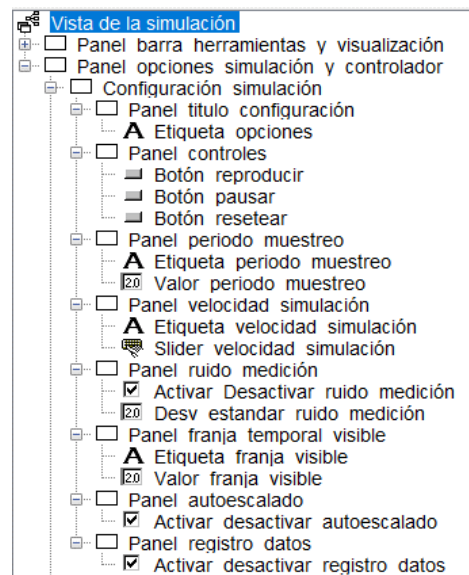


Figura 42 - Estructura panel de configuración de la simulación

El panel llamado “Panel\_titulo\_configuración” aloja la etiqueta que realiza las veces de título con el texto “OPCIONES”.

El panel llamado “Panel\_controles” aloja los botones que regulan la ejecución de la simulación. Son tres:

- Botón\_reproducir reanuda la simulación. Muestra tanto el texto como un icono. El código que se ejecuta tras el evento OnClick es el siguiente:

```
playColor=selecColor;  
pauseColor=unSelecColor;  
_play();
```

Dicho código además de reanudar la simulación cambia el color de los botones reanudar y pausar para que se aprecie el estado de la simulación.

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

- Botón\_pausar detiene la simulación. Muestra tanto el texto como un icono. El código que se ejecuta tras el evento OnClick es el siguiente:

```
pauseColor=selecColor;  
playColor=unSelecColor;  
_pause();
```

Dicho código pausa la simulación y provoca un comportamiento en el color de los botones análogo al producido por el botón Reanudar.

- Botón\_resetear reinicia el laboratorio virtual. Muestra tanto el texto como un icono. El código que se ejecuta tras el evento OnClick es el siguiente:

```
_reset();
```

Dado que el laboratorio se reinicia no es necesario modificar el color de fondo de los botones.

El panel “Panel\_periodo\_muestreo” aloja un cuadrado de texto en el que especificar el periodo deseado que se almacena en la variable “T”.

El panel “Panel\_velocidad\_simulación” aloja una slider y un cuadro de texto con los cuales modificar la velocidad a la que se ejecuta la simulación, almacenada en la variable “velt”.

El panel “Panel\_ruido\_medición” contiene una casilla de selección con la cual activar o desactivar el ruido de medición con la variable “ruido” y un cuadro de texto mediante el cual modificar su desviación estándar, con la variable “des\_est”.

El panel “Panel\_franja\_temporal\_visible” permite modificar mediante un cuadro de texto la variable “tvisible”, que especifica la cantidad de tiempo visible en las gráficas.

El panel “Panel\_autoescalado” permite activar o desactivar el autoescalado de las gráficas en el eje de ordenadas mediante una casilla de selección que modifica el valor de la variable “autoescalado”.

El panel “Panel\_registro\_Datos” contiene una casilla de selección mediante la cual modificar el valor de la variable booleana “registro”, que determina si los datos se almacenan o no.

#### 4.1.5 Opciones Operación Manual

El panel “Panel\_operación\_manual” contiene los elementos gráficos que permiten configurar el control manual (Figura 43Figura 43 - Estructura panel operación manual).

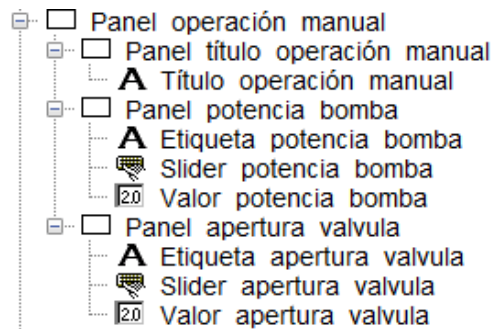


Figura 43 - Estructura panel operación manual

El panel llamado “Panel\_titulo\_operación\_manual” aloja la etiqueta que realiza las veces de título con el texto “CONTROL MANUAL”.

El panel llamado “Panel\_potencia\_bomba” permite modificar el valor de la señal de control almacenado en la variable “u” por medio de una barra deslizante y un cuadro de texto.

El panel llamado “Panel\_apertura\_valvula” permite modificar el valor de la apertura de la valvula almacenada en la variable “pSsal” por medio de una barra deslizante y un cuadro de texto.

El aspecto logrado es el siguiente (Figura 44).

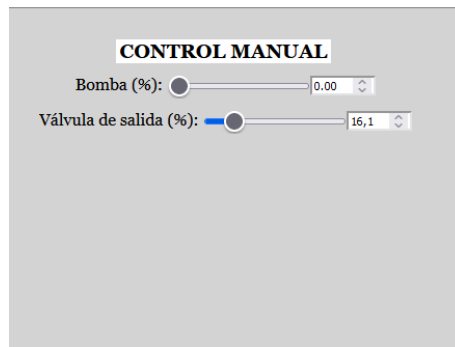


Figura 44 - Interfaz operación manual

#### 4.1.6 Opciones Controlador PID

El panel "Panel\_controlador\_PID" contiene los elementos gráficos que permiten ajustar los parámetros del controlador PID (Figura 45).

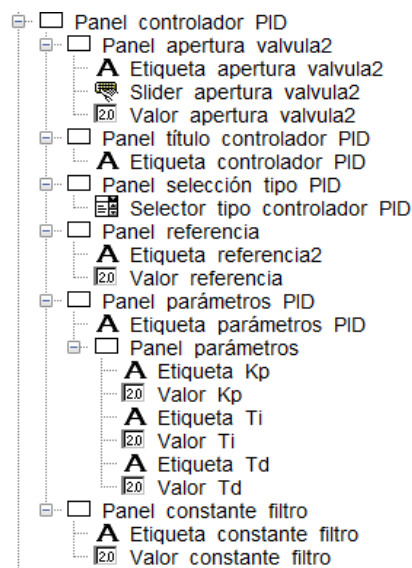


Figura 45 - Estructura panel controlador PID

El panel llamado "Panel\_apertura\_valvula" permite modificar el valor de la apertura de la válvula almacenada en la variable "pSsal" por medio de una barra deslizante y un cuadro de texto.



El panel llamado “Panel\_titulo\_controlador\_PID” aloja la etiqueta que realiza las veces de título con el texto “CONTROLADOR PID”.

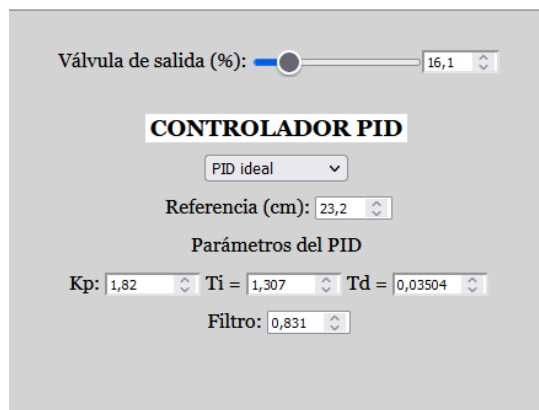
El panel llamado “Panel\_selección\_tipo\_PID” permite modificar el tipo de PID a utilizar por medio de menú desplegable que muestra las opciones almacenadas en el array “opcPID” y almacena la selección en “elPID”. Adicionalmente, modifica el valor de la variable “control” para que guarde 2.1 si el control es “Ideal” o 2.2 si es la versión de “JavaRegula”.

El panel llamado “Panel\_referencia” aloja un cuadro de texto mediante el cual modificar el valor de la referencia almacenada en “ref”.

El panel llamado “Panel\_parametros” aloja diversos cuadros de textos mediante los cuales modificar los valores de la constante proporcional, el tiempo integral y el tiempo derivativo almacenados en “kp”, “ti” y “td” respectivamente.

El panel llamado “Panel\_constante\_filtro” aloja un cuadro de texto mediante el cual modificar el valor del filtro del término derivativo almacenado en “alfa”.

El aspecto final es el siguiente (Figura 46).



*Figura 46 - Interfaz controlador PID*

### 4.1.7 Opciones Controlador Todo/Nada

El panel “Panel\_controlador\_Todo\_nada” contiene los elementos gráficos que permiten ajustar los parámetros del controlador Todo/Nada con histéresis (Figura 47).

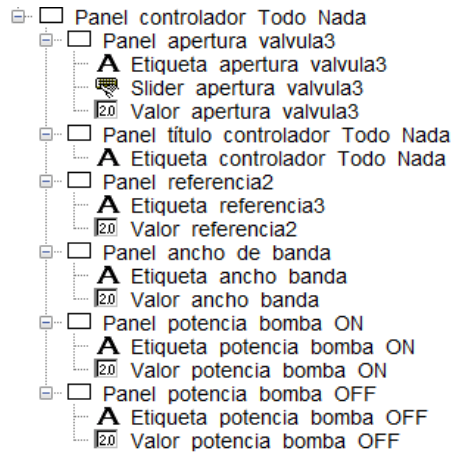


Figura 47 - Estructura panel controlador Todo/Nada

El panel llamado “Panel\_apertura\_valvula” permite modificar el valor de la apertura de la valvula almacenada en la variable “pSsal” por medio de una barra deslizante y un cuadro de texto.

El panel llamado “Panel\_titulo\_controlador\_Todo\_Nada” aloja la etiqueta que realiza las veces de título con el texto “CONTROLADOR TODO-NADA CON HISTÉRESIS”.

El panel llamado “Panel\_referencia” aloja un cuadro de texto mediante el cual modificar el valor de la referencia almacenada en “ref”.

El panel llamado “Panel\_ancho\_de\_banda” aloja un cuadro de texto mediante el cual modificar el valor del ancho de banda almacenado en “anchob”.

El panel llamado “Panel\_potencia\_bomba\_ON” aloja un cuadro de texto mediante el cual modificar el valor de la potencia máxima almacenada en “valON”.

El panel llamado “Panel\_potencia\_bomba\_OFF” aloja un cuadro de texto mediante el cual modificar el valor de potencia mínima almacenada en “valOFF”.

El resultado final es el siguiente (Figura 48):

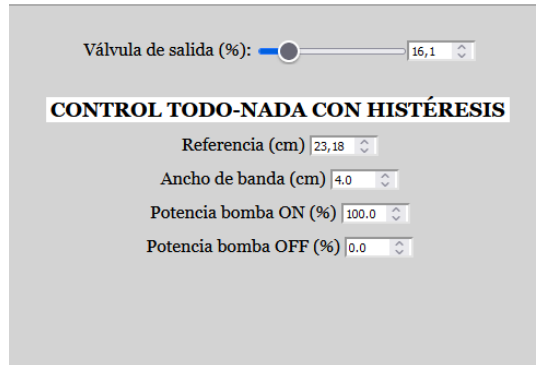


Figura 48 - Interfaz controlador Todo/Nada

## 4.2 Modelo

Se trata a continuación el desarrollo de la lógica interna del laboratorio virtual.

### 4.2.1 Variables

Las variables se encuentran clasificadas en pestañas según la función que desempeñan. Todas las variables utilizadas para indicar el estado actual de la simulación se encuentran expresadas en unidades del Sistema Internacional.

Las variables de la pestaña “Modelo” son:

NOMBRE	TIPO	VALOR ALMACENADO
h1	double	Altura del depósito 1 (m)
h2	double	Altura del depósito 2 (m)
h1f	double	Altura del depósito 1 con ruido de medida (cm)

*Laboratorio virtual de una planta piloto de dos tanques comunicantes*

t	double	Tiempo de simulación (s)
T	double	Periodo de muestreo
control	int	Tipo de control a aplicar
desbord	boolean	Si el sistema desborda o no
pSsal	double	Apertura de la válvula de salida (%)
Qent	double	Caudal de entrada (m3/s)
Qtub	double	Caudal que atraviesa la tubería (m3/s)
Qsal	double	Caudal de salida (m3/s)
velt	double	Velocidad de simulación
delay	double	Tiempo entre iteraciones de la simulación

Las variables de la pestaña “Interfaz” son:

NOMBRE	TIPO	VALOR ALMACENADO
opcontrol	String	Opciones disponibles para el menú de tipo de control

*Laboratorio virtual de una planta piloto de dos tanques comunicantes*

elcontrol	String	Selección del menú desplegable de tipos de control
selecColor	String	Color usado para el botón seleccionado
unSelecColor	String	Color usado para el botón no seleccionado
dispopManual	String	Valor de la etiqueta de estilo CSS display para el panel de Operación manual
dispControlador	String	Valor de la etiqueta de estilo CSS display para el panel de Controlador PID
dispTodoNada	String	Valor de la etiqueta de estilo CSS display para el panel de Controlador Todo/Nada con histéresis
playColor	String	Color del botón Play
pauseColor	String	Color del botón Pausar
tmin	double	Tiempo mínimo visible en las gráficas
tmax	double	Tiempo máximo visible en las gráficas
refx	double	Valor de altura del depósito de la referencia
refy	double	Valor de tiempo de la referencia

*Laboratorio virtual de una planta piloto de dos tanques comunicantes*

opcPID	String	Opciones de controladores PID
eIPID	String	Opción de controlador PID marcada
tvisible	double	Cantidad de segundos visibles en las gráficas
npuntos	double	Número máximo de puntos representados en las gráficas
ymaxalt	double	Valor máximo eje Y gráfica altura depósito
ymaxcaudal	double	Valor máximo eje Y gráfica caudal entrada

Las variables de la pestaña “Control” son:

NOMBRE	TIPO	VALOR ALMACENADO
ref	double	Referencia del controlador (cm)
kp	double	Constante proporcional PID (cm/%)
td	double	Tiempo derivativo (min)
ti	double	Tiempo integral (min)
alfa	double	Constante del filtro del término derivativo

*Laboratorio virtual de una planta piloto de dos tanques comunicantes*

anchob	double	Ancho de banda del control Todo/Nada (cm)
valON	double	Valor máximo control Todo/Nada (cm)
valOFF	double	Valor mínimo control Todo/Nada (cm)
e0	double	Error actual (cm)
e1	double	Error periodo anterior (cm)
e2	double	Error 2 periodos anteriores (cm)
ef	double	Error filtrado (cm)
ef1	double	Error filtrado anterior (cm)
ef2	double	Error filtrado 2 periodos anteriores (cm)
g0	double	Constante 1 PID JavaRegula
g1	double	Constante 2 PID JavaRegula
g2	double	Constante 3 PID JavaRegula
u	double	Señal de control (%)

*Laboratorio virtual de una planta piloto de dos tanques comunicantes*

u1	double	Señal de control anterior (%)
_y	double	Variación salida (cm)
_yf	double	Variación salida filtrada (cm)
y1	double	Salida actual (cm)
yf	double	Salida filtrada (cm)
yf1	double	Salida filtrada anterior (cm)

Las variables de la pestaña “Dimensiones” son:

NOMBRE	TIPO	VALOR ALMACENADO
Qentmax	double	Caudal de entrada máximo (m <sup>3</sup> /s)
Ssalmax	double	Sección máxima de la válvula de salida (m <sup>2</sup> )
Ssal	double	Sección de la válvula de salida (m <sup>2</sup> )
Stub	double	Sección de la tubería (m <sup>2</sup> )
A1	double	Área del depósito 1 (m <sup>2</sup> )



*Laboratorio virtual de una planta piloto de dos tanques comunicantes*

A2	double	Área del depósito 2 (m2)
----	--------	--------------------------

Las variables de la pestaña “Distribución normal” son:

NOMBRE	TIPO	VALOR ALMACENADO
disu	double	Variable auxiliar en el cálculo de la distribución normal del ruido
disv	double	Variable auxiliar en el cálculo de la distribución normal del ruido
disz	double	Variable auxiliar en el cálculo de la distribución normal del ruido
media	double	Media del ruido
des_est	double	Desviación estándar del ruido
ruido	boolean	Si debe tenerse en cuenta el ruido

Las variables de la pestaña “Registro Datos” son:

NOMBRE	TIPO	VALOR ALMACENADO
arraydatos	String	Array en el que se almacenan los datos

registro	boolean	Si se deben registrar los datos
espacio	String	Variable auxiliar que almacena 8 espacios
fecha	String	Fecha de realización de la simulación

## 4.2.2 Inicialización

Existen dos páginas de código de inicialización. La primera se llama “Velocidad\_simulación” y establece el ritmo de ejecución.

```
_model.setRunningMode("MODE_RESPECT_TIME");  
_setDelay(500);
```

Con estas instrucciones se especifica que el tiempo entre iteraciones del método numérico va a depender del tiempo real y dado que el periodo inicial es de 500ms, se especifica como tiempo entre iteraciones 500ms. De esta manera el tiempo de simulación evoluciona al mismo ritmo que el tiempo real.



## Laboratorio virtual de una planta piloto de dos tanques comunicantes

Var. Indep.	Incremento	Prelim
t		
Estado	Derivada	
$\frac{dh1}{dt} =$	$(Qtub - Qsa1) / A1$	
$\frac{dh2}{dt} =$	$(Qent - Qtub) / A2$	

Método Runge-Kutta 4 Tol 0.0001 Eventos 0

Figura 49 - Ecuaciones diferenciales ordinarias

En la página “Control” se encuentra el código necesario para realizar la simulación.

Primeramente, se comprueba que las alturas de los depósitos no sean negativas y en caso contrario se hacen nulas. Este suceso es físicamente imposible y se debería a que en alturas cercanas a cero durante algunas iteraciones el método numérico calcule con demasiado error. Este paso es necesario también dado que las ecuaciones que describen el modelo poseen una raíz cuadrada de la altura de cada depósito. Si la altura es negativa es imposible calcular la raíz cuadrada para el método numérico y este cae.

```
if (h1 < 0) {  
    h1 = 0;  
}  
  
if (h2 < 0) {  
    h2 = 0;  
}
```

A continuación, se calcula el caudal que atraviesa la tubería y el caudal de salida del depósito izquierdo.

```
Qtub = Stub * (Math.sqrt(2 * 9.8 * h2) - Math.sqrt(2 * 9.8 * h1));  
Qsa1 = Ssa1 * Math.sqrt(2 * 9.8 * h1);
```

En caso de que el ruido esté activado, se debe añadir al valor real de la altura del depósito izquierdo el resultado de un número aleatorio que sigue una

distribución normal, representado el error de medida. Este valor es el que mostrará como lectura el medidor de altura.

```
if(ruido){  
  
    h1f=h1+disNormal();  
}else {  
  
    h1f=h1;  
}
```

JavaScript no es capaz de generar números aleatorios de forma nativa por lo que se emplea una función propia llamada “disNormal” que se explicará al llegar a la pestaña “Propio”.

Se calcula el valor del error, la señal filtrada y la sección de la válvula marcada por el usuario.

```
e0=ref-h1f*100;  
yf=yf1+alfa*(y1-yf1);  
Ssal=Ssalmax*pSsal/100;
```

Si la variable control tiene valor 1, el control es manual y se ejecuta el siguiente algoritmo:

```
if (control == 1) {  
    ref=h1*100;  
}
```

No es necesario calcular la señal de control y únicamente se cambia el valor de la referencia al valor actual de la altura del depósito izquierdo. Esto se realiza para que, en caso de cambiar el tipo de control a automático, ya sea PID o Todo/Nada, el sistema trate de mantener la altura existente en dicho momento.

### Laboratorio virtual de una planta piloto de dos tanques comunicantes

Si la variable control tiene valor 2.1, el control se realiza mediante un controlador PID ideal con filtro en el término derivativo y se ejecuta el siguiente algoritmo:

```
if (control == 2.1) {  
  
    ef=(alfa*td) / (alfa*td+(T/60)) *ef1+e0*( (T/60) / (alfa  
    *td+T/60));  
    u=u1+kp*(1+T/(ti*60))*e0-kp*e1+kp*ef*(60*td/T) -  
    2*ef1*kp*td*60/T+kp*td*60*ef2/T;  
}
```

Se calcula el error filtrado y el valor de la señal de control.

Si la variable control tiene valor 2.2, el control se realiza mediante un controlador PID con la versión de JavaRegula y se ejecuta el siguiente algoritmo:

```
if(control == 2.2){  
    g0=kp;  
    g1=(kp*T)/(ti*60)-kp;  
    g2=(kp*td*60*alfa/T);  
    _y=(h1f-y1)*100;  
    _yf=(yf-yf1)*100;  
    u=u1+g0*e0+g1*e1-g2*(_y-_yf);  
}
```

Si la variable control tiene valor 3, el control se realiza mediante un controlador Todo/Nada con histéresis y se ejecuta el siguiente algoritmo:

```
if(control==3){  
    if(h1f>ref/100+anchob/200){  
        u=valOFF;  
    }  
    if(h1f<ref/100-anchob/200){  
        u=valON;  
    }  
}
```

Una vez calculada la señal de control deben tenerse en cuenta las limitaciones físicas de la bomba. Su capacidad no puede superar el 100% ni pasar por debajo del 0%. Para asegurar este hecho se ejecuta el siguiente código:

```
if(u<0){
    u=0;
}else if( u>100){
    u=100;
}
```

Se calcula el caudal de entrada al depósito derecho y se actualizan los valores de señales de control, errores y salidas pasadas.

```
Qent=Qentmax*u/100;u1=u;
e2=e1;
e1=e0;
y1=h1f;
yf1=yf;
ef2=ef1;
ef1=ef;
```

Además, se tiene en cuenta el fenómeno de desbordamiento el cual se da cuando el depósito supera su altura máxima. En ese caso, se fuerza que la altura sea su valor máximo dado que el resto de agua abandona el depósito.

```
if(h2>=0.5){
    if(Qent>Qtub){
        desbord=true;
    }else {
        desbord=false;
    }
    h2=0.5;
}else {
    desbord = false;
}
```

En la página “Guardar\_datos” se encuentra el código que almacena los datos de la simulación en “arraydatos” para su posterior volcado en el fichero de texto plano. El código es el siguiente:

```
if(registro) {
    arraydatos.push("
    "+Number.parseFloat(t).toFixed(2)+espacio + "
    "+Number.parseFloat(h1f*100).toFixed(2)+espacio+ "
    "+control+espacio + "
    "+Number.parseFloat(ref).toFixed(2)+espacio+ "
    "+Number.parseFloat(u).toFixed(2)+"\n");
}
```

Para evitar introducir excesivos espacios dentro del código se ha utilizado la variable espacio que almacena los espacios equivalentes a cuatro tabulaciones. No se ha querido mezclar tabulaciones con espacios dado que en el momento de importar los datos del fichero de texto plano a programas como Excel se requiere de un único delimitador entre datos.

#### **4.2.4 Relaciones fijas**

En esta pestaña se encuentra el código a ejecutar de forma continua con cada iteración del método numérico o cambio en la interfaz.

La página creada se llama “Interfaz”.

En el caso de que el control se realice mediante un control automático la referencia gana interés. Es por ello por lo que para dibujarla en la gráfica se actualizan sus valores.

```
if(control!=1) {
    refy=ref;
    refx=t;
}
```



Se cambian también los valores de “tmin” y “tmax”, que son los valores mínimos y máximos en el eje x representados. Y el valor de “npuntos”, que es el número máximo de puntos almacenados por la gráfica. Si se aumenta la franja temporal visible se hace necesario aumentar dicho valor o de lo contrario parte de la gráfica permanecería vacía.

```
if( t > tvisible){
    tmin=t-tvisible;
    tmax=t;
}else{
    tmax=tvisible;
}

npuntos=Math.round(1.1*tvisible/T);
```

#### 4.2.5 Propio

Existen dos páginas de métodos propios.

La primera almacena la función “Guardar”, la cual se ejecuta al pulsar el botón de guardado de datos. Se encarga de crear un objeto tipo documento y un objeto tipo blob con los datos. El archivo se guarda como si de una descarga se tratase.

```
function guardar () {
    const a = document.createElement("a");
    const archivo = new Blob(arraydatos, { type:
'text/plain' });
    const url = URL.createObjectURL(archivo);
    a.href = url;
    a.download = "Datos Laboratorio Tanques
Comunicantes.txt";
    a.click();
    URL.revokeObjectURL(url);
}
```

La segunda contiene la función “DisNormal”, la cual genera números aleatorios que siguen una distribución normal. No existen en JavaScript funciones nativas que cumplan dicha función por lo que se usa la transformada de Box-Muller.

```
function disNormal () {  
  
    disu = 1 - Math.random();  
    disv = Math.random();  
    disz = Math.sqrt(-2.0*Math.log(disu))*Math.cos( 2.0  
* Math.PI*disv );  
    return disz * des_est/100 + media;  
}
```

Esta transformada es “un método de muestreo de números aleatorios con el cual generar números aleatorios, independientes, estándar, normalmente distribuidos (media cero, varianza unitaria), dada una fuente de números aleatorios distribuidos” (Canche, 2020) (Figura 50).

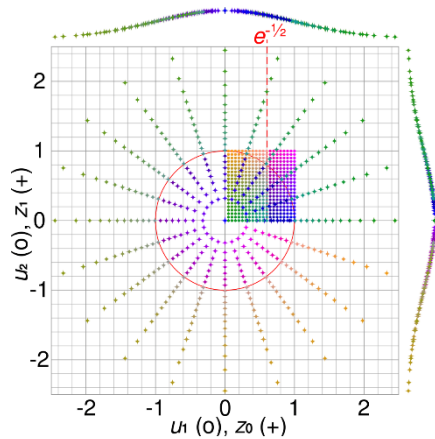


Figura 50 - Distribución de números normales mediante Box Muller

Una vez obtenido el número normalmente distribuido se realiza un cambio de base a la media y desviación estándar deseadas.

## 4.3 Créditos y Descripción

Para la creación de la página de descripción y de créditos no se ha querido usar el apartado que destina Easy Java/JavaScript Simulations para ello, debido a que genera una página poco vistosa con un formato pobre. En su lugar, se ha preferido crear las páginas web basadas en HTML, CSS y JavaScript utilizando el entorno de desarrollo Microsoft Visual Studio (Figura 51).

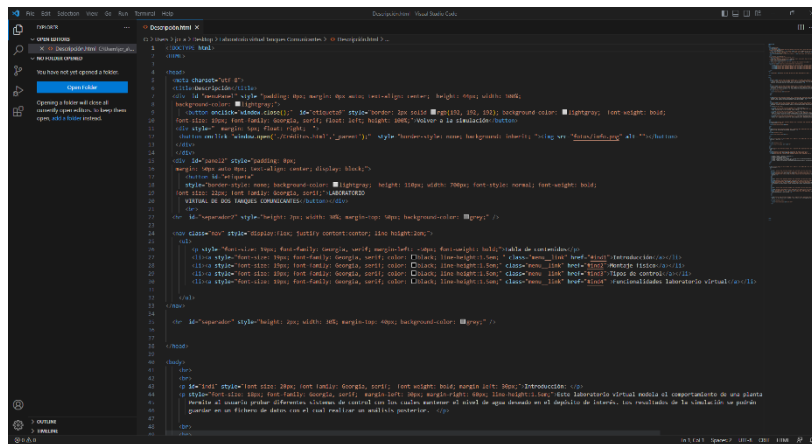


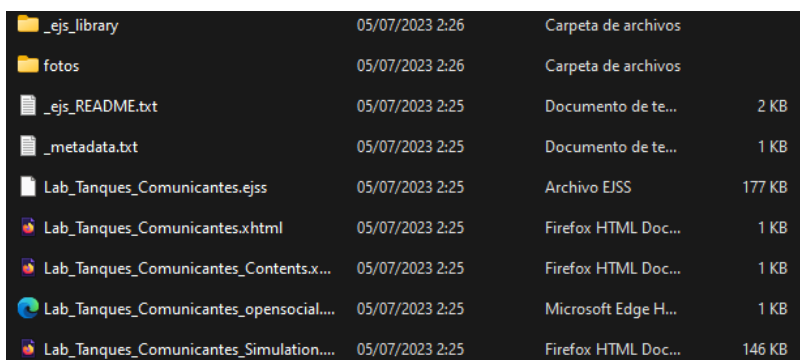
Figura 51 - Microsoft Visual Studio

Esta herramienta es compatible con numerosos lenguajes de programación y permite editar, depurar y compilar código, así como publicarlo. Es considerada como uno de los entornos de desarrollo más completos que existen.

La creación de estas páginas no posee especial interés y no será tratada. Dichas páginas se encuentran en los anejos y cualquier persona que lo desee puede editarlas en los lenguajes en los que se basan usando cualquier entorno.

## 4.4 Empaquetamiento de la simulación y detalles finales

Una vez terminado el laboratorio virtual desarrollado con Easy Java/JavaScript Simulations se empaqueta la simulación, creando un archivo comprimido (Figura 52).



Nombre	Fecha	Formato	Tamaño
_ejs_library	05/07/2023 2:26	Carpeta de archivos	
fotos	05/07/2023 2:26	Carpeta de archivos	
_ejs_README.txt	05/07/2023 2:25	Documento de te...	2 KB
_metadata.txt	05/07/2023 2:25	Documento de te...	1 KB
Lab_Tanques_Comunicantes.ejss	05/07/2023 2:25	Archivo EJSS	177 KB
Lab_Tanques_Comunicantes.xhtml	05/07/2023 2:25	Firefox HTML Doc...	1 KB
Lab_Tanques_Comunicantes_Contents.x...	05/07/2023 2:25	Firefox HTML Doc...	1 KB
Lab_Tanques_Comunicantes_opensocial....	05/07/2023 2:25	Microsoft Edge H...	1 KB
Lab_Tanques_Comunicantes_Simulation....	05/07/2023 2:25	Firefox HTML Doc...	146 KB

Figura 52 - Resultado del empaquetamiento de la simulación

Como ya se comentó, Easy Java/JavaScript Simulations crea por defecto una serie de archivos adicionales a la propia simulación. Este es el caso de los ficheros “\_ejs\_README.txt” y “\_metadata.txt”, los cuales se pueden borrar por carecer el primero de utilidad y el segundo haber sido sustituido por la página de créditos. El fichero “Lab\_Tanques\_Comunicantes.ejss” contiene el fichero interpretable por el entorno de desarrollo y es necesario para poder modificar la simulación, pero no para ejecutarla.

Adicionalmente, EJS crea una serie de páginas mediante en una página principal mostrar a la izquierda un índice de contenidos y a la derecha la simulación propiamente dicha. No poseyendo especial interés y añadiendo complejidad innecesaria se eliminan también. Conservando únicamente el fichero llamado “Lab\_Tanques\_Comunicantes\_Simulation.xhtml” el cual es el que realmente aloja la simulación.

Otro aspecto a tener en cuenta es que por defecto EJS también crea un bloque de créditos en la parte inferior de la página web de la simulación, el cual tampoco posee especial interés. Para eliminarlo, se abre con Microsoft Visual Studio el archivo “Lab\_Tanques\_Comunicantes\_Simulation.xhtml” y se busca la sección con id “metadata” (Figura 53) y se borra.

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

```
<div id="metadata" class="metadata">
<br />
<div id="title_author">
<hr />
<b>Title and author:</b>
<p>
Tanques Comunicantes<br />
</p>
<p>
Juan Carlos Ríos Vicente
</p>
</div>
<hr />
<p></p>
<div id="copyright_message">
<div class="cc_left" style="float:left">&#169; 2023, Juan Carlos Ríos Vicente. Compiled with EJS 6.01 (201013)
</div>
<div class="cc_right" style="float:right"> Released under a <a rel="license" target="_blank" href="http://creativecommons.org/
</div>
</div>
```

Figura 53 - Sección "metadata"

Se guardan los cambios y se agrupa en la misma carpeta (Figura 54) el fichero de la simulación, la carpeta “\_ejs\_library”, la página de descripción y la página de créditos. Deberá añadirse a la carpeta “fotos” todas las imágenes necesarias tanto para las páginas de descripción y créditos.

📁 _ejs_library	23/06/2023 4:00	Carpeta de archivos	
📁 fotos	23/06/2023 3:48	Carpeta de archivos	
🌐 Créditos.html	24/06/2023 20:55	Chrome HTML Do...	2 KB
🌐 Descripción.html	29/06/2023 12:22	Chrome HTML Do...	12 KB
🌐 Lab_Tanques_comunicantes.xhtml	26/06/2023 0:16	Firefox HTML Doc...	157 KB

Figura 54 - Archivos del programa final

Este conjunto de archivos supone el resultado del proyecto y para su ejecución bastaría con ejecutar el fichero “Lab\_Tanques\_Comunicantes.xhtml”. Si se desea subir el laboratorio virtual a un directorio en red bastaría con dar una url pública visible al fichero mencionado y que el resto de los archivos se encuentren bajo el mismo directorio del servidor.



# CAPÍTULO 5: Validación del laboratorio virtual

Tras la creación de un modelo el cual caracteriza un cierto fenómeno físico con el fin de realizar una simulación, es proceso imprescindible realizar una validación del mismo. El proceso de validación no es más que “obtener un grado de confianza en el mismo para el fin al que se destina” (Pardo Martín, 2018). En el caso particular del proyecto acometido, es esperable que el modelo no muestre los mismos valores absolutos que los obtenidos en la planta real. De igual manera, dos plantas reales previsiblemente no mostrarán los mismos valores absolutos para las mismas entradas. No obstante, sí se espera que la dinámica del sistema sea análoga.

Para realizar la validación del laboratorio virtual, se realizarán un conjunto de tres experimentos en la planta real. Cada experimento se someterá bajo unas condiciones de contorno y unas señales de control diferentes. Después, esos mismos experimentos se realizarán en el laboratorio virtual realizado y se compararán los resultados. El grado de similitud entre los resultados de la planta real y la planta virtual definirán su aceptabilidad y el criterio de decisión se basará en una dinámica similar y unos valores absolutos relativamente cercanos.

## 5.1 Primer experimento

Las condiciones de contorno consistirán en mantener la potencia de la bomba al 30% y la apertura de la válvula de salida al 38%. Se esperará hasta que se alcance el equilibrio y tomando ese instante como instante cero, se mandarán a la bomba las siguientes señales de control.

*Laboratorio virtual de una planta piloto de dos tanques comunicantes*

$u = 30 \%$	$t < 154$
$u = 35 \%$	$154 \leq t < 309$
$u = 30 \%$	$309 \leq t < 401$
$u = 28 \%$	$401 \leq t < 681$
$u = 25 \%$	$681 \leq t < 760$
$u = 35 \%$	$760 \leq t < 838$
$u = 32 \%$	$838 \leq t < 919$
$u = 29 \%$	$919 \leq t < 996$
$u = 27 \%$	$996 \leq t < 1087.5$
$u = 25 \%$	$1087.5 \leq t < 1165$
$u = 35 \%$	$1165 \leq t < 1236.5$
$u = 25 \%$	$1236.5 \leq t < 1315.5$
$u = 33 \%$	$1315.5 \leq t < 1418$
$u = 30 \%$	$1418 \leq t$

Estando expresada la señal de control en % y el tiempo en segundos. Para que el programa efectivamente emplee las señales de control definidas por la anterior función a trozos es necesario modificar el código de control. Para ello, se introduce el siguiente algoritmo en la página “Control” de la pestaña “Modelo”.



```
if (t<=154) {  
u=30;  
}  
if (t>=154.5 && t<=308.5) {  
u=35;  
}  
if (t>=309 && t<=400.5) {  
u=30;  
}  
if (t>=401 && t<=681) {  
u=28;  
}  
if (t>=681.5 && t<=760 ) {  
u=25;  
}  
if (t>=760.5 && t<=838 ) {  
u=35;  
}  
if (t>=838.5 && t<=918.5 ) {  
u=32;  
}
```

```
if (t>=919 && t<=995.5 ) {  
u=29;  
}  
if (t>=996 && t<=1087 ) {  
u=27;  
}  
if (t>=1087.5 && t<=1164.5  
) {  
u=25;  
}  
if (t>=1165 && t<=1236 ) {  
u=35;  
}  
if (t>=1236.5 && t<=1315 ) {  
u=25;  
}  
if (t>=1315.5 && t<=1417.5  
) {  
u=33;  
}  
if (t>=1418) {  
u=30;  
}
```

Una vez introducido el código, se ejecuta la simulación, la cual se realiza sin ruido de medida para una mayor precisión en los resultados. Se guardan los datos y se comparan con los datos reales de la planta. Primeramente, se comparan los datos de las señales de control. Al ser una señal mandada por el controlador y cuyo cambio es instantáneo no deberían existir diferencias entre los dos ensayos.

Efectivamente la coincidencia es exacta. En la gráfica de las señales de control puede observarse cómo ambas se encuentran solapadas (Figura 56).

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

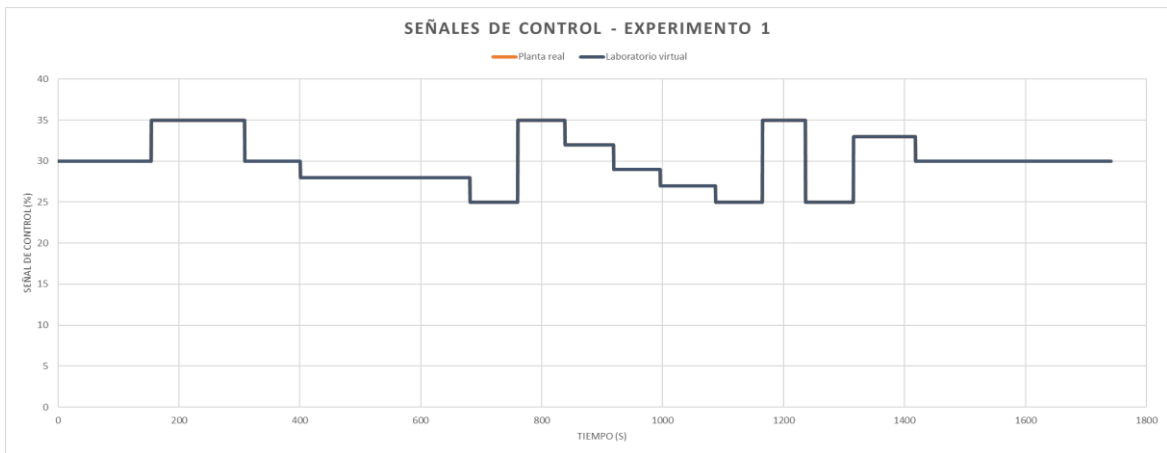


Figura 56 - Gráfica señal de control del experimento 1 (validación)

A continuación, se analiza la similitud entre las alturas del depósito izquierdo obtenidas en cada ensayo (Figura 55).

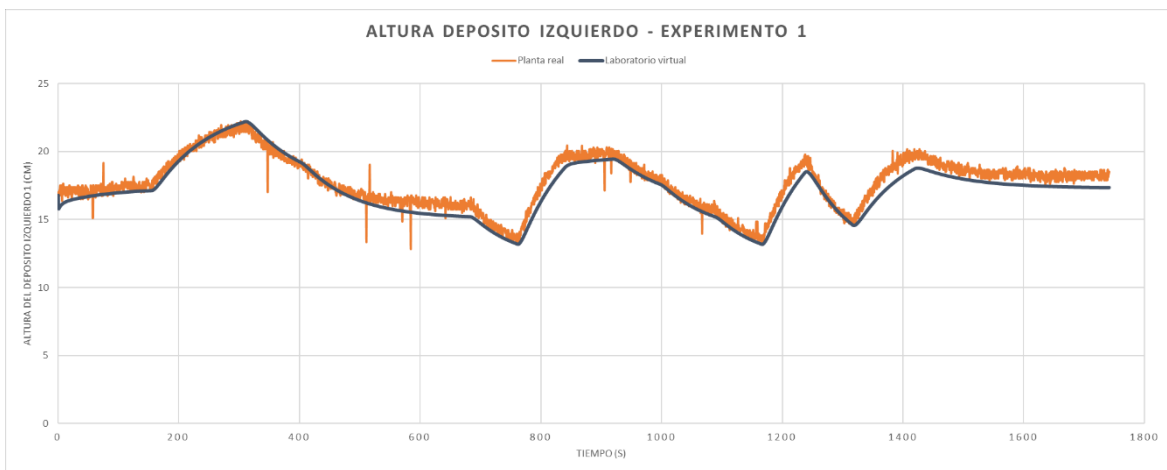


Figura 55 - Gráfica altura deposito izquierdo del experimento 1 (validación)

Como era esperable los valores absolutos no son idénticos, aunque sí similares. La dinámica del sistema sí se considera similar en ambos casos.

En base a los resultados de este experimento se determina que el laboratorio virtual sí representa con la suficiente fidelidad el fenómeno real de dos tanques comunicantes.

## 5.2 Segundo experimento

En este caso se partirá de condiciones iniciales nulas, lo cual quiere decir que ambos depósitos en el instante cero se encontrarán vacíos. El experimento durará 430 segundos y para una apertura de la válvula de salida del 24 % la señal de control tomará los siguientes valores:

$$\left\{ \begin{array}{ll} u = 100 \% & t < 14 \\ u = 30 \% & 14 \leq t < 33 \\ u = 20 \% & 33 \leq t < 167 \\ u = 22 \% & 167 \leq t < 375 \\ u = 30 \% & 375 \leq t \end{array} \right.$$

De nuevo, es necesario modificar el algoritmo de control de la simulación situado en la página "Control" de la pestaña "Modelo". Se introduce el código siguiente:

```
if(t<14) {
    u=100;
}
if(t>=14 && t<33) {
    u=30;
}
if(t>=33 && t<167) {
    u=20;
}
if(t>=167 && t<375) {
    u=22;
}
if(t>=375) {
    u=30;
}
```

Una vez introducido el código se vuelve a ejecutar la simulación. Se guardan los datos y de nuevo se comparan con los obtenidos del ensayo en la planta real. Primero se comparan los datos de las señales de control, las cuales deberían ser idénticos y después la altura del depósito izquierdo.

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

Las señales de control son idénticas y se encuentran solapadas (Figura 57). Tal y como se esperaba.

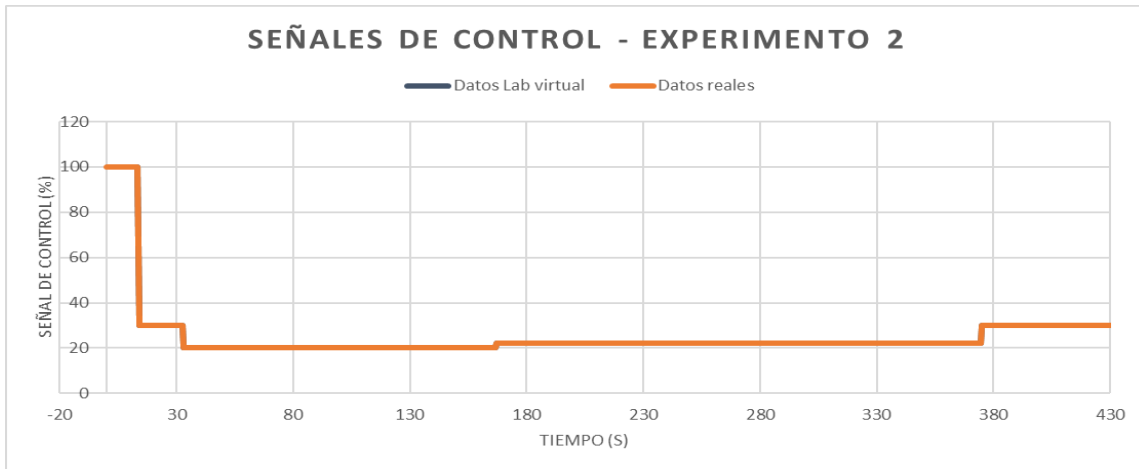


Figura 57 - Gráfica señales de control del experimento 2 (validación)

A continuación, se comparan los valores de las alturas del depósito izquierdo obtenidas en el ensayo real y en el ensayo del laboratorio virtual (Figura 58).

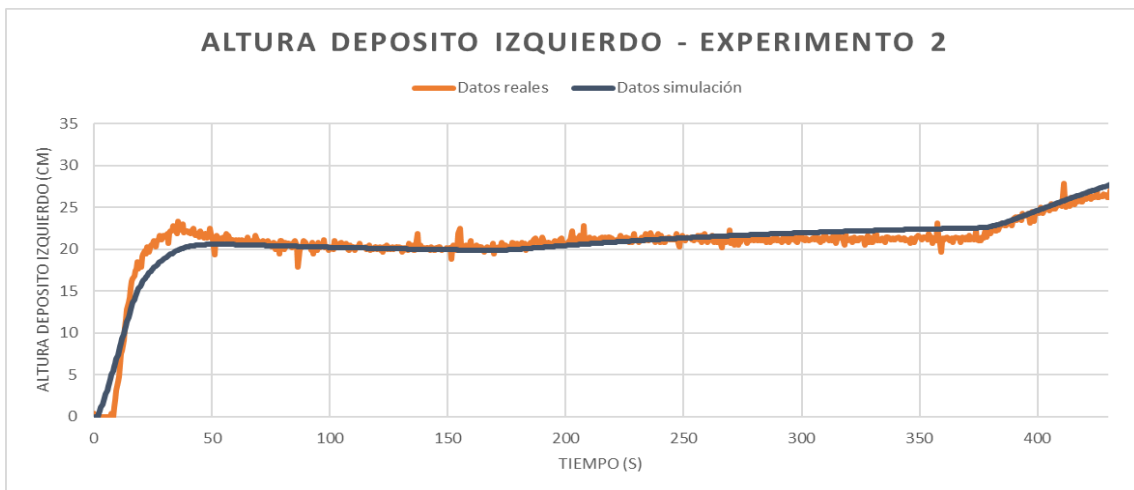


Figura 58 - Gráfica altura deposito izquierdo del experimento 2 (validación)

La dinámica sigue siendo similar, aunque los valores absolutos no sean idénticos.

Este experimento también valida el modelo al demostrar que la simulación representa con suficiente fidelidad el comportamiento del fenómeno a estudio.

### 5.3 Tercer experimento

Las condiciones de contorno consistirán en mantener la potencia de la bomba al 30% y la apertura de la válvula de salida al 36.2%. Se esperará hasta que se alcance el equilibrio y tomando ese instante como instante cero, se mandarán a la bomba las siguientes señales de control. Estando expresada la señal de control en % y el tiempo en segundos.

}	$u = 30 \%$	$t < 432$
	$u = 35 \%$	$432 \leq t < 1006$
	$u = 25 \%$	$1006 \leq t < 1602$
	$u = 35 \%$	$1602 \leq t < 1900,5$
	$u = 25 \%$	$1900,5 \leq t < 2196,5$
	$u = 35 \%$	$2196,5 \leq t < 2346,5$
	$u = 25 \%$	$2346,5 \leq t < 2496,5$
	$u = 35 \%$	$2496,5 \leq t < 2567$
	$u = 25 \%$	$2567 \leq t < 2613,5$
	$u = 35 \%$	$2613,5 \leq t < 2642,5$
	$u = 25 \%$	$2642,5 \leq t < 2672$
	$u = 30 \%$	$2672 \leq t$

Por última vez, se modifica el algoritmo de control de la simulación situado en la página “Control” de la pestaña “Modelo”. Se introduce el código siguiente:

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

```
if (t<432) {  
    u=30;  
}  
if (t>=432 && t<1006) {  
    u=35;  
}  
if (t>=1006 && t<1602) {  
    u=25;  
}  
if (t>=1602 && t<1900.5) {  
    u=35;  
}  
if (t>=1900.5 && t<2196.5) {  
    u=25;  
}  
if (t>=2196.5 && t<2346.5) {  
    u=35;  
}
```

```
if (t>=2346.5 && t<2496.5) {  
    u=25;  
}  
if (t>=2496.5 && t<2567) {  
    u=35;  
}  
if (t>=2567 && t<2613.5) {  
    u=25;  
}  
if (t>=2613.5 && t<2642.5) {  
    u=35;  
}  
if (t>=2642.5 && t<2672) {  
    u=25;  
}  
if (t>=2672) {  
    u=30;  
}
```

Una vez introducido el código se ejecuta la simulación y se guardan los datos. De nuevo se comparan los resultados obtenidos en ambos ensayos. Primero se comparan los datos de las señales de control, las cuales deberían ser idénticas y después las alturas del depósito izquierdo.

Las señales de control son idénticas y se encuentran solapadas, siendo el resultado esperable (Figura 59).

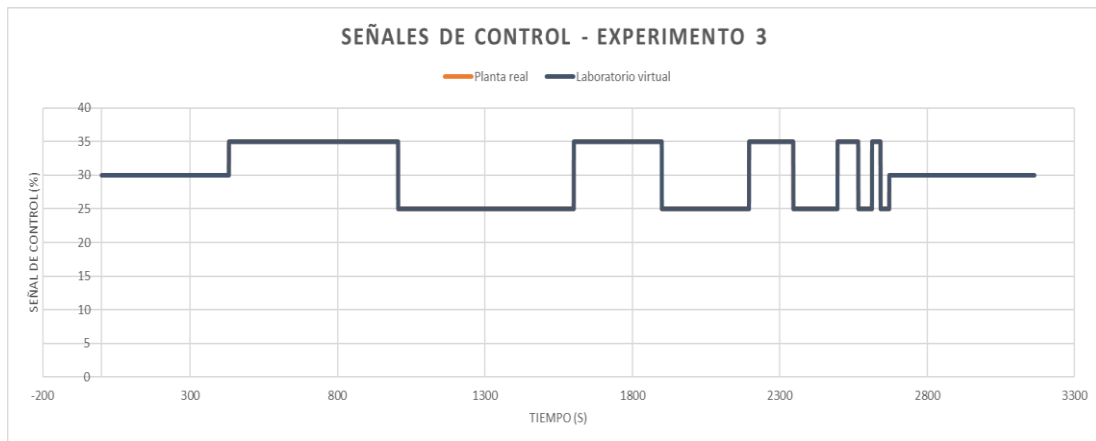


Figura 59 - Gráfica señales de control del experimento 3 (validación)

A continuación, se comparan los valores de las alturas del depósito izquierdo obtenidas en el ensayo real y las obtenidas en el ensayo mediante el laboratorio virtual (Figura 60).

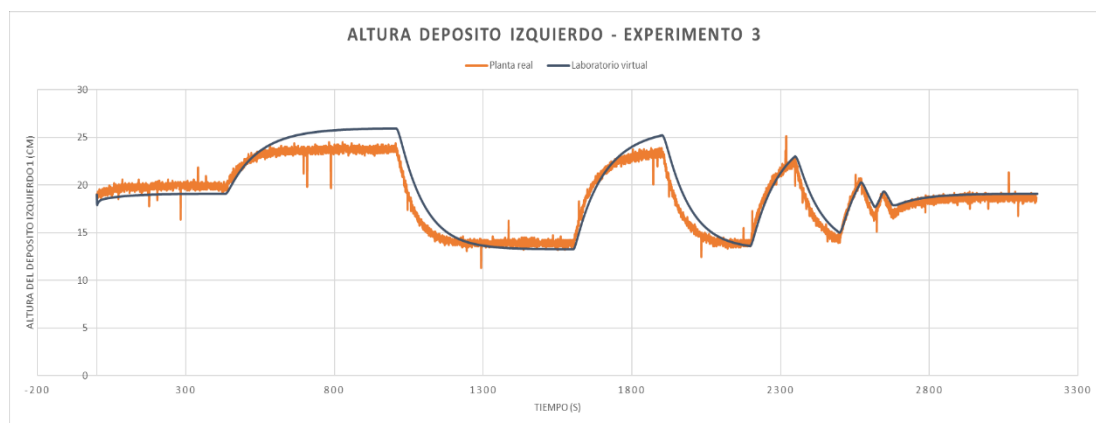


Figura 60 - Gráfica altura deposito izquierdo del experimento 3 (validación)

La dinámica sigue siendo similar, aunque los valores absolutos no sean idénticos.

Este experimento también valida el modelo al demostrar que la simulación representa con suficiente fidelidad el comportamiento del fenómeno a estudio.

## **5.4 Conclusión**

Los tres experimentos validan el laboratorio virtual al demostrar que este posee una dinámica análoga al sistema real a estudio. Pueden apreciarse ciertas diferencias en los valores absolutos de la altura del depósito izquierdo, pero es esperable y entendible dada las diferencias entre diferentes montajes de la misma planta.



# CAPÍTULO 6: Utilización del laboratorio virtual

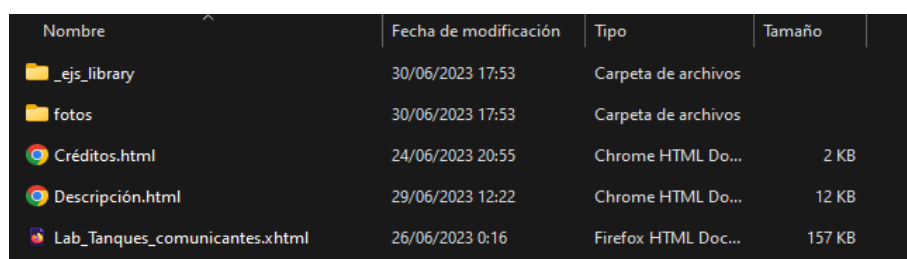
---

## 6.1 Apertura del laboratorio virtual

Para ejecutar el laboratorio virtual existen dos posibilidades:

La primera consiste en que el programa se encuentre subido en un servidor web. En dicha circunstancia, con acceder al enlace suministrado por su distribuidor mediante cualquier navegador es suficiente.

La segunda consiste en que recibamos los archivos del laboratorio virtual y lo ejecutemos de forma local. Previsiblemente se reciba un archivo comprimido cuyo contenido sea el siguientes (Figura 61):



Nombre	Fecha de modificación	Tipo	Tamaño
_ejs_library	30/06/2023 17:53	Carpeta de archivos	
fotos	30/06/2023 17:53	Carpeta de archivos	
Créditos.html	24/06/2023 20:55	Chrome HTML Do...	2 KB
Descripción.html	29/06/2023 12:22	Chrome HTML Do...	12 KB
Lab_Tanques_comunicantes.xhtml	26/06/2023 0:16	Firefox HTML Doc...	157 KB

Figura 61 - Archivos que componen el laboratorio virtual

En este caso, para ejecutar el laboratorio virtual será necesario abrir el archivo “Lab\_Tanques\_comunicantes.xhtml”. Es importante comentar que para el correcto funcionamiento del programa la carpeta “Fotos” se debe encontrar en el mismo directorio que los archivos tipo “html” y “xhtml”. Salvo indicación expresa, se abrirá con el navegador por defecto del usuario, ajuste irrelevante para su funcionamiento.

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

Una vez abierto el programa, se encuentra la siguiente interfaz (Figura 62):

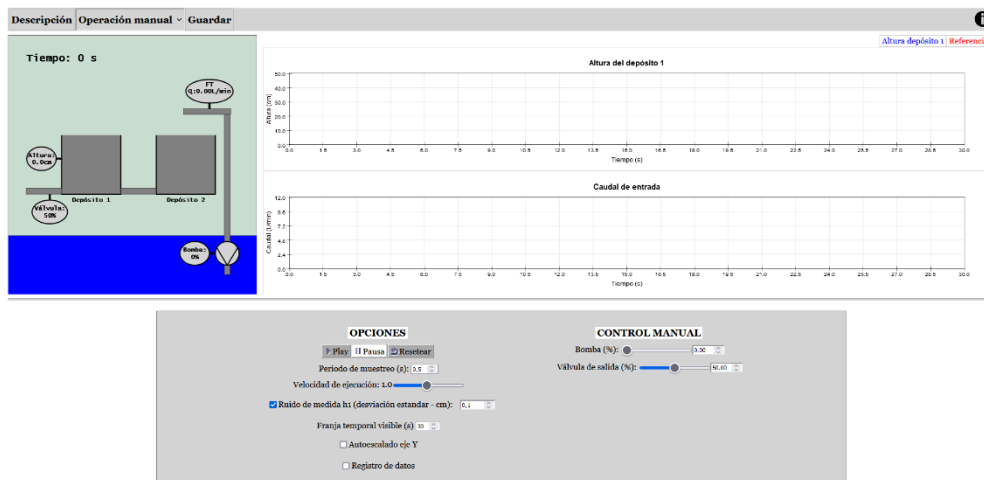


Figura 62 - Interfaz del laboratorio virtual

## 6.2 Barra de herramientas

En la parte superior puede encontrarse una barra de herramientas con diferentes opciones. La primera de ellas es el botón “Descripción”, el cual abre la página de descripción del montaje en una ventana emergente (Figura 63).

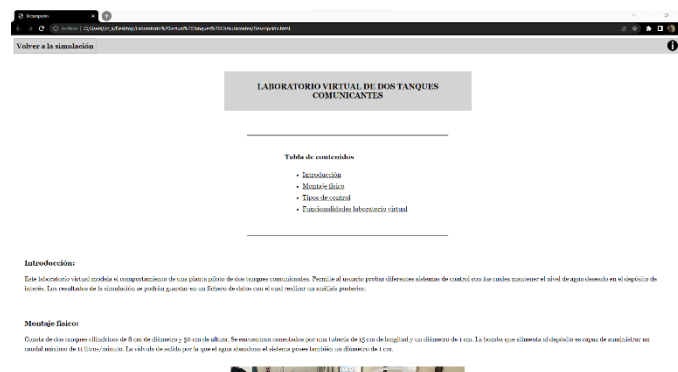
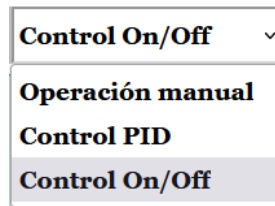


Figura 63 - Página de descripción del laboratorio virtual

Esta página explica brevemente el montaje a simular, así como las funcionalidades del laboratorio virtual. Consta de un índice interactivo el cual nos lleva al contenido que se desee visualizar. En su parte superior posee también una barra de herramientas con dos botones. El primero, a la izquierda de la barra, “Volver a la simulación”, cierra la ventana emergente de descripción. El otro botón, situado en la parte derecha, con un icono de

información, abre una ventana emergente con los créditos, en los cuales se muestra la autoría del programa.

Volviendo a la interfaz principal del laboratorio virtual (Figura 62), el botón a la derecha del botón “Descripción” indica “Operación manual”. Se trata en realidad de un menú desplegable que permite escoger la tipología de control que se desea realizar (Figura 64).



*Figura 64 - Menú desplegable de tipología de control*

Las tres opciones disponibles son las siguientes:

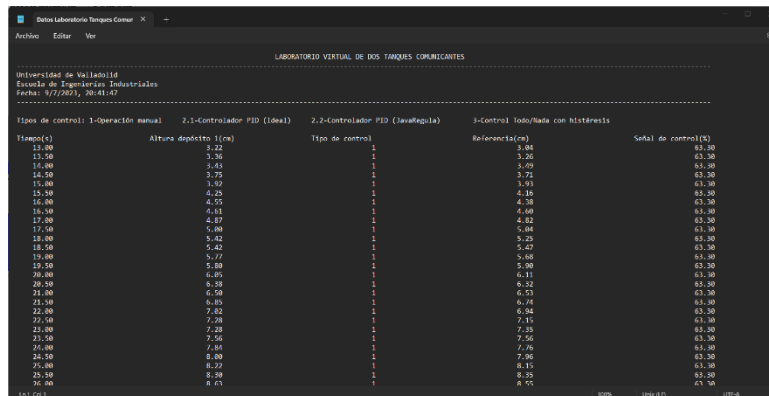
- Operación manual: permite al usuario fijar el valor de la potencia de la bomba y de apertura de la válvula manual de salida deseados en cualquier instante.
- Control PID: permite al usuario utilizar un controlador tipo PID que se encarga de determinar el valor de la potencia de la bomba para que el depósito izquierdo alcance el nivel de agua indicado por la referencia.
- Control On/Off: permite al usuario utilizar una tipología de control Todo/Nada con histéresis que de igual manera trata de mantener el valor del nivel de agua del depósito izquierdo próximo a la referencia.

La opción seleccionada en este menú desplegable determina qué parámetros se pueden ver y modificar en la interfaz. Los parámetros de control visibles serán en cada instante los pertenecientes a la topología de control en uso. Se comentará posteriormente de qué ajustes se trata.

A continuación, a la derecha de este menú desplegable se encuentra el botón “Guardar”, el cual almacena los datos de la simulación en un fichero de texto plano denominado “Datos Laboratorio Tanques Comunicantes”. Cabe notar que los estados de la simulación guardados en el fichero son únicamente los que hayan tenido lugar con la casilla de selección “Registro de datos” marcada.

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

La estructura del archivo de guardado de datos es la siguiente (Figura 65):



LABORATORIO VIRTUAL DE DOS TANQUES COMUNICANTES

Universidad de Valladolid  
Escuela de Ingenierías Industriales  
Fecha: 9/7/2021, 20:41:47

Tipos de control: 1-Operación manual 2,1-Controlador PID (Ideal) 2,2-Controlador PID (Lavafregula) 3-Control todo/nada con histéresis				
Tiempo(s)	Altura depósito 1(cm)	Tipo de control	Referencia(cm)	Señal de control(K)
13.00	3.22	1	3.84	63.00
13.50	3.56	1	3.26	63.00
16.00	3.43	1	3.69	63.00
16.50	3.75	1	3.71	63.00
17.00	3.93	1	3.93	63.00
18.50	4.25	1	4.16	65.00
19.00	4.55	1	4.38	63.00
19.50	4.61	1	4.60	63.00
19.90	4.87	1	4.82	63.00
19.90	5.00	1	5.04	63.00
19.90	5.42	1	5.26	63.00
19.90	5.42	1	5.47	63.00
19.90	5.77	1	5.69	63.00
19.90	5.89	1	5.90	63.00
20.00	6.05	1	6.11	63.00
20.00	6.59	1	6.32	65.00
21.00	6.50	1	6.53	63.00
21.50	6.95	1	6.74	63.00
21.50	7.02	1	6.94	63.00
22.00	7.28	1	7.15	63.00
23.00	7.49	1	7.35	63.00
23.50	7.56	1	7.56	63.00
24.00	7.65	1	7.76	63.00
24.50	8.00	1	7.96	63.00
25.00	8.27	1	8.15	63.00
25.50	8.59	1	8.35	63.00
26.00	8.63	1	8.55	63.00

Figura 65 - Archivo guardado datos

Este archivo posee un título que indica el laboratorio al que pertenecen los datos y una cabecera que denota la universidad y escuela a la que pertenece, así como la fecha y hora de realización de la simulación. Posteriormente se incluye una aclaración de a qué tipo de control se refieren los diferentes valores que puede adoptar la variable y a continuación están los datos, separados en columnas. Los datos registrados son el tiempo, la altura del depósito izquierdo, la tipología de control empleada, la referencia aplicada para los sistemas de control PID o Todo/Nada, y la señal de control de cada instante.

Si se desea realizar un análisis de estos datos en aplicaciones terceras es posible exportarlos mediante, por ejemplo, Excel, un programa de hojas de cálculo del paquete MS Office.

El procedimiento a seguir consiste en eliminar la cabecera del archivo, dejando únicamente los datos y usar la opción de "Obtener datos de texto/CSV". A continuación, aparece un cuadro de diálogo (Figura 66).

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

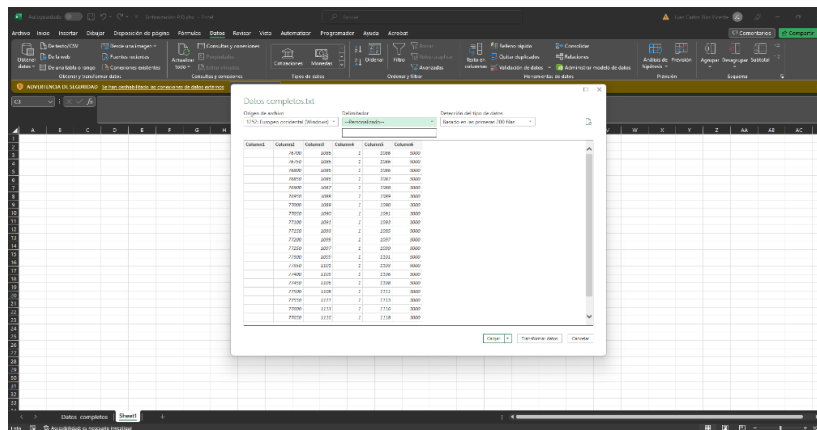


Figura 66 - Exportación de datos a Excel

El programa nos pide indicar el tipo de separación entre datos y nos permite previsualizar las columnas en las que los separa.

Por defecto Excel considera un delimitador de datos “Personalizado”, consistente en los espacios existentes entre valores. Ese ajuste es suficiente y solo resta hacer clic sobre el botón “Cargar” para importarlos a la hoja de cálculo.

Si por el contrario se desea importar los datos en Matlab, no es necesario eliminar la cabecera del archivo de datos. Se abre Matlab y en la barra de herramientas, en la pestaña “Home”, se hace clic sobre el botón “Import Data” situado en la sección “Variable”. Se abre un cuadro de selección de archivo en el que se busca el fichero de datos y se abre (Figura 67).

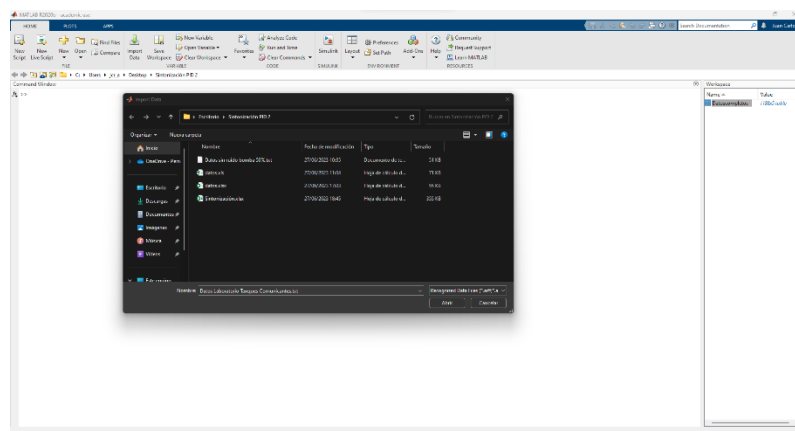


Figura 67 - Cuadro de selección de archivo de datos para exportación de datos a Excel

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

A continuación, se abre el cuadro de dialogo de importación (Figura 68) en el que se escoge la opción “Fixed Width” (ancho fijo) y en la opción Range (Rango de datos a exportar) se indica de “A9” (posición de la celda de la primera fila de datos de la primera columna) a “AX”. Siendo “X” el número de la última fila con datos.

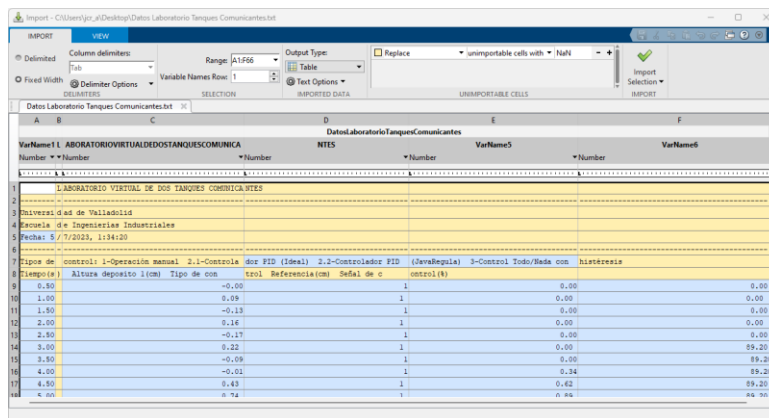


Figura 68 - Cuadro de importación de datos a Matlab

Los datos se cargan en Matlab en forma de tabla (Figura 69). En caso de mal reconocimiento de los datos es posible ajustar los datos cargados manualmente.

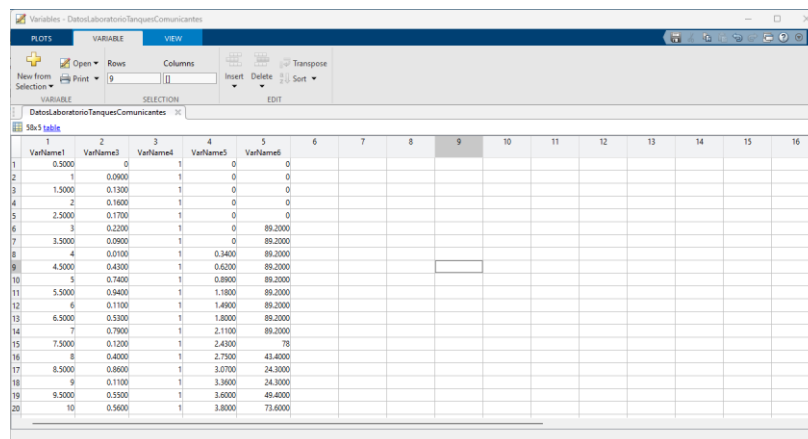


Figura 69 - Datos importados a Matlab

Volviendo a la interfaz principal del programa (Figura 62), en el extremo derecho de la barra de herramientas se encuentra también un botón de información el cual al realizar un clic se abre la página de créditos (Figura 70):



Figura 70 - Página de créditos

La página de descripción se abre en una ventana emergente y en ella se reconoce la autoría de este proyecto, realizado por Juan Carlos Ríos Vicente, alumno de la Escuela de Ingenierías Industriales de la Universidad de Valladolid. El único botón disponible en la barra de herramientas dice “Volver a la simulación” y su acción es cerrar la ventana emergente.

### 6.3 Visualización gráfica

Volviendo a la interfaz principal del laboratorio virtual (Figura 62), puede apreciarse cómo debajo de la barra de herramientas se encuentran la visualización de la simulación y a la derecha las gráficas.

La visualización de la simulación, la cual se encuentra en la parte izquierda, es la siguiente (Figura 71):

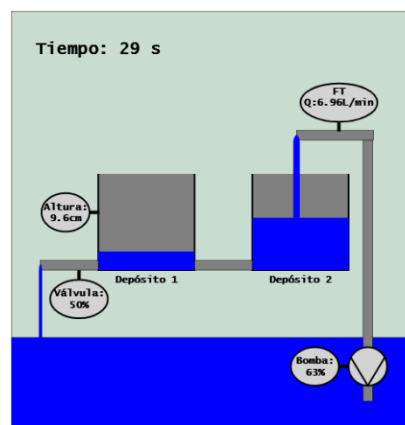


Figura 71 - Sinóptico de la simulación

### Laboratorio virtual de una planta piloto de dos tanques comunicantes

El dibujo representa los tanques comunicados a estudio y su evolución temporal. En la esquina superior izquierda aparece el tiempo de simulación y también pueden verse los instrumentos de medición que incorpora el montaje con sus respectivas lecturas. Estos son un indicador de la potencia de la bomba en %, un caudalímetro que muestra el caudal suministrado por la bomba en L/min, un indicador de la altura del nivel de agua del depósito izquierdo en cm, y un indicador de la apertura de la válvula de salida en %. Sus lecturas se actualizan en cada instante a los valores de la simulación. Los instrumentos representados reproducen el tipo de instrumentación que se encuentra en la planta real. Todo ello con el objetivo de asemejar lo máximo posible la experiencia de laboratorio del programa a la del montaje real.

En caso de que el sistema desborde, se representan dos flujos que caen desde la parte superior del tanque a la bancada (Figura 72).

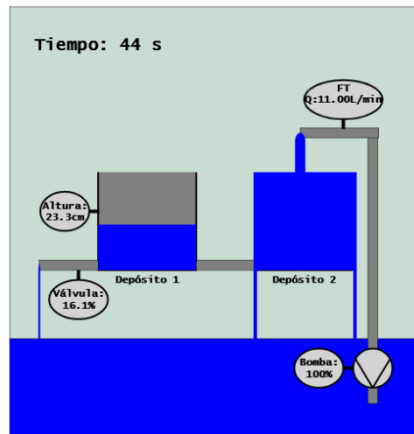


Figura 72 – Representación del desbordamiento del tanque

Nótese cómo el ancho de las representaciones de los caudales de entrada y salida del sistema, así como los que representan su desbordamiento, varían según sus respectivas magnitudes.



En cuanto a las gráficas de la parte derecha, poseen la siguiente forma (Figura 73):

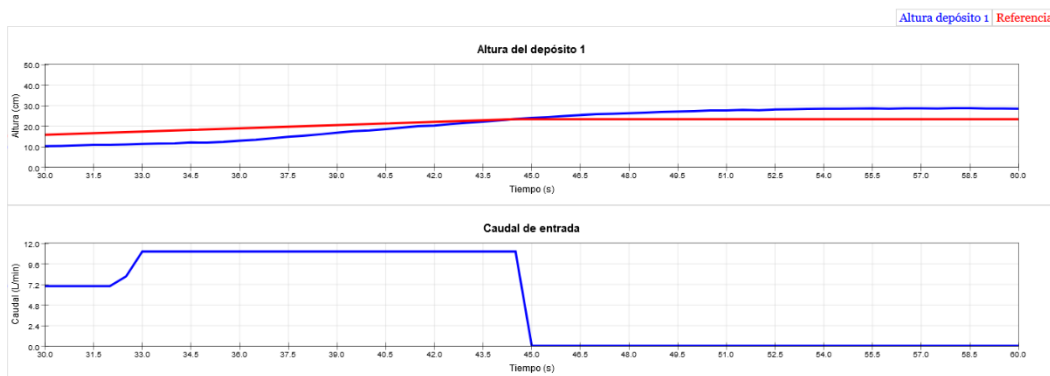


Figura 73 – Gráficas del laboratorio virtual

La gráfica de la parte superior representa con una traza azul la evolución de la altura del depósito izquierdo (cm) respecto del tiempo (s). En caso de que el tipo de control sea diferente del manual, adicionalmente aparecerá con trazo rojo la referencia (cm) seguida por el controlador.

En la gráfica de la parte inferior, se representa con una traza también azul el caudal de entrada al depósito derecho (L/min) respecto del tiempo (s).

Para evitar confusiones en la interpretación de los datos de la gráfica de la parte superior, que posee dos rectas, se ha incluido una leyenda en la parte superior derecha.

## 6.4 Ajustes de la simulación y los controladores

Debajo de la ilustración de la simulación y de las gráficas puede encontrarse tanto los ajustes de la simulación como los parámetros de la tipología de control en uso. Se comentarán primero los ajustes de la simulación (Figura 74).

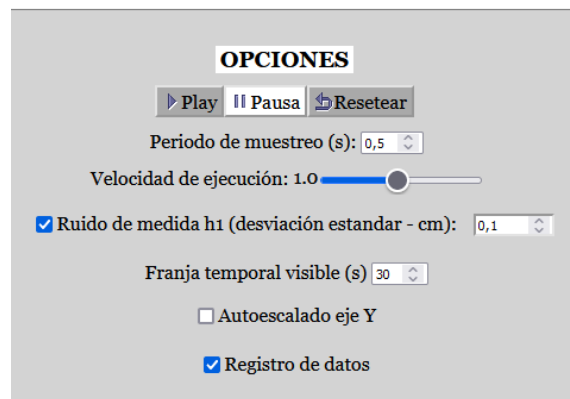


Figura 74 - Ajustes de la simulación

Después del título que indica que se trata de las opciones, la primera fila consta de los botones “Play”, “Pausa” y “Resetear”. Sus funciones son respectivamente reanudar la simulación, detenerla, y resetearla; esta última implica borrar todos los datos y volver a las condiciones iniciales.

A continuación, aparece el periodo de muestreo, el cual marca la diferencia de tiempo entre tiempos en los cuales el método numérico calcula el estado del sistema. Cuanto más pequeño sea el periodo de muestreo con mayor precisión se conocerá la evolución del sistema, pero peor rendimiento tendrá el laboratorio virtual. Esto debido a la necesidad de realizar un mayor número de iteraciones por unidad de tiempo.

En la siguiente fila se encuentra “Velocidad de ejecución”. La alteración del slider modifica el valor inicialmente en 1, y que puede oscilar entre 0,1 y 2. Representa la relación entre el tiempo de simulación y el tiempo real. Un valor de 2 implica que la simulación se ejecuta a una velocidad doble a la que el fenómeno tendría lugar en una planta física real.

La siguiente fila permite ajustar el error de medida, el cual representa el error que poseería el instrumento en un montaje real. Consta de una casilla de selección y de un cuadro de texto. La casilla de selección permite activar o desactivar el ruido de medida, y el cuadro de texto modificar su valor. El ruido de medida sigue una distribución normal cuya desviación estándar es el valor indicado en el recuadro de texto. Desactivar el ruido de medida implica suponer un instrumento de medida ideal, sin error en la toma de medidas.

La siguiente fila permite ajustar la franja temporal visible, la cual se ajusta por medio de un cuadro de texto. Este valor, introducido en segundos, marca

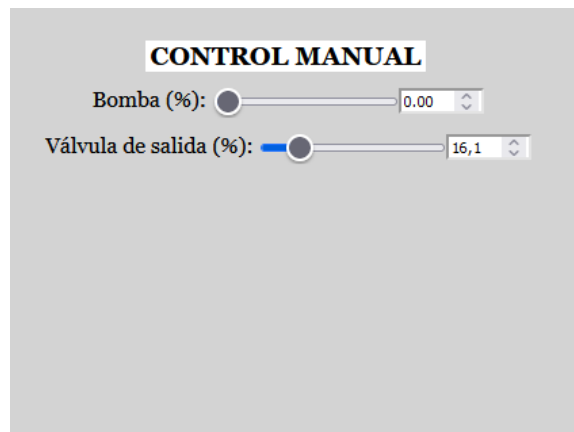
cuantos segundos anteriores al instante actual se pueden visualizar en las gráficas. Un mayor valor implica apreciar mejor el histórico de estados del fenómeno, pero reduce el rendimiento del programa.

El siguiente ajuste consiste en una casilla de selección mediante la cual se activa o se desactiva el autoescalado en el eje de ordenadas en ambas gráficas.

Como último ajuste se encuentra el registro de datos. Consiste en una casilla de selección que se puede activar y desactivar. Si esta casilla está marcada los datos de la simulación se registrarán internamente y, en el caso de hacer uso posteriormente del botón “Guardar”, se almacenarán en el fichero de texto plano. Si se da el caso de haber mantenido esta casilla desactivada y hacer clic sobre el botón “Guardar”, se almacenará un fichero de texto plano, con la cabecera y los títulos de las columnas, pero sin datos.

A la parte derecha de las opciones de la simulación se encuentran los ajustes de los parámetros de la tipología de control en uso.

En el caso de “Operación manual” los ajustes disponibles son los siguientes (Figura 75):



*Figura 75 - Parámetros de Operación manual*

Es posible modificar tanto la potencia de la bomba como la apertura de la válvula salida. Ambos valores en porcentaje.

En el caso de “Control PID”, los ajustes disponibles son (Figura 76):

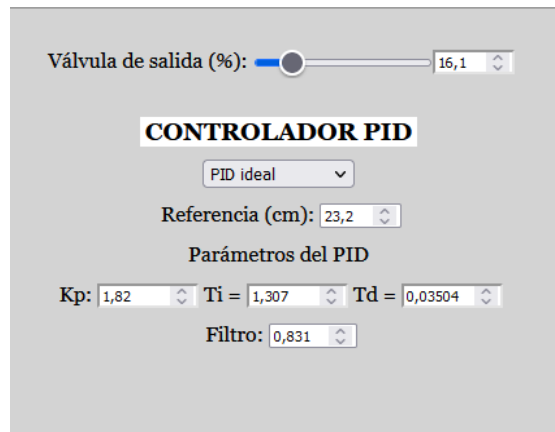


Figura 76 - Parámetros de control PID

Es posible modificar la apertura de la válvula de salida (en %), el tipo de controlador PID, la referencia (cm), la constante proporcional (cm/%), el tiempo integral (minutos), el tiempo derivativo (min) y la constante del filtro.

El tipo de controlador PID se selecciona según un menú desplegable y las dos opciones disponibles son PID ideal con filtro y PID de JavaRegula.

El controlador tratará de alcanzar la referencia especificada mediante los parámetros especificados.

En el caso de "Control On/Off", los ajustes disponibles son (Figura 77):

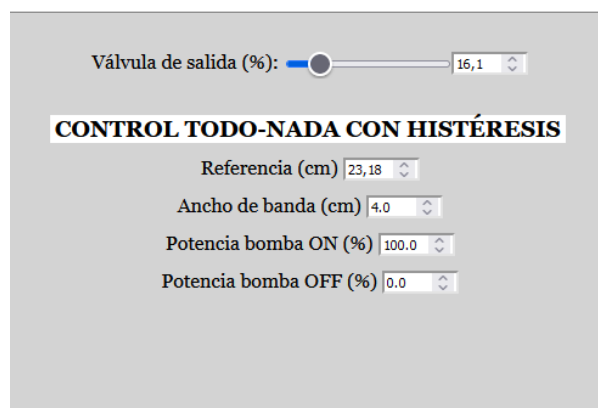


Figura 77 - Parámetros del controlador Todo/Nada con histéresis

### *Laboratorio virtual de una planta piloto de dos tanques comunicantes*

Es posible modificar la apertura de la válvula de salida (%), la referencia (cm), el ancho de banda (cm), el valor de la potencia de la bomba en el caso ON (%), y en el caso OFF (%).

La potencia de la bomba alternará entre los valores ON y OFF según la altura del depósito evolucione dentro de la franja delimitada por la referencia y el ancho de banda.



# CAPÍTULO 7: Casos prácticos

## 7.1 Operación manual

La operación manual puede ser útil para conocer el punto de equilibrio dado un caudal de entrada y una apertura de la válvula de salida o bien para realizar el experimento de obtención de datos para la sintonización de un controlador PID. Habitualmente, se calcula el sistema de primer orden equivalente correspondiente a un salto en la señal de control desde un punto de equilibrio.

Ejemplo de obtención de un punto de equilibrio: Para una potencia de la bomba del 50% y para una apertura de salida del 50% se obtiene la siguiente evolución de la altura del depósito (Figura 78).

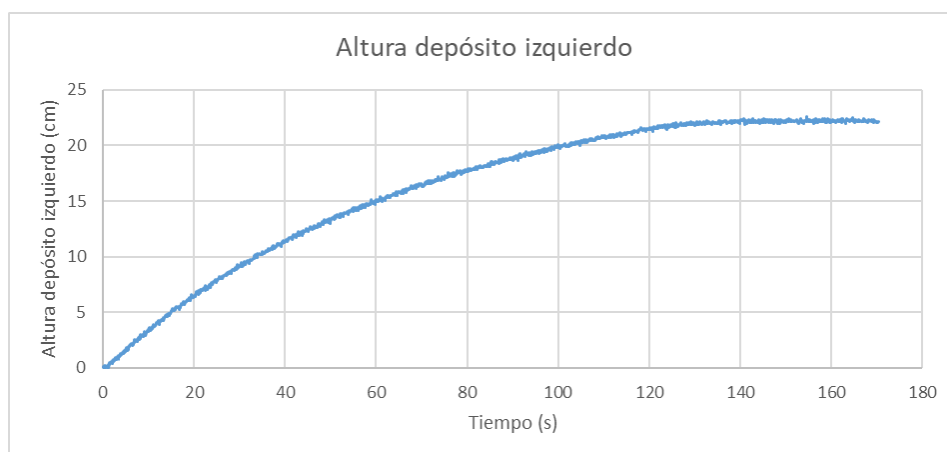


Figura 78 - Evolución de la altura del depósito izquierdo

Se observa que el punto de equilibrio se encuentra a una altura de 22,15 cm aproximadamente.

## 7.2 Controlador PID (Sintonización)

El controlador PID es la tipología de control más comúnmente utilizada. La situación habitual de utilización de este tipo de control es trabajar en torno a un punto de equilibrio a partir del cual se realizan pequeñas variaciones en la referencia o bien aparecen perturbaciones (cambio en la apertura de la válvula de salida). Es necesario sintonizar el controlador, lo cual consiste en determinar los parámetros a aplicar. Se trata de un problema de optimización con solución

múltiple donde se decidirá en base a las especificaciones de diseño deseadas (mejor respuesta ante cambios en la referencia, frente a perturbaciones...).

Las sintonizaciones aquí realizadas se van a realizar por uso de tablas y fórmulas, aplicando las dos siguientes metodologías:

- Sintonía de Rivera-Morari (también conocida como “Lambda Tuning”) permite obtener los parámetros del controlador con los cuales lograr la constante de tiempo deseada en el sistema en lazo cerrado. Según la rapidez de transitorio obtenida se clasifica al controlador obtenido como conservador, moderado o agresivo).
- Sintonía de López permite obtener los parámetros del controlador enfocándose en un buen rechazo de perturbaciones. Es válido para procesos monótonos con  $d / \tau < 1$ . Existen tres diferentes criterios de sintonía aplicables según la magnitud del error que se desea minimizar.

### 7.2.1 Obtención del sistema de primer orden aproximado

Seleccionando como potencia de la bomba el 25% y como apertura de la válvula el 40% se alcanza un punto de equilibrio en cual la altura del depósito izquierdo es de 10,86 cm. A continuación, se aumenta la potencia de la bomba hasta el 30%.

La evolución temporal del sistema es la siguiente (Figura 79):

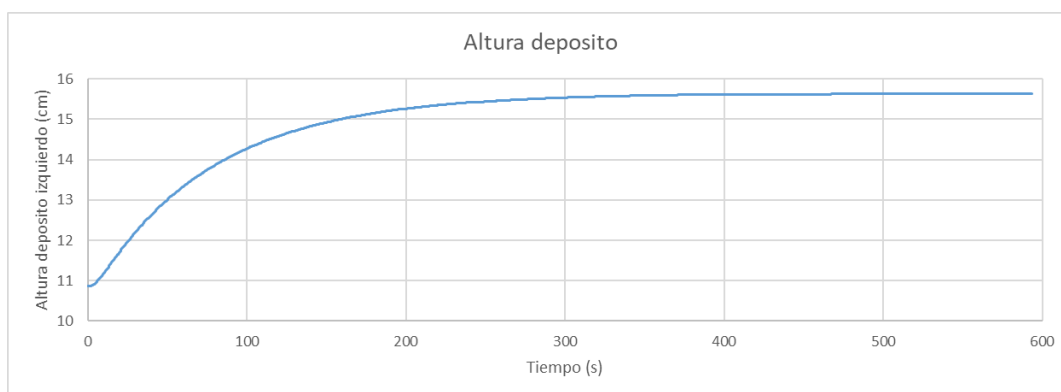


Figura 79 - Evolución temporal de la altura del depósito



Este sistema se asemeja a uno de primer orden por lo que se procede a obtener el sistema aproximado equivalente (Ganancia, constante de tiempo y retardo). Para ello, se utiliza el software “Loop-Pro”, el cual realiza un ajuste por mínimos cuadrados (Figura 80).

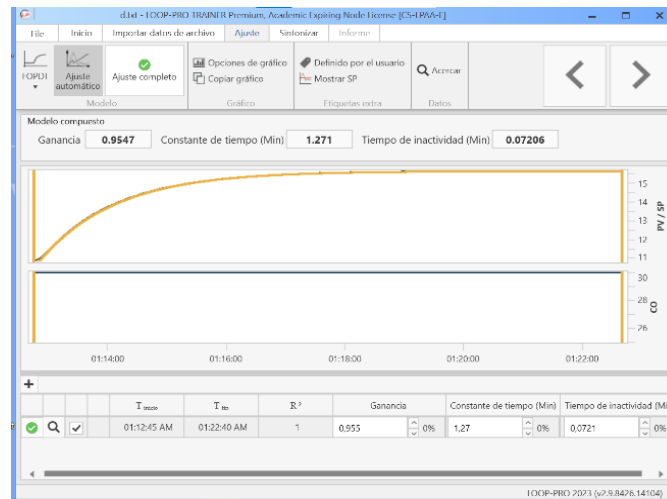


Figura 80 - Ajuste por mínimos cuadrados

Siendo los parámetros obtenidos mediante el ajuste los siguientes:

- $K$  (Ganancia) = 0.9547 (cm/%).
- $\tau$  (Constante de tiempo) = 1.271 (min).
- $d$  (Retardo) = 0.07206 (min).

## 7.2.2 Sintonía Lambda Tuning PID sin filtro

Mediante el programa “Loop-Pro” se realiza la sintonía de un controlador PID sin filtro aplicando el método “Lambda Tuning”. Fijando un tiempo de respuesta del sistema en lazo cerrado de 0,30415 min los parámetros obtenidos son los siguientes:

- $K_p$  = 2.013 (cm/%).
- $T_i$  = 1.307 (min).
- $T_d$  = 0.03504 (min).
- Alfa = 0.

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

A continuación, se introducen los parámetros del controlador en el laboratorio y se experimenta realizando un salto en la referencia a 15 cm desde el punto de equilibrio (Figura 81).

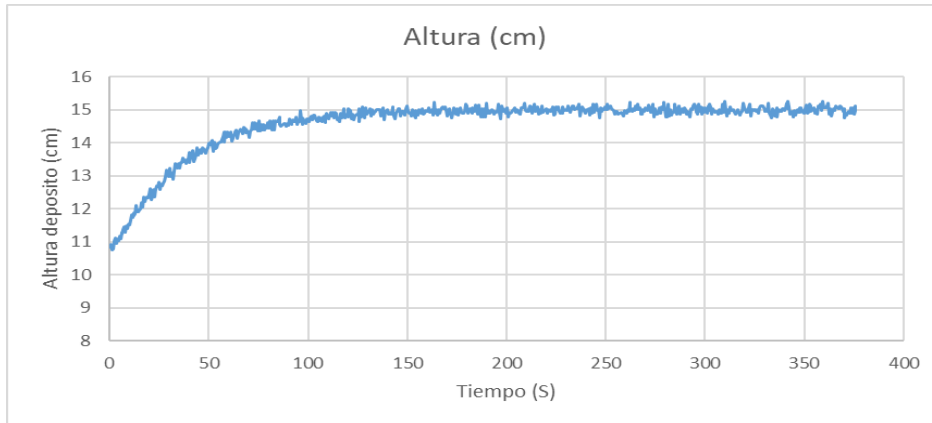


Figura 81 - Evolución altura depósito ante salto en la referencia con controlador PID sin filtro (Lambda Tuning)

La evolución temporal de la señal de control es la siguiente (Figura 82):

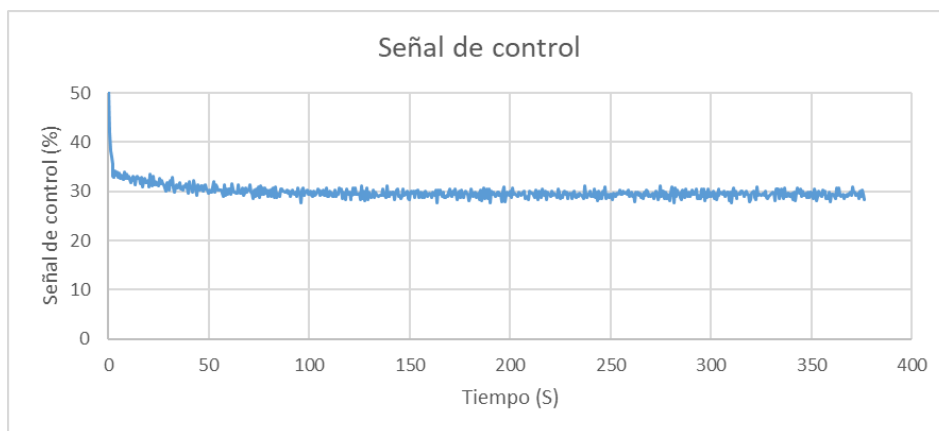


Figura 82 - Evolución señal de control ante salto en la referencia con controlador PID sin filtro (Lambda Tuning)

A continuación, se vuelve al punto de equilibrio y se realiza una perturbación consistente en incrementar la apertura de la válvula de salida hasta el 50%. La evolución de la altura del depósito es la siguiente (Figura 83):

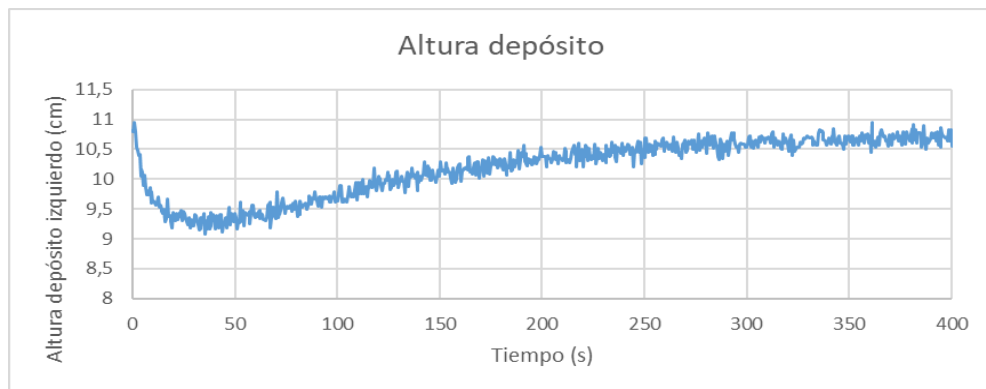


Figura 83 - Evolución altura depósito ante perturbación con controlador PID sin filtro (Lambda Tuning)

La señal de control de este último experimento es la siguiente (Figura 84):

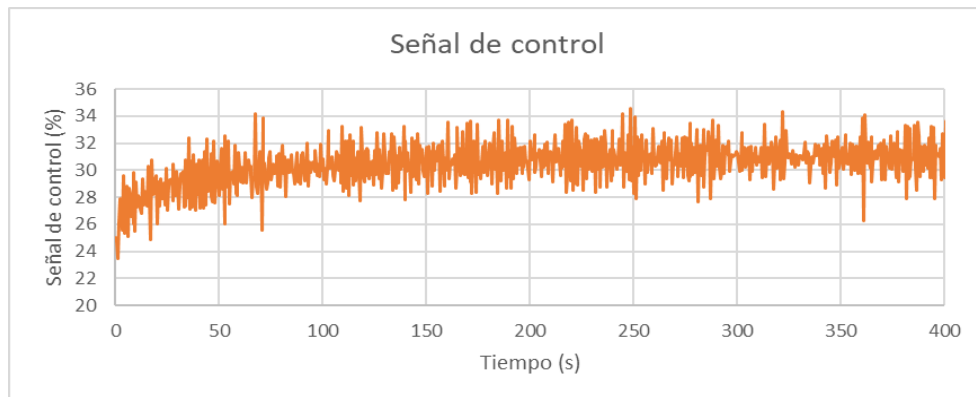


Figura 84 - Evolución señal de control ante perturbación con controlador PID sin filtro (Lambda Tuning)

El ruido de medida provoca una conmutación elevada en la señal de control al carecer de filtro en el término derivativo.

### 7.2.3 Sintonía Lambda Tuning PID con filtro

Mediante el programa "Loop-Pro" se realiza la sintonía de un controlador PID con filtro aplicando el método "Lambda Tuning". Fijando un tiempo de respuesta del sistema en lazo cerrado de 0,30415 min los parámetros obtenidos son los siguientes:

- $K_p = 1.82$  (cm/%).
- $T_i = 1.307$  (min).

### Laboratorio virtual de una planta piloto de dos tanques comunicantes

- $T_d = 0.03504$  (min).
- Alfa = 0.8314.

A continuación, se introducen los parámetros del controlador en el laboratorio y si experimenta realizando un salto en la referencia a 15 cm desde el punto de equilibrio (Figura 85).

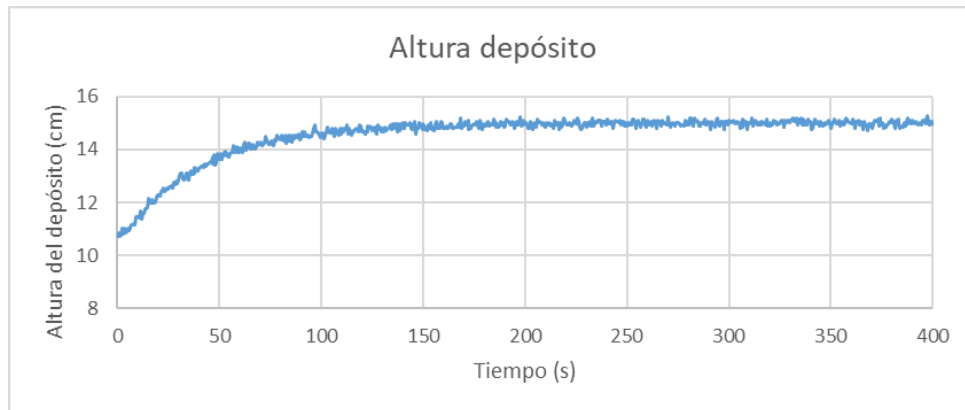


Figura 85 - Evolución altura depósito ante salto en la referencia con controlador PID con filtro (Lambda Tuning)

La evolución temporal de la señal de control es la siguiente (Figura 86):

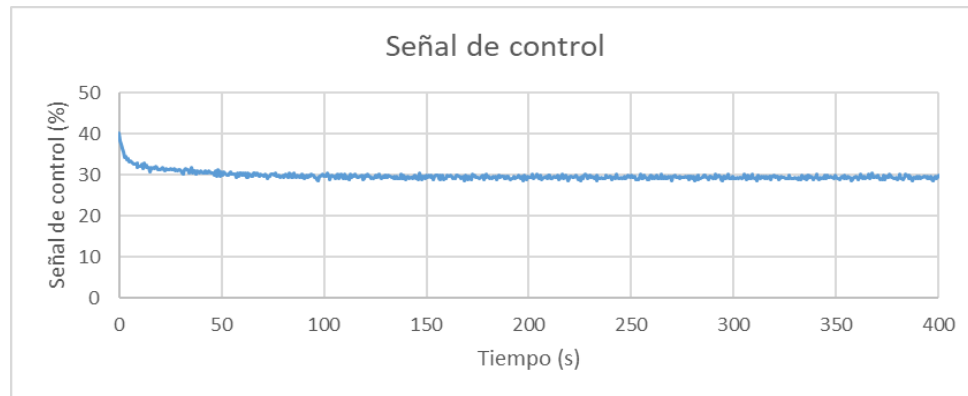


Figura 86 - Evolución señal de control ante salto en la referencia con controlador PID con filtro (Lambda Tuning)

Gracias a la presencia del filtro de paso bajo la señal de control reduce su oscilación, reduciendo el esfuerzo de control del actuador.

A continuación, se vuelve al punto de equilibrio y se realiza una perturbación consistente en incrementar la apertura de la válvula de salida hasta el 50%. La evolución de la altura del depósito es la siguiente Figura 87):

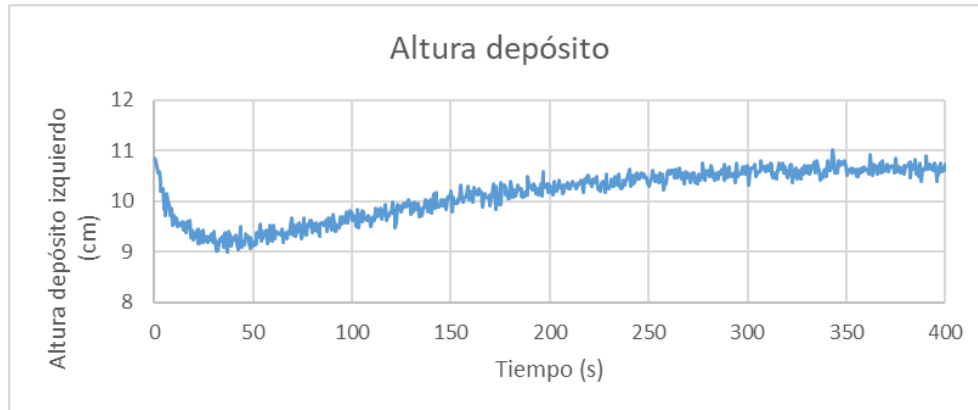


Figura 87 - Evolución altura depósito ante perturbación con controlador PID con filtro (Lambda Tuning)

La señal de control de este experimento es la siguiente (Figura 88):

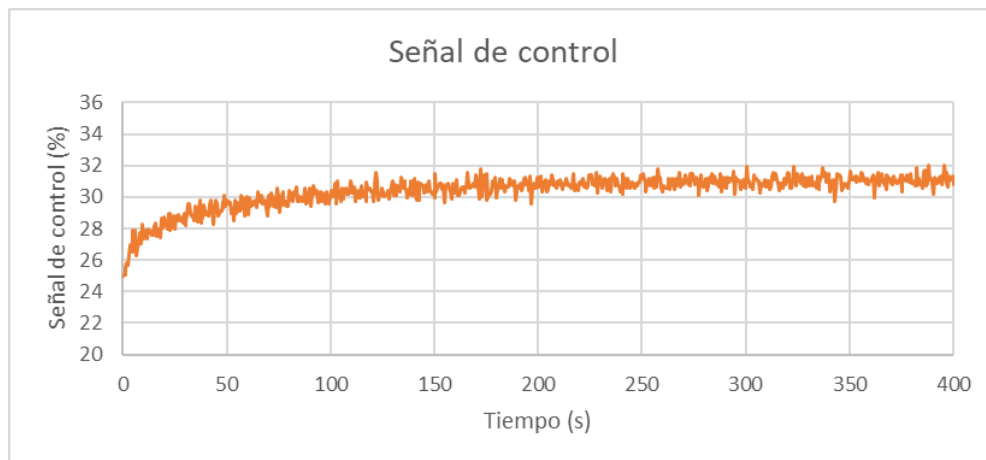


Figura 88 - Evolución señal de control ante perturbación con controlador PID con filtro (Lambda Tuning)

Gracias al filtro de paso bajo en el término derivativo la señal de control y la altura del depósito izquierdo presentan una oscilación mucho menor. Al haberse especificado la misma constante de tiempo que en el caso anterior ambos controladores presentan la misma velocidad en transitorio.

## 7.2.4 Sintonía Lambda Tuning PI

Mediante el programa “Loop-Pro” se realiza la sintonía de un controlador PI aplicando el método “Lambda Tuning”. Fijando un tiempo de respuesta del sistema en lazo cerrado de 0,30415 min los parámetros obtenidos son los siguientes:

- $K_p = 1.77$  (cm/%).
- $T_i = 1.271$  (min).
- $T_d = 0$  (min).
- $\text{Alfa} = 0$ .

A continuación, se introducen los parámetros del controlador en el laboratorio y se experimenta realizando un cambio de referencia a 15 cm desde el punto de equilibrio (Figura 89).

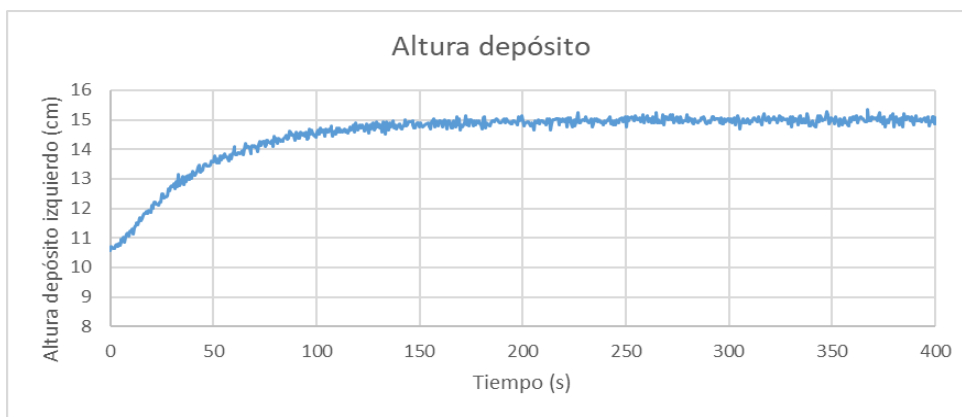


Figura 89 - Evolución altura depósito ante salto en la referencia con controlador PI (Lambda Tuning)

Se aprecia como debido a no poseer acción derivativa el sistema no amplifica el ruido, pero su transitorio es más lento. La evolución temporal de la señal de control es la siguiente (Figura 90):



Figura 90 - Evolución señal de control ante salto en la referencia con controlador PI (Lambda Tuning)

A continuación, se vuelve al punto de equilibrio y se realiza una perturbación consistente en incrementar la apertura de la válvula de salida hasta el 50%. La evolución de la altura del depósito es la siguiente (Figura 91).

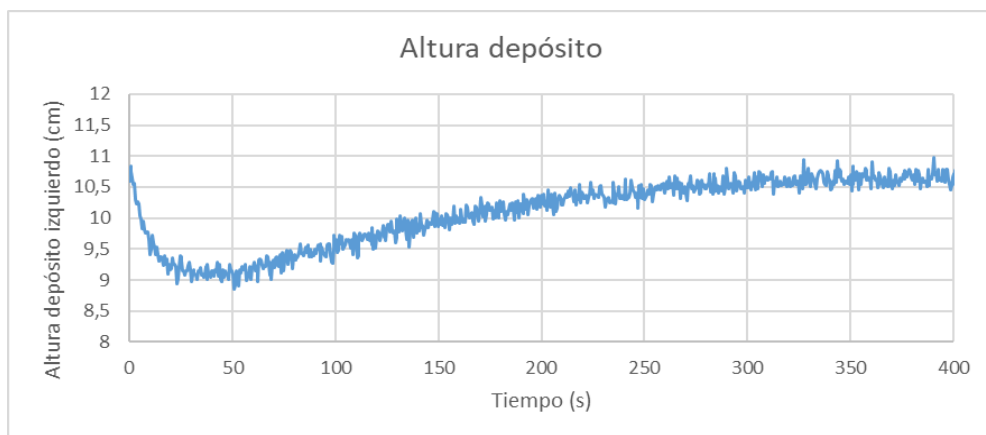


Figura 91 Evolución altura depósito ante perturbación con controlador PI (Lambda Tuning)

La señal de control de este experimento es la siguiente (Figura 92):

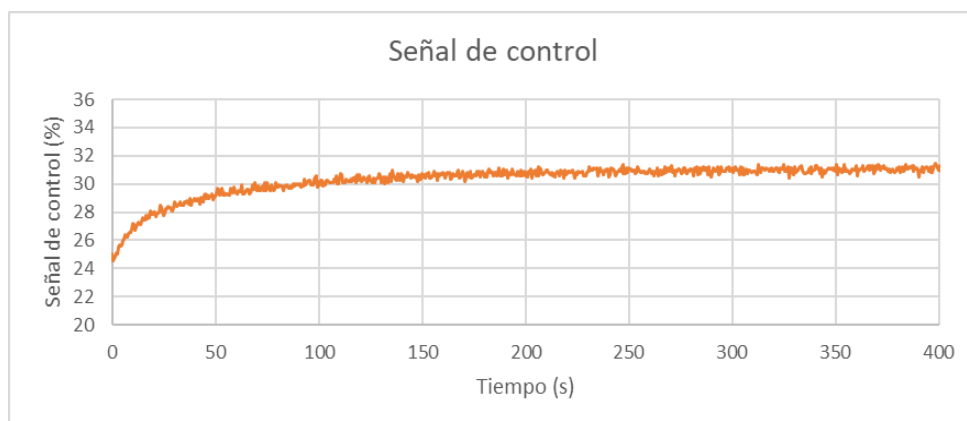


Figura 92 - Evolución señal de control ante perturbación con controlador PI (Lambda Tuning)

Frente a una perturbación las especificaciones obtenidas son las mismas que para un salto en la referencia.

### 7.2.5 Sintonía López PID

Se utiliza la tabla de sintonía de López para obtener los parámetros del controlador PID. Esta sintonización, basada en un modelo de primer orden con retardo, tiene como objetivo un correcto rechazo de las perturbaciones. El controlador obtenido carece de filtro. La tabla de sintonía a aplicar es la siguiente (Figura 93):

Tablas de Sintonía de López (III)			
Regulador PID			
Criterio	Proporcional	Integral	Derivativo
<b>MIAE</b>	a=1.435 b=-0.921	a=0.878 b=-0.749	a=0.482 b=1.137
<b>MISE</b>	a=1.495 b=-0.945	a=1.101 b=-0.771	a=0.560 b=1.006
<b>MITAE</b>	a=1.357 b=-0.947	a=0.842 b=-0.738	a=0.381 b=0.995

$$K_p K = a \left( \frac{d}{\tau} \right)^b$$

$$\frac{\tau}{T_i} = a \left( \frac{d}{\tau} \right)^b$$

$$\frac{T_d}{\tau} = a \left( \frac{d}{\tau} \right)^b$$

Válidas para procesos monótonos con  $d < \tau$

Figura 93 - Tabla de sintonía de Lopez



Se aplica el criterio "MIAE", el cual busca minimizar la integral del error absoluto. Los parámetros obtenidos tras aplicar las fórmulas son:

- $K_p = 21.13$  (cm/%).
- $T_i = 0.17$  (min).
- $T_d = 0.0234$ (min).
- Alfa = 0.

A continuación, se introducen los parámetros del controlador en el laboratorio y se experimenta realizando un salto en la referencia a 15 cm desde el punto de equilibrio (Figura 94).

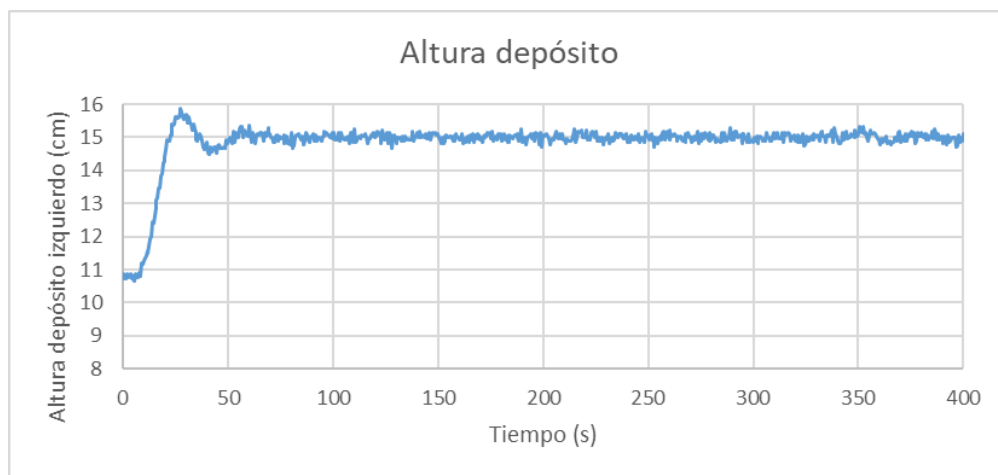


Figura 94 - Evolución altura depósito ante salto en la referencia con controlador PID (López)

La señal de control de este experimento es la siguiente (Figura 95):

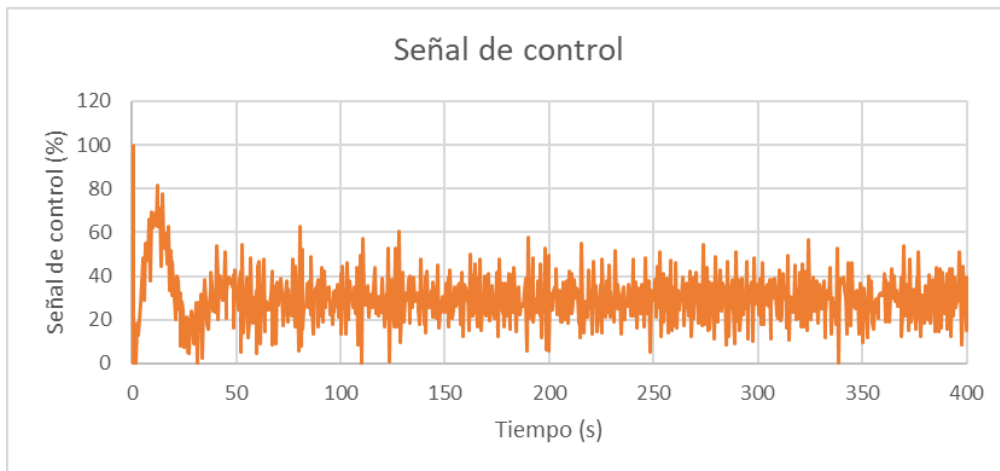


Figura 95 - Evolución señal de control ante salto en la referencia con controlador PID (López)

A continuación, se vuelve al punto de equilibrio y se realiza una perturbación consistente en incrementar la apertura de la válvula de salida hasta el 50%. La evolución de la altura del depósito es la siguiente (Figura 96):

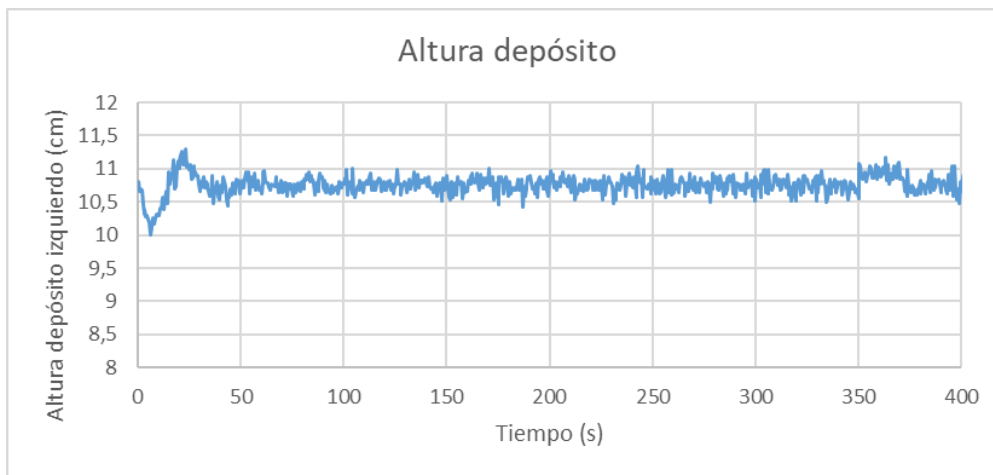


Figura 96 - Evolución altura depósito ante perturbación con controlador PID (López)

Siendo la señal de control aplicada en el experimento la siguiente (Figura 97):

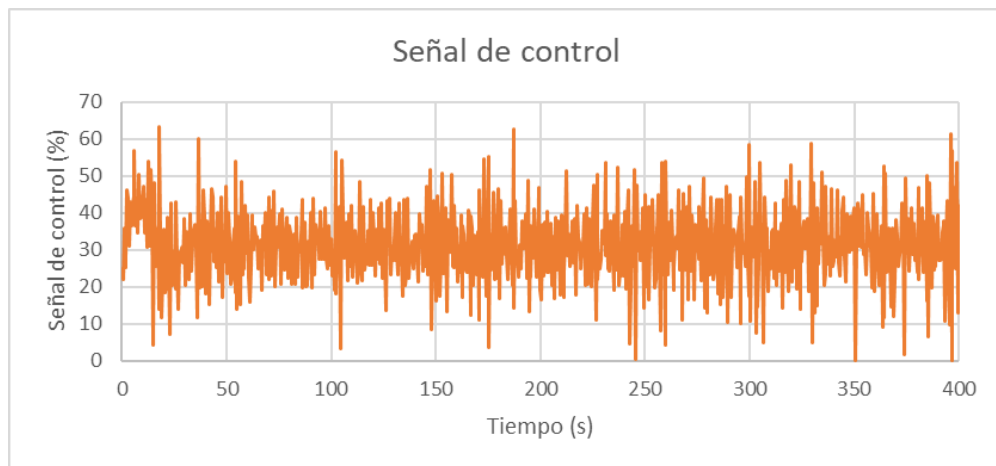


Figura 97 - Evolución señal de control ante perturbación con controlador PID (López)

Al no incorporarse un filtro en el término derivativo y poseer una constante proporcional considerablemente mayor que el resto de los controladores, la oscilación debida al ruido de medida es considerablemente más notable. No obstante, el sistema en lazo cerrado es considerablemente más rápido (al tener una mayor ganancia) y la sintonización logra su objetivo: el sistema responde realmente bien ante una perturbación.

### 7.3 Controlador Todo/Nada con histéresis

El control Todo/Nada con histéresis es una tipología de control sencilla de implantar ampliamente utilizado en sistemas de regulación de temperatura de calderas o aires acondicionados. Esta tipología puede ayudar a conocer el comportamiento de dicho tipo de sistemas.

Ejemplo: Para una apertura de la válvula de salida del 50%, una referencia de 15 cm y un ancho de banda de 6 cm, una potencia de la bomba en ON del 85 % y una potencia de la bomba en OFF del 15% se obtiene la siguiente evolución de la altura del depósito (Figura 98).

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

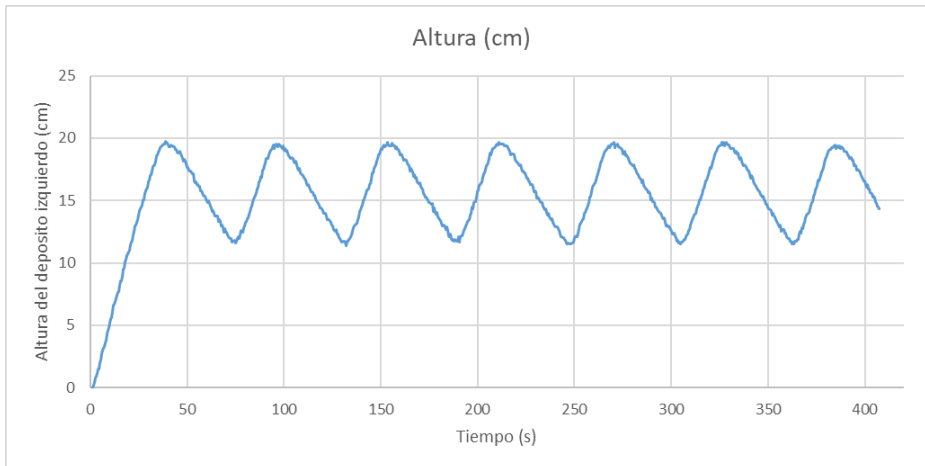


Figura 98 - Evolución altura del depósito con control On/Off

Se produce la oscilación característica del tipo de control Todo/Nada y es posible distinguir cómo la variación en la señal de control se produce cuando la variable controlada alcanza el límite superior o inferior.

La señal de control alterna entre el valor máximo y mínimo (Figura 99).

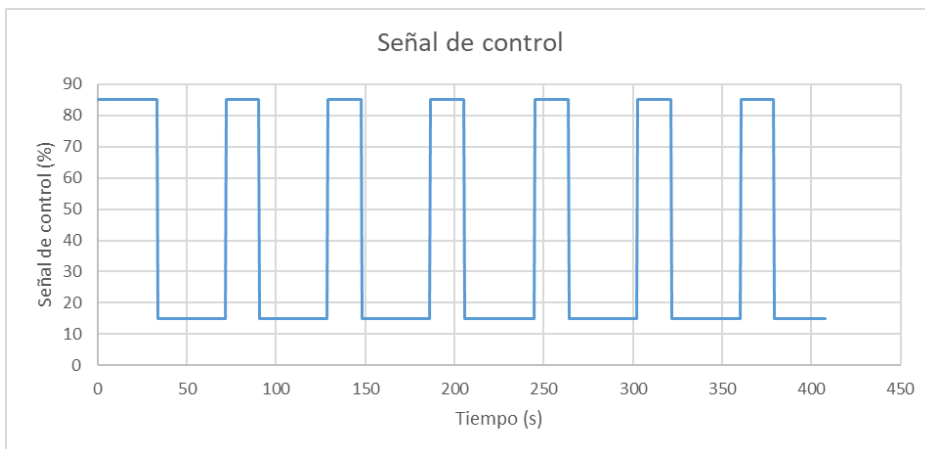


Figura 99 - Evolución señal de control con control On/Off

# CAPÍTULO 8: Conclusiones y trabajo futuro

---

## 8.1 Conclusiones

El objetivo principal del proyecto era la creación de un laboratorio virtual que modele el montaje de dos tanques comunicantes para que los estudiantes pudieran realizar una práctica previa al uso del montaje real. En vista del resultado del proceso de validación puede afirmarse que dicho laboratorio ha sido creado satisfactoriamente con las funcionalidades buscadas.

Su diseño se ha basado en la obtención del modelo del fenómeno a estudio, la programación del laboratorio virtual, su validación, la optimización del programa, y la experimentación final con la cual dejar constancia de ejemplos prácticos. No se han presentado inconvenientes remarcables ni cambios de rumbo importantes en la línea de desarrollo.

La simulación creada no presenta unos valores idénticos a los de la planta real, de la misma manera que dos plantas físicas tampoco se corresponderían exactamente. No obstante, la dinámica es análoga y la respuesta del sistema en ambos casos ante el mismo tipo de entrada presenta la misma forma.

En su elaboración se han empleado conocimientos de Fluidomecánica, Fundamentos de Automática y Fundamentos de Informática y ha sido necesario ahondar en el uso HTML, CSS y JavaScript.

En cuanto al entorno de desarrollo, Easy Java/JavaScript Simulations posee partes de su documentación desactualizadas o inaccesibles y se debería aumentar el número de funcionalidades disponibles. La traducción del entorno al español no es completa y se hace necesario trabajar en inglés, o al menos, conocer los nombres que ciertos conceptos de desarrollo de aplicaciones reciben en dicho idioma.

La creación de una aplicación sigue requiriendo grandes dosis de programación, lo que centra su público en aquellos que ya poseen un cierto nivel del diseño web. Se echa en falta la revisión de la sintaxis del código en tiempo real tal y como es habitual en un entorno de desarrollo estándar.

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

El empaquetamiento de la simulación genera una serie de archivos y estructuras adicionales superfluas que es recomendable eliminar. Es necesario mantener todos los archivos que lo componen bajo el mismo directorio.

Pese a sus defectos, se trata de un programa de código abierto muy interesante y que simplifica enormemente el desarrollo de este tipo de proyectos. Además, el continuo desarrollo del entorno posibilita la subsanación de los puntos anteriormente mencionados a corto-medio plazo.

## 8.2 Trabajo futuro

En cuanto a las perspectivas de trabajo futuro, la línea natural de continuación del presente trabajo, además de posibles mejoras en la interfaz, consiste en la incorporación de un servidor RIP en la planta real que permita modificar el laboratorio virtual para convertirlo en un laboratorio remoto.

Un laboratorio remoto es “una herramienta tecnológica, compuesta por software y hardware, que permite a docentes y estudiantes, realizar, a través de internet, sus prácticas como si estuvieran en un laboratorio tradicional”. En vez de mostrarse los resultados de una simulación, el usuario operaría telemática sobre la planta real y observaría su evolución mediante una cámara web. Adicionalmente, los valores obtenidos por los instrumentos de medida se mostrarían en la interfaz gráfica del laboratorio. Puede encontrarse un ejemplo de un laboratorio remoto consistente en el estudio del balanceo de una bola sobre una placa móvil (Figura 100).



Figura 100 - Ejemplo laboratorio remoto

El servidor RIP (Remote Interoperability Protocol), es el encargado de realizar las comunicaciones entre la planta real y el laboratorio remoto, lo que lo hace imprescindible en esta línea de trabajo. Este servidor se instalaría en el

## Laboratorio virtual de una planta piloto de dos tanques comunicantes

ordenador situado en las inmediaciones de la planta y debería mantenerse activo de forma continua en el periodo de tiempo en el que se desee que el laboratorio remoto sea operable. El esquema de comunicaciones del laboratorio virtual es el siguiente (Figura 101).

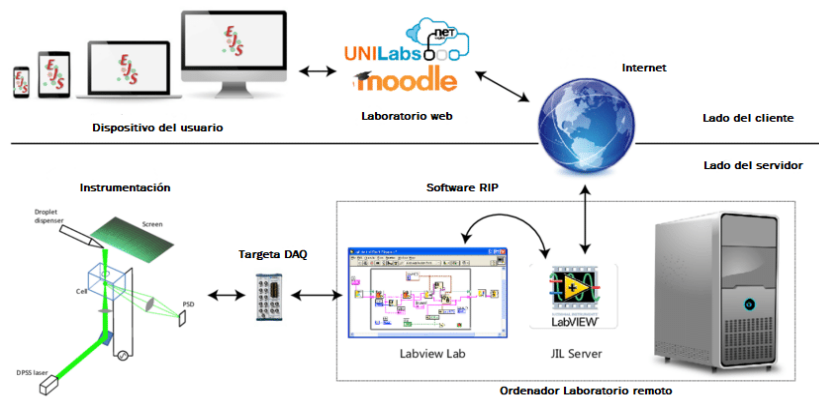


Figura 101 - Esquema comunicaciones servidor RIP

Una vez terminados tanto el laboratorio virtual como el laboratorio remoto sería interesante incluir ambos en el proyecto UNILabs (Figura 102). Esta iniciativa consiste en “una red de laboratorios de automática compartidos entre diferentes universidades españolas e internacionales, liderada por la UNED”.

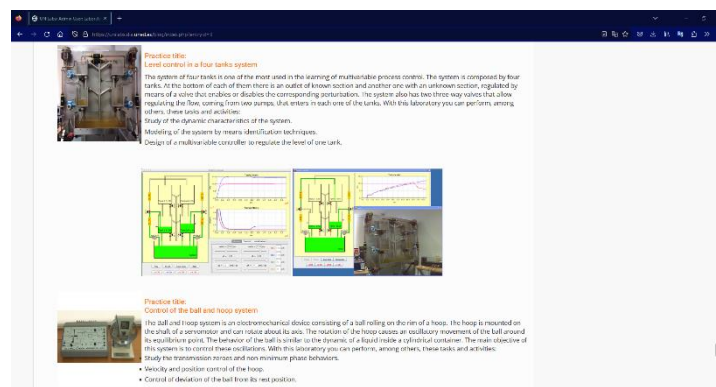


Figura 102 - Repositorio de UNILabs

Este repositorio permite a cualquier persona con interés en la materia experimentar libremente con cualquiera de las simulaciones disponibles.

La inclusión del software aquí propuesto en el programa UNILabs apuesta por una formación libre, inclusiva y de calidad, con la intención de apoyar al

Objetivo de Desarrollo Sostenible (ODS) número 4 (Figura 103). El cual aboga por “Garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida para todos”.



Figura 103 - Objetivos de desarrollo sostenible

Los objetivos de desarrollo sostenibles (ODS) son “17 objetivos globales interconectados para ser un plan para lograr un futuro mejor y más sostenible para todos”. Fueron establecidos en 2015 por la Asamblea General de las Naciones Unidas (AG-ONU) y se encuentran incluidos dentro de lo que se conoce como agenda 2030 (medidas y objetivos que se desean alcanzar para dicho año).



## CAPÍTULO 9: Bibliografía

Åström, K. J., & Murray, R. (2021). *ingenieros, Sistemas retroalimentados: introducción para científicos e* (2 ed.). Princeton University Press.

Canche, M. X. (24 de Agosto de 2020). *Algoritmo Box-Müller*. Obtenido de <https://github.com/xaviercanche/AlgoritmoBoxMuller>

Cooper, D. (2020). *Control práctico de procesos usando el software Loop-Pro. Control Station*. Obtenido de [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjnj5un8oKAAxVsVqQEHTXGA40QFnoECAkQAQ&url=https://www.comet.eng.unipr.it/~miccio/FDCPC\\_teaching\\_aids/Practical\\_Process\\_Control\\_Textbook\\_20060612.pdf&usq=AOvVaw2GiywKOav\\_7KmZ6Aw](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjnj5un8oKAAxVsVqQEHTXGA40QFnoECAkQAQ&url=https://www.comet.eng.unipr.it/~miccio/FDCPC_teaching_aids/Practical_Process_Control_Textbook_20060612.pdf&usq=AOvVaw2GiywKOav_7KmZ6Aw)

Esquembre, F. (Septiembre de 2013). *Easy Java Simulations Wiki*. Obtenido de <https://www.um.es/fem/EjsWiki/>

Fabregas, E., & Dormido Canto, S. (18 de Octubre de 2017). *Laboratorio Virtual y remoto de bola y sistema de placa*. Obtenido de [https://laris.univ-angers.fr/\\_resource/IFAC2017/IFAC\\_Paper\\_777.pdf](https://laris.univ-angers.fr/_resource/IFAC2017/IFAC_Paper_777.pdf)

García, Á. F. (Diciembre de 2010). *Vasos comunicantes*. Obtenido de Física con ordenador: <http://www.sc.ehu.es/sbweb/fisica/fluidos/dinamica/vasos/vasos.htm>

INCIBE. (s.f.). *UNILabs*. Obtenido de <https://www.incibe.es/incibe-cert/laboratorios/unilabs>

*Is it possible to write data to file using only JavaScript?* (mayo de 2017). Obtenido de Stack overflow: <https://stackoverflow.com/questions/21012580/is-it-possible-to-write-data-to-file-using-only-javascript>

MDN Web Docs. (s.f.). *Documentación de HTML, CSS y JavaScript*. Obtenido de Recursos para desarrolladores: <https://developer.mozilla.org/es/>

*Laboratorio virtual de una planta piloto de dos tanques comunicantes*

*Método de Box-Muller.* (7 de mayo de 2023). Obtenido de Wikipedia:  
[https://es.wikipedia.org/wiki/M%C3%A9todo\\_de\\_Box-Muller](https://es.wikipedia.org/wiki/M%C3%A9todo_de_Box-Muller)

*Método de Runge-Kutta.* (21 de abril de 2023). Obtenido de Wikipedia:  
[https://es.wikipedia.org/wiki/M%C3%A9todo\\_de\\_Runge-Kutta](https://es.wikipedia.org/wiki/M%C3%A9todo_de_Runge-Kutta)

*Métodos numéricos para ecuaciones diferenciales ordinarias.* (30 de noviembre de 2022). Obtenido de [https://es.wikipedia.org/wiki/M%C3%A9todos\\_num%C3%A9ricos\\_para\\_ecuaciones\\_diferenciales\\_ordinarias](https://es.wikipedia.org/wiki/M%C3%A9todos_num%C3%A9ricos_para_ecuaciones_diferenciales_ordinarias)

Moya, J. L. (13 de octubre de 2022). *¿Qué es un laboratorio remoto?* Obtenido de Labs Land: <https://labsland.com/blog/es/2022/10/13/que-es-un-laboratorio-remoto/>

Naciones Unidas. (25 de septiembre de 2015). *Objetivos de desarrollo sostenible.* Obtenido de <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>

Obando, L. F. (6 de febrero de 2022). *Modelo matemático de sistemas - Sistemas de control.* Obtenido de <https://dademuch.com/2022/02/06/modelo-matematico-de-sistemas-sistemas-de-control/>

*Objetivos de Desarrollo Sostenible.* (13 de junio de 2023). Obtenido de Wikipedia:  
[https://es.wikipedia.org/wiki/Objetivos\\_de\\_Desarrollo\\_Sostenible](https://es.wikipedia.org/wiki/Objetivos_de_Desarrollo_Sostenible)

Ogata, K. (1996). *Sistemas de control en tiempo discreto* (2 ed.). Pearson.

Ogata, K. (2010). *Ingeniería de control moderna* (5 ed.). Pearson.

Pardo Martín, C. (10 de diciembre de 2018). *Controlador PID.* Obtenido de Picuino: <https://www.picuino.com/es/control-pid.html>

Prada, C. d. (marzo de 2007). *Validación de modelos.* Obtenido de <https://www.eii.uva.es/~prada/Validacion.pdf>

*Laboratorio virtual de una planta piloto de dos tanques comunicantes*

RAE. (s.f.). *Diccionario de la lengua española*. Obtenido de <https://dle.rae.es/similar>

Santos, C. R. (2020). *Simulación dinámica de una planta de intercambio de calor [Tesis de pregrado]*. Universidad de Valladolid. Obtenido de <https://uvadoc.uva.es/handle/10324/40829>

Shannon, R. E. (2012). *Simulación de sistemas: Diseño, desarrollo e implantación*. Trillas.

*Teorema de Torricelli*. (s.f.). Obtenido de Wikipedia: [https://es.wikipedia.org/wiki/Teorema\\_de\\_Torricelli](https://es.wikipedia.org/wiki/Teorema_de_Torricelli)

Torre, L. d. (Enero de 2018). *Laboratorio remoto para levitación óptica de gotas cargadas*. Obtenido de [https://www.researchgate.net/publication/322728730\\_A\\_remote\\_laboratory\\_for\\_optical\\_levitation\\_of\\_charged\\_droplets](https://www.researchgate.net/publication/322728730_A_remote_laboratory_for_optical_levitation_of_charged_droplets)

UNED. (s.f.). *University Network of Interactive Laboratories (UNILabs)*. Obtenido de <https://unilabs.dia.uned.es/>

Villajulca, J. C. (junio de 2018). *Control ON/OFF o Todo/Nada*. Obtenido de [InstrumentaciónyControl.net](https://instrumentacionycontrol.net/): <https://instrumentacionycontrol.net/control-on-off-o-todo-nada/>



## CAPÍTULO 10: Anexos

El conjunto de archivos que compone el laboratorio virtual ha sido incluido en los anejos. La relación de archivos contenidos en estos es la siguiente:

- “Lab\_Tanques\_Comunicantes.ejss”: código fuente del programa. Para su apertura es necesario emplear el entorno de desarrollo EJS.
- “\_ejs\_library”: carpeta que contiene las librerías necesarias para la ejecución del laboratorio virtual.
- “fotos”: carpeta que contiene las fotos empleadas por el programa.
- “Creditos.html”: página web con los créditos del laboratorio.
- “Descripción.html” página web con la descripción del laboratorio.
- “Lab\_Tanques\_Comunicantes.xhtml”: página web que contiene el laboratorio virtual. Es el fichero principal.