



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



DEPARTAMENTO DE INGENIERÍA
DE SISTEMAS Y AUTOMÁTICA

**TRABAJO DE FIN DE MÁSTER EN AUTOMÁTICA E INFORMÁTICA
INDUSTRIAL**

**DESARROLLO DE UN SIMULADOR
HARDWARE-IN-THE-LOOP DE PROCESOS
DINÁMICOS MULTIVARIABLES BASADO EN
RASPBERRY PI**

AUTOR: MARCELO ALEJANDRO ALVAREZ LUNA

TUTOR: JUAN MANUEL HERRERO DURÁ

Curso Académico: 2018 - 19

AGRADECIMIENTOS

Agradezco a Dios por cuidar de mi en cada a momento, a mis padres por poner siempre el listón muy alto y empujarme a buscar la excelencia, a mi familia por el apoyo incondicional que siempre me lo han dado y a cada amigo que he hecho en mi caminar que me ha brindado un poco de fuerza y locura para continuar.

¡Gracias totales!

RESUMEN

El presente Trabajo de Fin de Máster tiene como objetivo el desarrollo de un sistema de simulación avanzado de sistemas multivariables que estará basado en la plataforma computacional de bajo coste Raspberry Pi y una tarjeta de adquisición de datos con cuatro entradas y cuatro salidas analógicas ya disponible.

La clave del proyecto es el desarrollo del núcleo programático de simulación dinámica mediante el lenguaje de programación científico Python.

La plataforma se encargará de realizar el procesamiento necesario para obtener las respuestas de las ecuaciones dinámicas que rigen el comportamiento de cada uno de los procesos a simular.

El usuario no solo dispondrá a través de una página web toda la información necesaria de las variables controladas y manipuladas de los procesos en cada momento, sino que podrá interactuar con los modelos y configurarlos según lo requiera.

Adicionalmente se realizará una prueba de concepto del simulador Hardware-In-The-Loop para analizar el control de uno de los procesos desarrollados a través del software LabVIEW y una tarjeta de adquisición de datos USB.

Cabe recalcar que el simulador HIL desarrollado en el presente trabajo a pesar de su bajo coste (comparado con otras plataformas más específicas), presenta prestaciones realmente aceptables que le permiten simular procesos con respuestas dinámicas de segundos con bastante fidelidad como se ha mostrado en la prueba de concepto llevada a cabo.

Palabras Clave: Hardware-In-The-Loop, Procesos Multivariables, HIL, Simulación, Tiempo Real, Multi-Loop, Raspberry Pi, Python.

DOCUMENTOS CONTENIDOS EN EL TRABAJO

FINAL DE MÁSTER

- Memoria del proyecto
- Manual de usuario
- Presupuesto

DESARROLLO DE UN SIMULADOR
HARDWARE-IN-THE-LOOP DE PROCESOS
DINÁMICOS MULTIVARIABLES BASADO EN
RASPBERRY PI

MEMORIA DEL PROYECTO

ÍNDICE CONTENIDO

| | | |
|-----------|---|-----------|
| 1. | OBJETO DEL TFM | 1 |
| 2. | INTRODUCCIÓN..... | 2 |
| 2.1. | Antecedentes | 2 |
| 2.2. | Motivación | 2 |
| 3. | DESCRIPCIÓN DE LA SOLUCIÓN HARDWARE-IN-THE-LOOP..... | 4 |
| 3.1. | Procesos Multivariables | 5 |
| 3.1.1. | Dos tanques acoplados | 5 |
| 3.1.2. | Cuatro tanques acoplados..... | 8 |
| 3.1.3. | Wood & Berry..... | 11 |
| 3.1.4. | Caldera-Turbina..... | 13 |
| 3.1.5. | Modelo personalizado..... | 15 |
| 3.2. | Hardware empleado..... | 16 |
| 3.2.1. | Raspberry Pi..... | 16 |
| 3.2.2. | TAD 4x4 | 17 |
| 3.3. | Núcleo Programático..... | 19 |
| 3.3.1. | Python | 20 |
| 3.3.2. | Plataforma Web | 23 |
| 4. | PRUEBAS DE CONCEPTO DEL SISTEMA DESARROLLADO..... | 28 |
| 4.1. | Pruebas Dos Tanques | 29 |
| 5. | CONCLUSIONES Y TRABAJOS FUTUROS..... | 33 |
| 6. | BIBLIOGRAFÍA..... | 34 |

ÍNDICE FIGURAS

| | |
|---|----|
| Figura 1. Modelo Hardware-In-The-Loop (Chi) | 2 |
| Figura 2. Fases de diseño HIL..... | 4 |
| Figura 3. Raspberry Pi modelo 3 B+..... | 4 |
| Figura 4. Modelo dos tanques acoplados..... | 5 |
| Figura 5. Diagrama Simulink dos tanques. | 7 |
| Figura 6. Bloque de ecuaciones para definir modelo h1 dos tanques. | 7 |
| Figura 7. Respuesta altura dos tanques mediante modelo Simulink. | 7 |
| Figura 8. Modelo cuatro tanques acoplados..... | 8 |
| Figura 9. Diagrama Simulink cuatro tanques..... | 10 |
| Figura 10. Respuesta alturas cuatro tanques modelo Simulink. | 10 |
| Figura 11. Columna de destilación de Wood&Berry. | 11 |
| Figura 12. Diagrama Simulink Wood & Berry. | 12 |
| Figura 13. Respuesta Wood & Berry modelo Simulink..... | 13 |
| Figura 14. Modelo Caldera-Turbina..... | 13 |
| Figura 15. Respuesta Caldera modelo Simulink. | 15 |
| Figura 16. Adaptación Debian. | 17 |
| Figura 17. Diagrama de bloques del funcionamiento TAD 4x4 | 17 |
| Figura 18. Diagrama de acondicionamiento señales..... | 19 |
| Figura 19. Esquema para desarrollo de plataforma HIL. | 19 |
| Figura 20. Núcleo de programación en Raspberry Pi. | 20 |
| Figura 21. Esquema de ficheros Python | 21 |
| Figura 22. Diagrama de flujo main.py..... | 22 |
| Figura 23. Diseño de la distribución de pantalla. | 23 |
| Figura 24. Barra de navegación. | 23 |
| Figura 25. Menú configuración proceso..... | 23 |
| Figura 26. Menú accionamiento de proceso | 24 |
| Figura 27. Información de las variables del proceso. | 24 |
| Figura 28. Sección de gráficas del proceso..... | 24 |
| Figura 29. Configuración modelo personalizado ejemplo 2x2 | 25 |
| Figura 30. Menú principal interfaz web..... | 25 |
| Figura 31. Interfaz del modelo de Dos Tanques Acoplados | 26 |
| Figura 32. Interfaz del modelo de Cuatro Tanques Acoplados..... | 26 |
| Figura 33. Interfaz del modelo Wood & Berry..... | 26 |
| Figura 34. Interfaz del modelo Térmico | 27 |
| Figura 35. Interfaz del modelo Personalizado. | 27 |
| Figura 36. Esquema de implementación de HIL. | 28 |
| Figura 37. PID VI (National Instruments, s.f.) | 29 |
| Figura 38. Cuadro de control de parámetros PID..... | 30 |
| Figura 39. HMI Proceso Dos Tanques LabVIEW..... | 31 |
| Figura 40. Toolbox File I/O de LabVIEW (National Instruments, s.f.) | 31 |
| Figura 41. Validación de resultados del Simulador vs. Simulink..... | 32 |

ÍNDICE TABLAS

| | |
|---|----|
| Tabla 1. Parámetros del modelo de dos tanques..... | 6 |
| Tabla 2. Punto de operación modelo dos tanques..... | 6 |
| Tabla 3. Rangos de operación modelo dos tanques..... | 6 |
| Tabla 4. Parámetros del modelo de cuatro tanques..... | 9 |
| Tabla 5. Punto de operación modelo cuatro tanques..... | 9 |
| Tabla 6. Rangos de operación modelo cuatro tanques..... | 9 |
| Tabla 7. Rangos de operación modelo Wood & Berry..... | 12 |
| Tabla 8. Parámetros del modelo de caldera-turbina..... | 14 |
| Tabla 9. Punto de operación modelo caldera-turbina..... | 14 |
| Tabla 10. Rangos de operación modelo Caldera-Turbina..... | 14 |
| Tabla 11. Características conversor A/D..... | 18 |
| Tabla 12. Características conversor D/A..... | 18 |
| Tabla 13. Librerías para el correcto uso del simulador..... | 20 |
| Tabla 14. Datos para corregir voltajes recibidos en la TAD..... | 29 |
| Tabla 15. Parámetros de los controladores PID..... | 30 |

1. OBJETO DEL TFM

El presente trabajo tiene como objeto el desarrollo de un sistema de simulación avanzada de sistemas multivariables *Hardware-In-The-Loop (HIL)*, capaz de procesar en tiempo real los modelos dinámicos de los procesos a ejecutar, para esta solución se utilizará el lenguaje de programación Python para el desarrollo de la solución dinámica ya que permite la integración de librerías de carácter científico que facilitan el manejo de los modelos matemáticos a resolver.

El sistema HIL se implementará sobre el ordenador de bajo costo Raspberry Pi el cual incorporará una tarjeta de adquisición de datos de cuatro entradas y cuatro salidas analógicas con la que se simularán las entradas y salidas de los procesos a simular.

Para la configuración y control de los modelos a simular se ha desarrollado una plataforma basada en un servidor web, en la cual se podrá configurar los parámetros de los modelos a discreción del usuario y observar su comportamiento mediante la representación de gráficas e indicadores que permitirán evaluar el correcto control de las plantas simuladas.

Es importante destacar que contar con un sistema HIL capaz de simular modelos en tiempo real de procesos industriales multivariables, que suelen ser muy costosos y de gran tamaño, permite al estudiante disponer de una plataforma potente y flexible con la que experimentar acerca del funcionamiento de los procesos simulados. Además, permite testear diferentes tipos de controladores clásicos, estableciendo control tipo PID multilazo o nuevas técnicas de control avanzadas.

2. INTRODUCCIÓN

2.1. Antecedentes

El sector industrial continuamente enfrenta nuevos retos, entre ellos que sus procesos sean más robustos y eficientes, esto con el objetivo no solo de garantizar un correcto consumo energético sino de lograr una producción acorde a lo requerido.

Hoy en día es muy común que antes de poner en marcha cualquier proceso industrial se utilicen herramientas informáticas para determinar su comportamiento y ensayar diferentes técnicas de control, esto tiene ventajas muy significativas, entre ellas evitar el costo de fallo, paradas innecesarias y averías ante una incorrecta sintonización de sus controladores.

Una de las metodologías que se ha utilizado en los últimos años para simular procesos industriales son los sistemas Hardware-In-The-Loop (HIL), este tipo de tecnología se ha empleado sobre todo en la industria del suministro de energía, aeronáutica y automotriz.

La simulación HIL es un tipo de simulación de control de procesos en el que la parte física del sistema en la cual se realizarán las pruebas está conectada virtualmente usando su modelo matemático. La planta (parte física) es una copia real en términos de modelo matemático que se encuentra dentro de la arquitectura del bucle (Mulka, 2016).

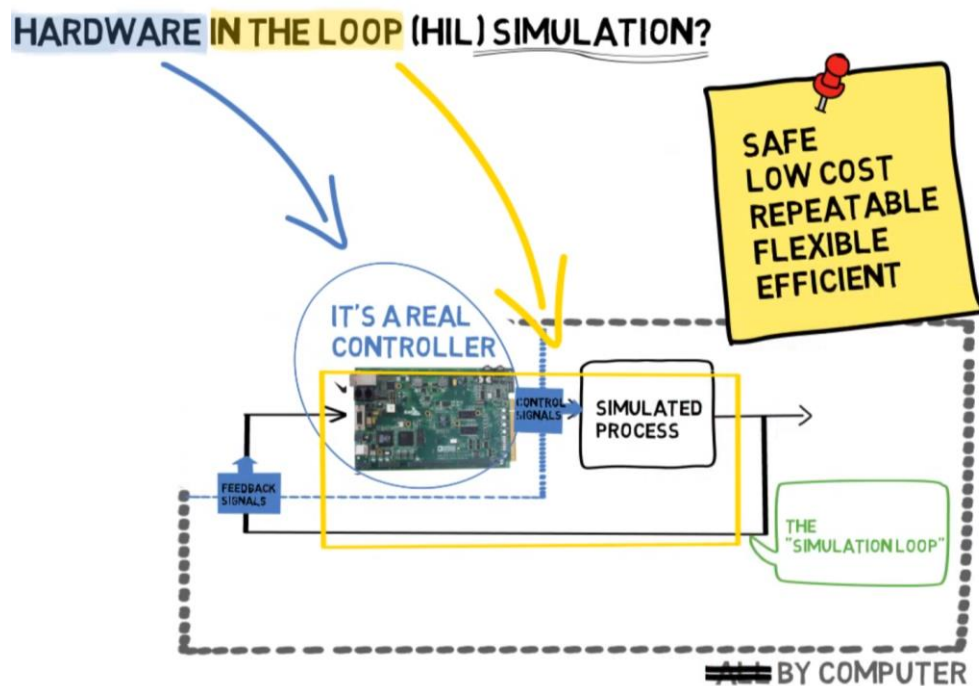


Figura 1. Modelo Hardware-In-The-Loop (Chi)

El modelo de la Figura 1 muestra una de las tantas configuraciones que puede tomar un simulador HIL y de esta manera se puede testear diferentes controladores de manera real sobre un proceso con dinámica compleja.

2.2. Motivación

Los centros de formación suelen poseer dentro sus instalaciones diferentes tipos de plantas para formar a sus estudiantes dentro de la teoría de control, con el objetivo que puedan tener una aproximación real a los procesos industriales.

El desarrollo de un simulador HIL es destacable a nivel educacional, sobre todo para los procesos multivariados ya que este tipo de tecnologías no se suelen encontrar disponibles por su alto costo y tamaño.

Generalmente productos de esta gama son muy costosos ya que implican un alto desarrollo de investigación en cuanto a software y hardware, además que suelen ser cerrados a ciertas plataformas y su flexibilidad en algunos casos es muy escasa.

Contar con un sistema que sea basado en código abierto y configurable en todos sus aspectos trae ventajas importantes para el usuario, ya que no solo le permite interactuar con diferentes tipos de plantas, sino que le lleva a descubrir una manera distinta de obtener y manipular los datos.

3. DESCRIPCIÓN DE LA SOLUCIÓN HARDWARE-IN-THE-LOOP

Para llegar a la solución del simulador HIL se ha precisado tres fases claves de diseño, en la Figura 2 se puede observar en primera instancia cuales han sido los modelos elegidos a realizar, luego se ha definido el hardware a utilizar y por último se ha especificado el software con el cual se desarrollará tanto las soluciones de los modelos matemáticos y la implementación de la plataforma.

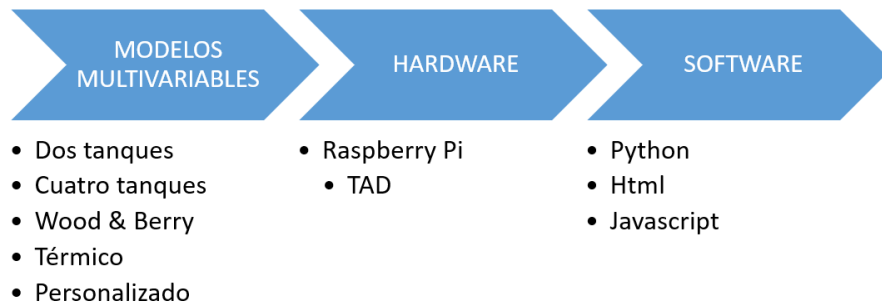


Figura 2. Fases de diseño HIL.

El hardware empleado para desarrollar el simulador HIL es el ordenador de bajo costo Raspberry Pi (Figura 3), este tipo de dispositivo es conocido por ser una computadora de placa única, significa exactamente cómo suena: es como una computadora de escritorio, una computadora portátil o un teléfono inteligente, pero construida en una sola placa de circuito impreso. Como la mayoría de las computadoras de una sola placa, la Raspberry Pi es pequeña, aproximadamente el mismo tamaño de una tarjeta de crédito, pero eso no significa que no sea potente: una Raspberry Pi puede hacer cualquier tarea que una computadora de mayor tamaño, aunque no necesariamente con la misma velocidad de procesamiento (Halfacree, 2018).



Figura 3. Raspberry Pi modelo 3 B+.

El lenguaje del núcleo programático con el cual se procesará los modelos matemáticos será Python en su versión 2.7, Python es en la actualidad uno de los lenguajes más utilizados para enseñar a programar además de uno de los más populares en la industria, compañías y organizaciones como Google, NASA, Disney, CERN y YouTube han escrito sus aplicaciones en este lenguaje debido a que es fácil de aprender, intuitivo y contiene un sinnúmero de librerías que facilitan al programador desarrollar todo tipo de aplicaciones (Tollervey, 2015).

3.1. Procesos Multivariables

Los procesos industriales se caracterizan por tener múltiples entradas y salidas que en la mayoría de casos interactúan entre sí, es por ello que el estudio de los procesos multivariables es de suma importancia para lograr no solo descubrir mejores maneras de controlarlos sino buscar la eficiencia en estos sistemas.

Un sistema de control multivariable (MIMO) o multilazo permite alcanzar el objetivo de mantener un conjunto de variables en un valor deseado, a diferencia del control de sistemas de una entrada y una salida (SISO), que sólo permite controlar una variable al tiempo (Lussón, Bausa, Barcaz, Beltrán, & Hernández, 2015).

Para el presente trabajo se han desarrollado cuatro procesos industriales multivariables y un proceso personalizado de hasta cuatro entradas por cuatro salidas, que permitirá al usuario en cada uno de ellos parametrizar los modelos según se requiera.

Es importante conocer los modelos matemáticos que definen el comportamiento dinámico de cada uno de los procesos a simular, es por ello que a continuación se estudiará el modelado de cada uno de ellos.

3.1.1. Dos tanques acoplados

El modelo de dos tanques acoplados es un sistema no lineal comúnmente estudiado en la teoría de control, como se puede ver en la Figura 4, el esquema muestra como la bomba envía flujo de líquido hacia los tanques 1 y 2 y la apertura de la válvula de tres vías permite el paso dependiendo del valor de gamma (γ).

El modelo en este caso será un proceso de dos entradas y dos salidas que tendrá como objetivo controlar la altura de los tanques 1 y 2 según se requiera.

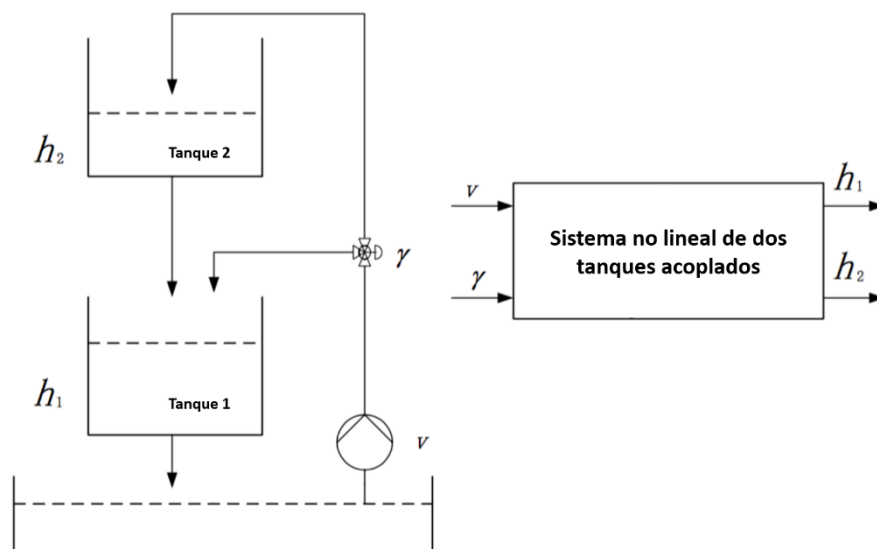


Figura 4. Modelo dos tanques acoplados.

A continuación, se han determinado las ecuaciones que definen el proceso no lineal a controlar:

$$A_1 \frac{dh_1}{dt} = -a_1 \sqrt{2gh_1} + a_2 \sqrt{2gh_2} + \gamma kv$$

$$A_2 \frac{dh_2}{dt} = a_2 \sqrt{2gh_2} + (1 - \gamma)kv$$

Tabla 1. Parámetros del modelo de dos tanques.

| Descripción | Parámetro | Valor |
|--------------------------------|-----------|-----------------------------|
| Sección del tanque 1 | A_1 | 28 [cm ²] |
| Sección del tanque 2 | A_2 | 32 [cm ²] |
| Sección de salida del tanque 1 | a_1 | 0.071 [cm ²] |
| Sección de salida del tanque 2 | a_2 | 0.057 [cm ²] |
| Parámetro de válvula | k | 3.33 [cm ² / Vs] |
| Gravedad | g | 981 [cm / s ²] |

Tabla 2. Punto de operación modelo dos tanques.

| Descripción | Parámetro | Valor |
|-----------------------------|------------|------------|
| Altura inicial del tanque 1 | h_1^0 | 5 [cm] |
| Altura inicial del tanque 2 | h_2^0 | 5 [cm] |
| Voltaje inicial de bomba | v^0 | 2.1181 [V] |
| Apertura de válvula | γ^0 | 0.1972 |

Tabla 3. Rangos de operación modelo dos tanques.

| Parámetros | Rango Operación | Rango TAD |
|----------------------------------|-----------------|------------|
| Altura del tanque 1 (h_1) | 0 - 9 [cm] | 0 - 9 [V] |
| Altura del tanque 2 (h_2) | 0 - 9 [cm] | 0 - 9 [V] |
| Voltaje de bomba (v) | 0 - 10 [V] | 0 - 10 [V] |
| Apertura de válvula (γ) | 0 - 1 [%] | 0 - 10 [V] |

Como se puede observar en la Tabla 1 se encuentran los valores con los que se ha definido el modelo para este ejemplo, en la Tabla 2 los valores para el punto de operación en el cual se ha trabajado y en la Tabla 3 los rangos en los que se operará tanto en el modelo como en la tarjeta de adquisición de datos.

Para lograr llegar a una aproximación del modelo y evaluar su comportamiento se ha hecho uso de la herramienta Simulink y se puede observar su diagrama de bloques en la Figura 5, además que mediante el bloque de funciones (Figura 6) se ha ingresado las ecuaciones que lo definen.

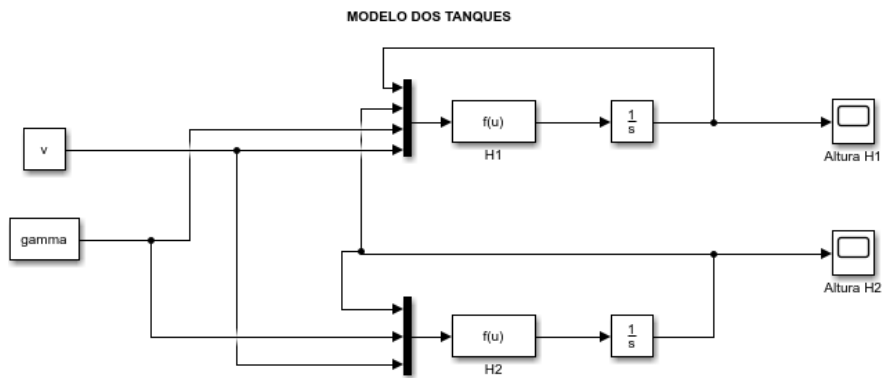


Figura 5. Diagrama Simulink dos tanques.

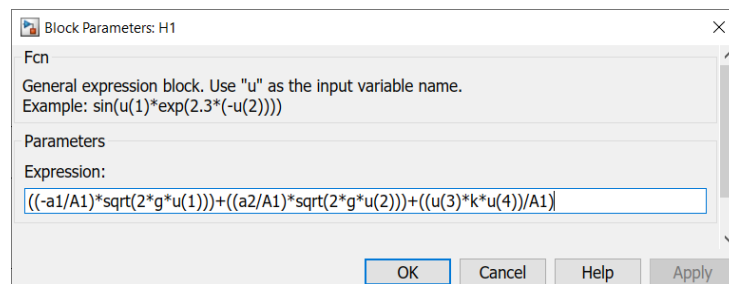


Figura 6. Bloque de ecuaciones para definir modelo h1 dos tanques.

Una vez determinado el modelo en Simulink es necesario observar su comportamiento (Figura 7) y de esta manera buscar que cuando se realice la simulación mediante el lenguaje de programación Python se obtengan valores muy similares a la respuesta entregada por Simulink, en este caso se observa su comportamiento ante una entrada escalón con valores de $v = 2.1181$ y $\gamma = 0.1972$ hacia el sistema.

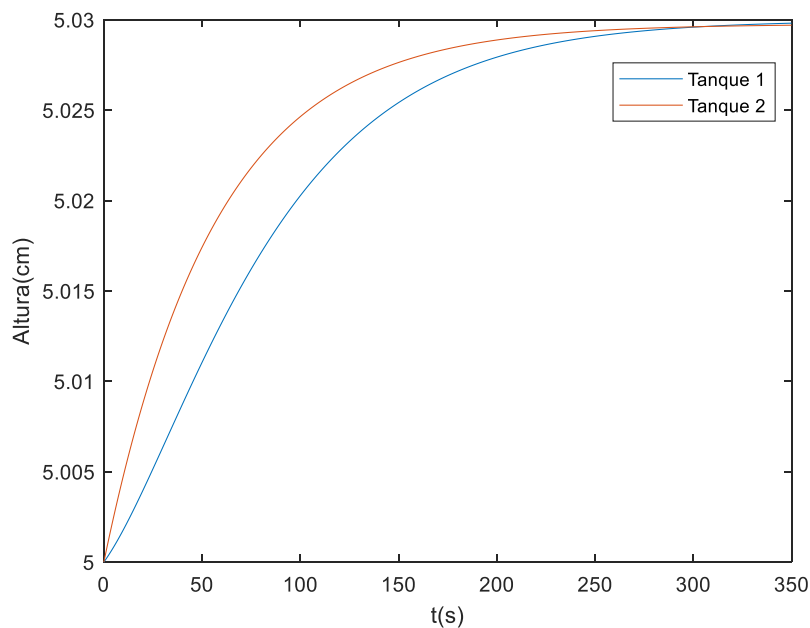


Figura 7. Respuesta altura dos tanques mediante modelo Simulink.

3.1.2. Cuatro tanques acoplados

El sistema de cuatro tanques es representativo de sistemas multivariables y análogo a procesos industriales los cuales en su mayoría están compuestos por subsistemas interconectados y dinámicas complejas con no linealidades en sus regiones de operación (Tlachi, Muñoz, & Maya, 2017).

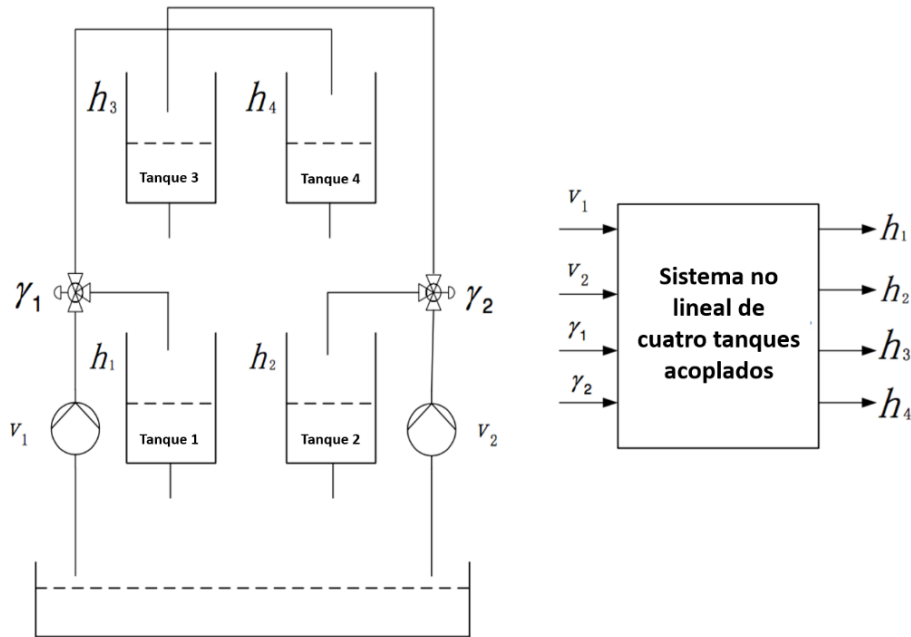


Figura 8. Modelo cuatro tanques acoplados.

En este caso se trata de un sistema de cuatro entradas y cuatro salidas, si las entradas γ_1 y γ_2 las dejamos constantes solo podemos controlar h_1 y h_2 , si las dejamos como entradas se pueden controlar las cuatro alturas.

A continuación, se han determinado las ecuaciones que definen el proceso no lineal a controlar:

$$\begin{aligned} \frac{dh_1}{dt} &= -\frac{a_1}{A_1} \sqrt{2gh_1} - \frac{a_3}{A_1} \sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1} v_1 \\ \frac{dh_2}{dt} &= -\frac{a_2}{A_2} \sqrt{2gh_2} + \frac{a_4}{A_2} \sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2} v_2 \\ \frac{dh_3}{dt} &= -\frac{a_3}{A_3} \sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3} v_2 \\ \frac{dh_4}{dt} &= -\frac{a_4}{A_4} \sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4} v_1 \end{aligned}$$

Como se puede observar en la Tabla 4 se encuentran los valores con los que se ha definido el modelo para este ejemplo, en la Tabla 5 los valores para el punto de operación en el cual se ha trabajado y en la Tabla 6 los rangos en los que se operará tanto en el modelo como en la tarjeta de adquisición de datos.

Tabla 4. Parámetros del modelo de cuatro tanques.

| Descripción | Parámetro | Valor |
|--------------------------------|------------|-----------------------------|
| Sección del tanque 1 | A_1 | 28 [cm ²] |
| Sección del tanque 2 | A_2 | 32 [cm ²] |
| Sección del tanque 3 | A_3 | 28 [cm ²] |
| Sección del tanque 4 | A_4 | 32 [cm ²] |
| Sección de salida del tanque 1 | a_1 | 0.071 [cm ²] |
| Sección de salida del tanque 2 | a_2 | 0.057 [cm ²] |
| Sección de salida del tanque 3 | a_3 | 0.071 [cm ²] |
| Sección de salida del tanque 4 | a_4 | 0.057 [cm ²] |
| Parámetro de válvula 1 | k_1 | 3.14 [cm ² / Vs] |
| Parámetro de válvula 2 | k_2 | 3.29 [cm ² / Vs] |
| Apertura de válvula 1 | γ_1 | 0.43 |
| Apertura de válvula 2 | γ_2 | 0.34 |
| Gravedad | g | 981 [cm / s ²] |

Tabla 5. Punto de operación modelo cuatro tanques.

| Descripción | Parámetro | Valor |
|-----------------------------|-----------|-----------|
| Altura inicial del tanque 1 | h_1^0 | 12.4 [cm] |
| Altura inicial del tanque 2 | h_2^0 | 13.2 [cm] |
| Altura inicial del tanque 3 | h_3^0 | 4.7 [cm] |
| Altura inicial del tanque 4 | h_4^0 | 4.9 [cm] |
| Voltaje inicial de bomba 1 | v_1^0 | 3.15 [V] |
| Voltaje inicial de bomba 2 | v_2^0 | 3.15 [V] |

Tabla 6. Rangos de operación modelo cuatro tanques.

| Parámetros | Rango Operación | Rango TAD |
|--------------------------|-----------------|------------|
| Altura del tanque 1 (h1) | 0 - 20 [cm] | 0 - 10 [V] |
| Altura del tanque 2 (h2) | 0 - 20 [cm] | 0 - 10 [V] |
| Altura del tanque 3 (h3) | 0 - 20 [cm] | 0 - 10 [V] |
| Altura del tanque 4 (h4) | 0 - 20 [cm] | 0 - 10 [V] |
| Voltaje de bomba 1 (v1) | 0 - 10 [V] | 0 - 10 [V] |
| Voltaje de bomba 2 (v2) | 0 - 10 [V] | 0 - 10 [V] |

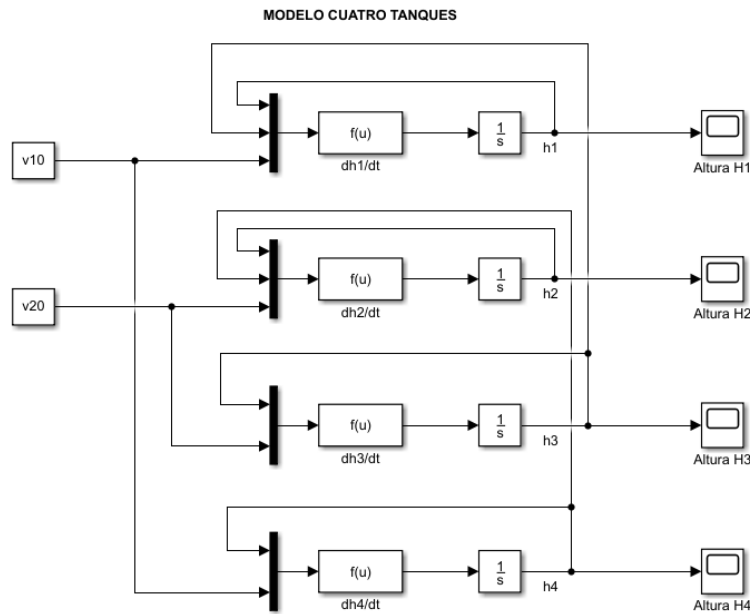


Figura 9. Diagrama Simulink cuatro tanques.

Al igual que el modelo de dos tanques se ha hecho uso de la herramienta Simulink para realizar una simulación que demuestre el comportamiento según los parámetros dados y se puede observar su diagrama de bloques en la Figura 9.

La respuesta del proceso determina las alturas en su punto de operación y de esta manera con los datos obtenidos se procederá a validar la salida del simulador HIL.

Es importante notar que para este caso en particular se ha dejado las entradas γ_1 y γ_2 como constantes, es decir, solo se podrá controlar las alturas de los tanques uno y dos aplicando cambios en los voltajes de las bombas.

En concreto la respuesta que presenta el modelo (Figura 10) muestra el resultado ante las entradas escalón de $v_1 = 3.15$ y $v_2 = 3.15$ respectivamente.

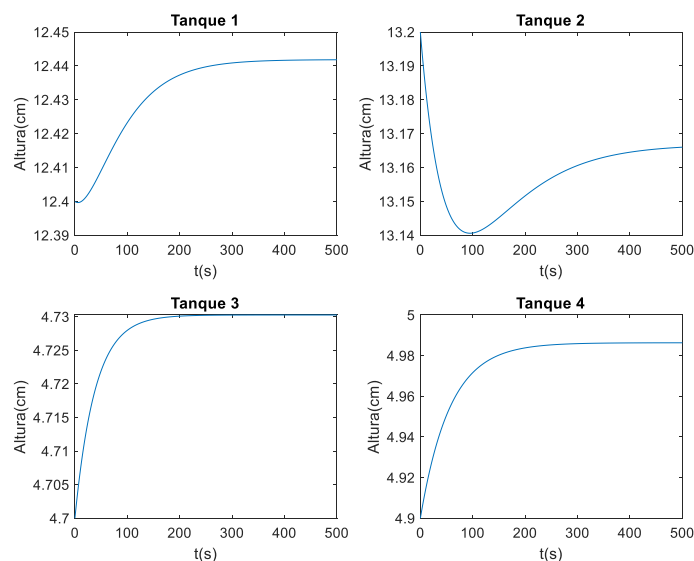


Figura 10. Respuesta alturas cuatro tanques modelo Simulink.

3.1.3. Wood & Berry

Los procesos de destilación son comúnmente estudiados ya que se usan frecuentemente en la industria química, el objetivo de estudiar estos procesos es mejorar el desempeño de los mismos ya que suelen utilizar grandes cantidades de energía.

La columna de destilación de Wood & Berry (Figura 11) se basa en una alimentación controlada y continua de la mezcla a separar. Una vez que la mezcla entra en la columna se produce la separación continua de los componentes, la fracción ligera sale por la parte superior de la columna y la fracción pesada deja la columna por el fondo (Montoya, 2012).

Las entradas al sistema son el reflujo (R) y el vapor (V), se encuentran en libras por minuto (lb/min), mientras que las salidas son el metanol destilado (xD) y el agua que ha quedado en el fondo (xB).

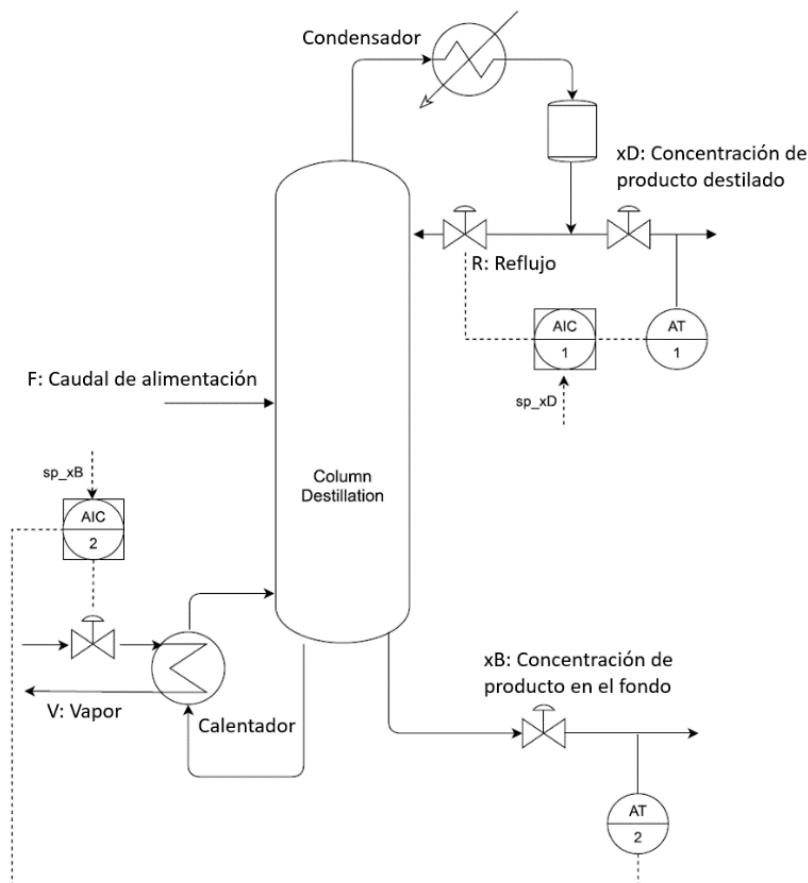


Figura 11. Columna de destilación de Wood&Berry.

El modelo que define el comportamiento del proceso Wood & Berry se encuentra a continuación y se lo ha obtenido parametrizando un punto de operación en concreto:

$$x_D = 0.96$$

$$x_B = 0.02$$

$$R = 1.71 \frac{lb}{min}$$

$$V = 2.45 \frac{lb}{min}$$

$$\begin{bmatrix} x_D \\ x_B \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s + 1} & \frac{-18.9e^{-3s}}{21s + 1} \\ \frac{6.6e^{-7s}}{10.9s + 1} & \frac{-19.4e^{-3s}}{14.4s + 1} \end{bmatrix} \begin{bmatrix} R \\ V \end{bmatrix}$$

Tabla 7. Rangos de operación modelo Wood & Berry.

| Parámetros | Rango Operación | Rango TAD |
|--|-----------------|------------|
| Concentración metanol destilado (xD) | 0 - 1 | 0 - 10 [V] |
| Concentración de agua en el fondo (xB) | 0 - 1 | 0 - 10 [V] |
| Caudal reflujo (R) | 0 - 5 [lb/min] | 0 - 10 [V] |
| Caudal vapor (V) | 0 - 5 [lb/min] | 0 - 10 [V] |

En la Tabla 7 se puede apreciar los rangos en los que se operará tanto en el modelo como en la tarjeta de adquisición de datos.

Con el objetivo de conocer la respuesta del sistema se ha hecho uso de la herramienta Simulink para simular la columna de destilación y conocer su respuesta (Figura 12), en esta ocasión se han usado los bloques de función de transferencia además del bloque Transport Delay el cual generará el retardo dado por el modelo.

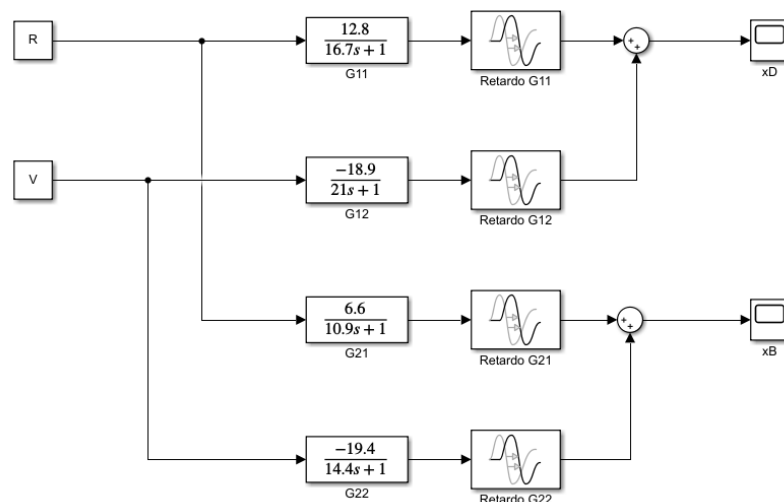


Figura 12. Diagrama Simulink Wood & Berry.

La Figura 13 muestra el comportamiento del sistema ante las entradas del reflujó $R = 0.07$ y vapor $V = 0.005$, esto con el objetivo de comparar los resultados que retornará el simulador HIL posteriormente.

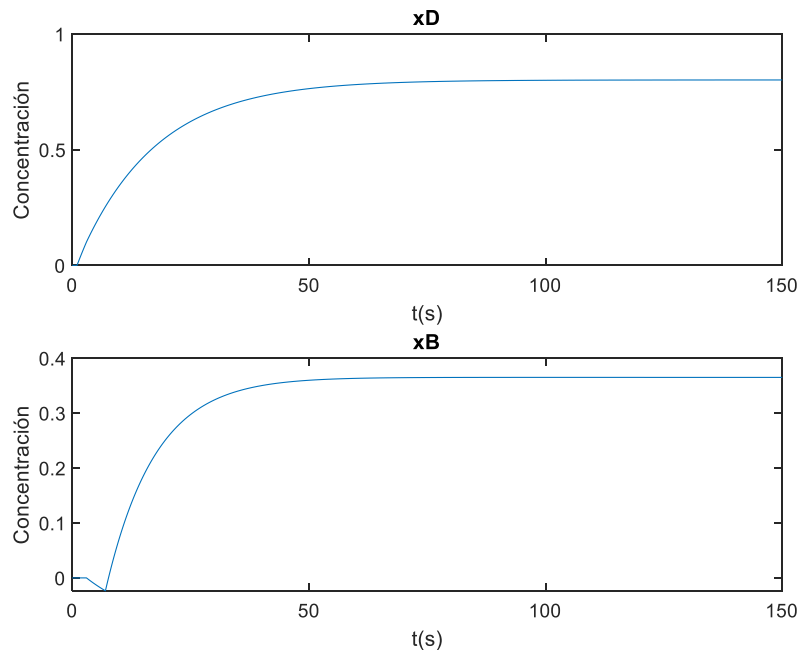


Figura 13. Respuesta Wood & Berry modelo Simulink.

3.1.4. Caldera-Turbina

El modelo de Caldera-Turbina es un sistema capaz de generar gran cantidad de vapor a presiones muy altas con el objetivo de mover una turbina y generar energía eléctrica, este tipo de sistemas y modelos dinámicos empezó a ser estudiado por Bell y Åström en la década de los setentas.

Como se puede observar en la Figura 14 el sistema multivariable cuenta con tres entradas y tres salidas siendo el objetivo principal la correcta producción de la potencia eléctrica.

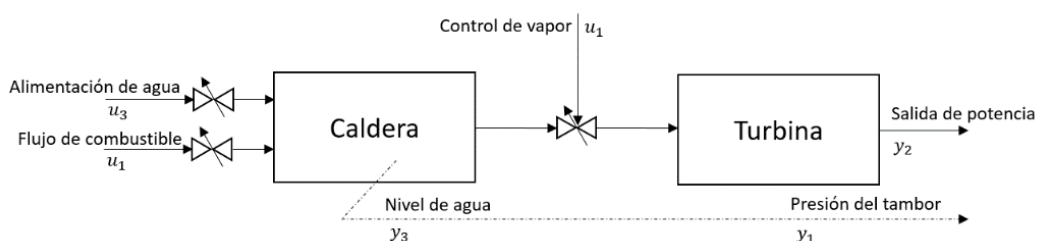


Figura 14. Modelo Caldera-Turbina.

A continuación, se define las ecuaciones que rigen el comportamiento del proceso, en la Tabla 8 se detalla los parámetros de medida y las unidades en las que se encuentra cada uno de ellos y en la Tabla 9 se encuentra el detalle del punto de operación con el cual se ha trabajado para este ejemplo.

$$\dot{x}_1 = -0.0018u_2x_1^{9/8} + 0.9u_1 - 0.15u_3$$

$$\dot{x}_2 = (0.073u_2 - 0.016)x_1^{9/8} - 0.1x_2$$

$$\dot{x}_3 = (141u_3 - (1.1u_2 - 0.19)x_1)/85$$

$$y_1 = x_1$$

$$y_2 = x_2$$

$$y_3 = 0.05(0.13073x_3 + 100a_{cs} + q_e/9 - 67.975)$$

Donde;

$$a_{cs} = \frac{(1 - 0.001538x_3)(0.8x_1 - 25.6)}{x_3(1.0394 - 0.0012304x_1)}$$

$$q_e = (0.854u_2 - 0.147)x_1 + 45.59u_1 - 2.514u_3 - 2.096$$

Tabla 8. Parámetros del modelo de caldera-turbina.

| Parámetro | Descripción | Unidades |
|-----------|----------------------------|--------------------|
| u_1 | Flujo combustible | % |
| u_2 | Control de vapor | % |
| u_3 | Alimentación agua | % |
| x_1 | Presión del tambor | kg/cm ² |
| x_2 | Salida de potencia | MW |
| x_3 | Densidad de flujo | kg/m ³ |
| y_3 | Nivel de agua en el tambor | m |

Tabla 9. Punto de operación modelo caldera-turbina.

| Parámetro | Valor |
|-----------|-------|
| x_1^0 | 108 |
| x_2^0 | 66.65 |
| x_3^0 | 428 |
| u_1^0 | 0.34 |
| u_2^0 | 0.69 |
| u_3^0 | 0.433 |
| y_3^0 | 0 |

Tabla 10. Rangos de operación modelo Caldera-Turbina.

| Parámetros | Rango Operación | Rango TAD |
|------------------------------|-------------------------------|------------|
| Presión del tambor (y_1) | 0 - 200 [kg/cm ²] | 0 - 10 [V] |
| Potencia generada (y_2) | 0 - 100 [MW] | 0 - 10 [V] |
| Nivel de agua (y_3) | 0 - 1 [m] | 0 - 10 [V] |
| Flujo combustible (u_1) | 0 - 1 [%] | 0 - 10 [V] |
| Control de vapor (u_2) | 0 - 1 [%] | 0 - 10 [V] |
| Alimentación agua (u_3) | 0 - 1 [%] | 0 - 10 [V] |

En la Tabla 10 se puede apreciar los rangos en los que se operará tanto en el modelo como en la tarjeta de adquisición de datos.

La respuesta que se ha obtenido de Simulink del modelo y teniendo como referencia las entradas definidas de u_1 , u_2 y u_3 en la Tabla 9 se muestra en la Figura 15.

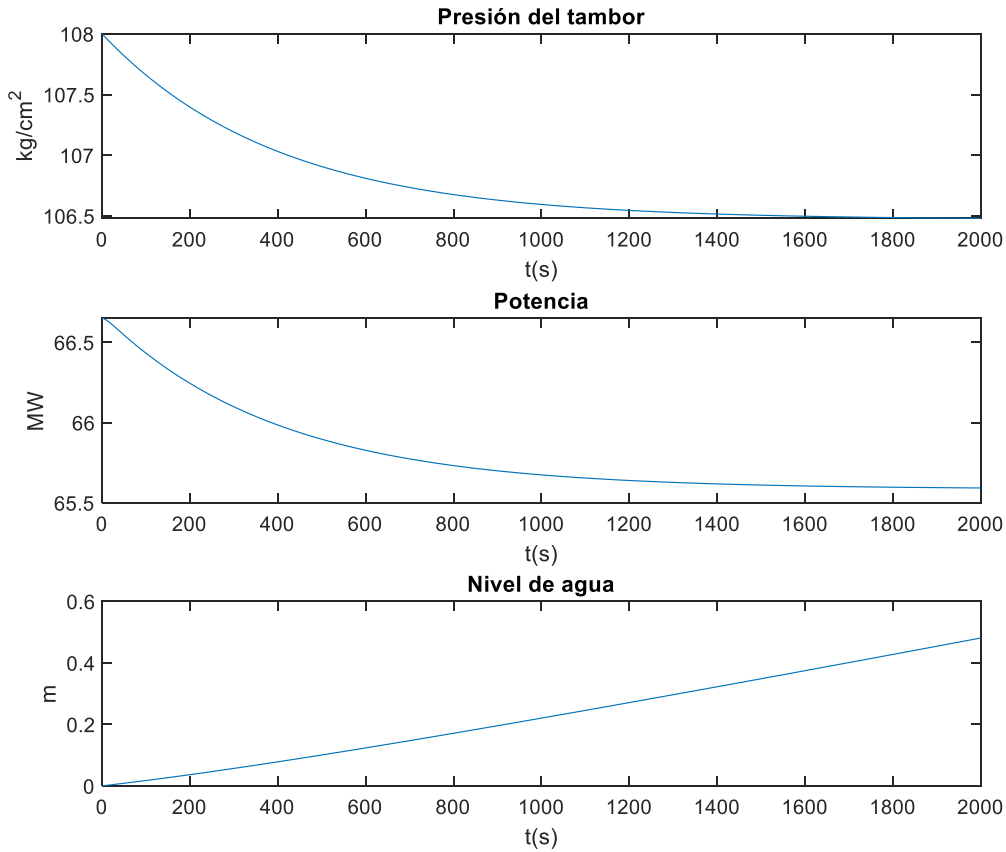


Figura 15. Respuesta Caldera modelo Simulink.

3.1.5. Modelo personalizado

Para el presente trabajo se ha propuesto realizar dentro del simulador HIL un modelo multivariable personalizado el cual tendrá una estructura de hasta cuatro entradas por cuatro salidas.

A continuación, se puede observar la configuración matricial que presenta este tipo de modelo.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} \\ G_{21} & G_{22} & G_{23} & G_{24} \\ G_{31} & G_{32} & G_{33} & G_{34} \\ G_{41} & G_{42} & G_{43} & G_{44} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

$$y_1 = G_{11}u_1 + G_{12}u_2 + G_{13}u_3 + G_{14}u_4$$

$$y_2 = G_{21}u_1 + G_{22}u_2 + G_{23}u_3 + G_{24}u_4$$

$$y_3 = G_{31}u_1 + G_{32}u_2 + G_{33}u_3 + G_{34}u_4$$

$$y_4 = G_{41}u_1 + G_{42}u_2 + G_{43}u_3 + G_{44}u_4$$

Cualquiera de las funciones puede anularse de manera que se puede anular una entrada y/o salida determinada reduciendo así la dimensión del sistema.

3.2. Hardware empleado

3.2.1. Raspberry Pi

Como se ha mencionado anteriormente para desarrollar el núcleo de programación se ha escogido el ordenador de placa reducida Raspberry Pi, la gran ventaja de utilizar este tipo de dispositivos no es solo su bajo costo, sino que su mínimo consumo energético, su reducido tamaño y su gran capacidad de procesamiento la hacen muy atractiva para alcanzar un sinfín de aplicaciones.

Algunas de las características más importantes en cuanto a sus periféricos y procesamiento se detallan a continuación:

- CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- RAM: 1GB LPDDR2 SDRAM
- Wi-Fi + Bluetooth: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE
- Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps)
- GPIO de 40 pines
- HDMI
- 4 puertos USB 2.0
- Puerto CSI para conectar una cámara.
- Puerto DSI para conectar una pantalla táctil
- Salida de audio estéreo y vídeo compuesto
- Micro-SD
- Power-over-Ethernet (PoE)
- Periféricos integrados:
 - 27 pines GPIO
 - UART
 - I2C
 - SPI (con 2 CS)
 - +3.3V
 - +5V
 - GND

El sistema operativo con el cual trabaja la Raspberry Pi es una versión adaptada de Debian llamado Raspbian (Figura 16), es de código abierto basado en una distribución GNU/Linux de

tipo Unix POSIX por lo tanto es multiplataforma, multiusuario y multitarea, además ha sido ampliamente adaptado para la enseñanza de la informática.



Figura 16. Adaptación Debian.

Como se ha podido observar en las principales características, la Raspberry Pi no cuenta con periféricos analógicos por lo que se ha montado una tarjeta de adquisición de datos con cuatro entradas y salidas analógicas ya disponible y que ha sido un trabajo previo a este TFM.

3.2.2. TAD 4x4

La tarjeta de adquisición de datos desarrollada previamente para este proyecto permite a la Raspberry Pi comunicar la respuesta de los procesos industriales simulados mediante cuatro entradas y cuatro salidas analógicas, esto quiere decir, que una vez procesadas las ecuaciones que definen las dinámicas de los sistemas, se obtendrá una salida en función del voltaje, este voltaje estará en el rango de +/- 10 V tanto para las entradas como para las salidas.

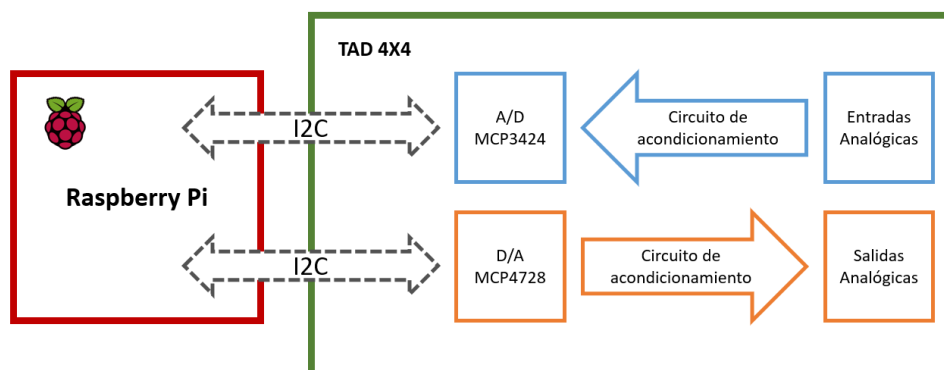


Figura 17. Diagrama de bloques del funcionamiento TAD 4x4

Los circuitos integrados que serán el núcleo de la conversión análogo digital y viceversa se pueden observar en la Figura 17, estos dos integrados se comunicarán a través del protocolo I2C con el que cuenta la Raspberry Pi para el envío y recepción de información.


Cabe destacar que la gran ventaja de utilizar este tipo de comunicación sobre todo para el desarrollo de dispositivos electrónicos es que permite comunicar varios nodos a través del mismo bus de comunicación, haciéndolo uno de los protocolos más usados en la actualidad.

A continuación, se presenta algunas de las características de los circuitos integrados que se han utilizado en la tarjeta de adquisición de datos:

Tabla 11. Características conversor A/D.

| MCP3424 Conversor Análogo Digital | |
|--|--|
| CARACTERÍSTICA | VALOR |
| Resolución | 18 bits |
| Número de canales | 4 |
| Voltaje de referencia | 2.048 V |
| Tipo de interfaz | I2C |
| Tasa de muestreo | 3.75 S/s, 15 S/s, 60 S/s, 240 S/s |
| Circuito integrado |  |

Tabla 12. Características conversor D/A.

| MCP4728 Conversor Digital Análogo | |
|--|---|
| CARACTERÍSTICA | VALOR |
| Resolución | 12 bits |
| Número de canales | 4 |
| Voltaje de referencia | 2.048 V 0.000V a 2.048V with Gain Setting = 1 0.000V a 4.096V with Gain Setting = 2 |
| Tipo de interfaz | Serial, I2C |
| Tasa de muestreo | 3.75 S/s, 15 S/s, 60 S/s, 240 S/s |
| Circuito integrado |  |

Si bien es cierto las características de los circuitos indicados en la Tabla 11 y la Tabla 12 muestran un voltaje de referencia de trabajo muy reducido, la TAD con la que se ha trabajado esta aplicación tiene entradas y salidas analógicas que se encuentran en el rango de +/-10V.

Para explicar el funcionamiento de estos circuitos es necesario observar con cuidado la Figura 18; el diagrama de bloques explica las ecuaciones que determinan el acondicionamiento tanto en la parte electrónica como en la del software para lograr dichos voltajes de trabajo.

Se debe tener en cuenta que dependiendo el dispositivo con el que se generará las señales de control será necesario una previa corrección de voltajes para no tener discrepancias con las ecuaciones que rigen los modelos.

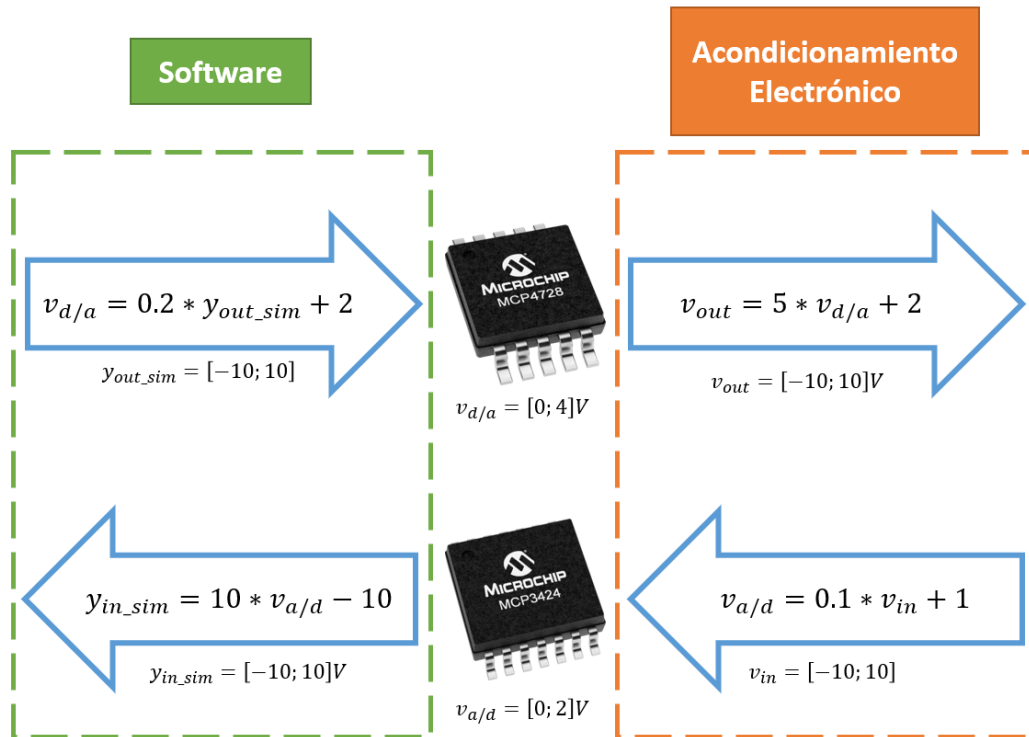


Figura 18. Diagrama de acondicionamiento señales.

3.3. Núcleo Programático

Una vez definidos los modelos dinámicos a desarrollar y seleccionado el hardware a utilizar se ha procedido a montar el esquema de programación que será usado para el desarrollo del simulador (Figura 19).

La plataforma cuenta con dos líneas claves de desarrollo, en primer lugar se encuentra el backend, encargado del procesamiento de los modelos matemáticos, la adquisición de datos y la puesta en marcha del servidor web y por otro lado el frontend que tiene como objetivo generar la interfaz para mostrar la información de los datos en tiempo real.

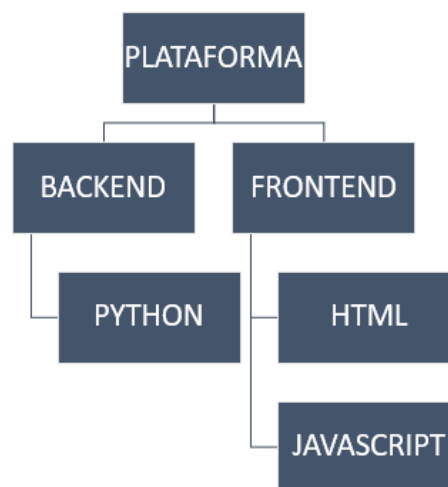


Figura 19. Esquema para desarrollo de plataforma HIL.

Al ser Python el lenguaje escogido para desarrollar el núcleo de desarrollo del sistema HIL en el siguiente apartado se explicará las librerías empleadas además de los ficheros que se han utilizado para lograr la puesta en marcha del proyecto.

3.3.1. Python

Al observar la Figura 20 destaca que el lenguaje Python será el núcleo de todo el procesamiento tanto para el servidor web, la simulación de los modelos y la adquisición de datos.

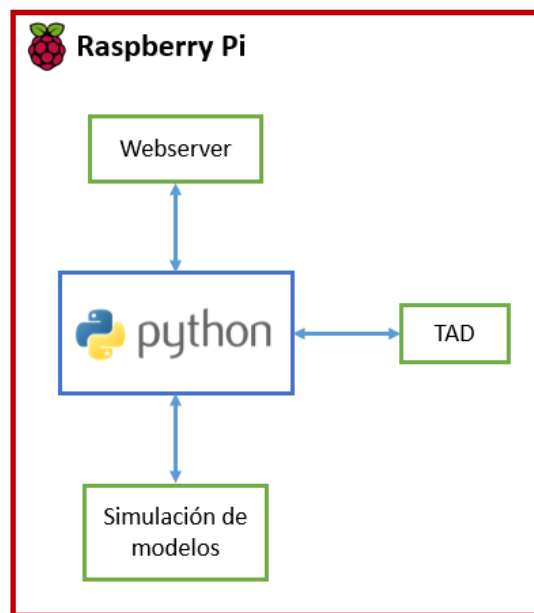


Figura 20. Núcleo de programación en Raspberry Pi.

Para lograr la simulación de los modelos de una manera correcta se han instalado ciertas librerías que facilitan el procesamiento de los modelos matemáticos, a continuación, se listan algunas de ellas y sus comandos para la instalación.

Tabla 13. Librerías para el correcto uso del simulador.

| LIBRERIA | COMANDO |
|--------------------------------|--|
| Numpy | sudo apt-get install python-numpy |
| Scipy | sudo apt-get install python-scipy |
| Matplotlib | sudo apt-get install python-matplotlib |
| Python Control Systems Library | pip install control |

Es importante tener en consideración que para algunas librerías como la de Python Control Systems Library se puede tener ciertos inconvenientes para instalarla directamente bajo el comando pip, por lo que para este caso se ha descargado desde su repositorio, compilado e instalado directamente sobre la Raspberry Pi de manera manual.

Antes de proceder a instalar cualquier librería es recomendable tener el sistema operativo actualizado, además es necesario tener ciertos paquetes de desarrollo previamente instalados. A continuación, se detalla el comando que ha sido utilizado para montar estas herramientas.

- `sudo apt-get install build-essential python-dev python-distlib python-setuptools python-pip python-wheel libzmq-dev libgdal-dev`

Con el sistema operativo actualizado y las librerías instaladas se ha esquematizado los ficheros que serán parte del funcionamiento del sistema HIL (Figura 21).

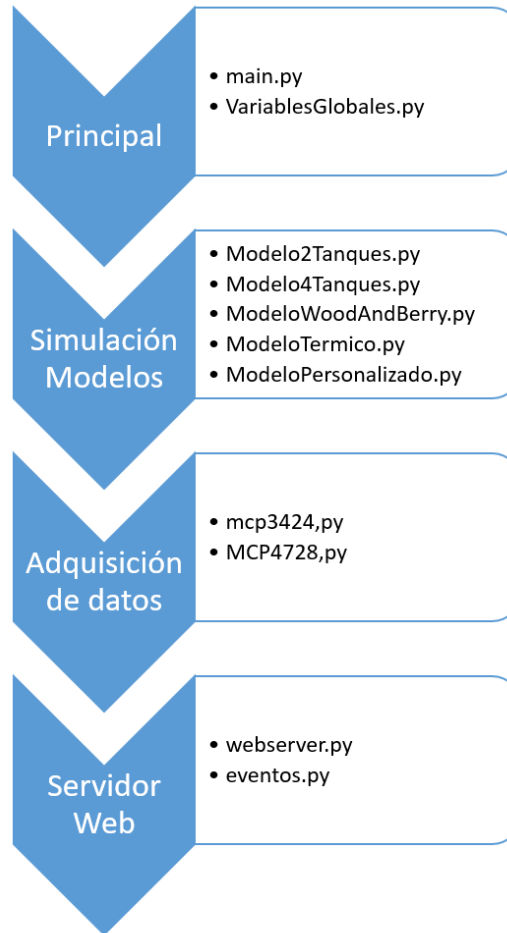


Figura 21. Esquema de ficheros Python

Dentro de los ficheros el cerebro de toda la operación se denomina `main.py` (ver Figura 22), el objetivo de este fichero es importar las librerías necesarias para trabajar en el sistema HIL, llamar a las funciones que definen el cálculo de los modelos, realizar el cálculo del retardo debido a la ejecución de los procesos y ejecutar el servidor web por el puerto 8000.

Las variables globales son definidas en un fichero diferente (`VariablesGlobales.py`) con la finalidad que puedan ser accedidas no solo desde el `main.py` sino también desde cada uno de los modelos definidos para la simulación.

Por otro lado, para la simulación se han definido cinco ficheros diferentes, en los cuales toda la información con respecto a los procesos y su modelo matemático se ha colocado en cada uno de ellos, de esta manera siempre existirá una función que al ser ejecutada devolverá los valores hacia las variables globales y de esta manera se podrá acceder a ellas para observar su resultado.

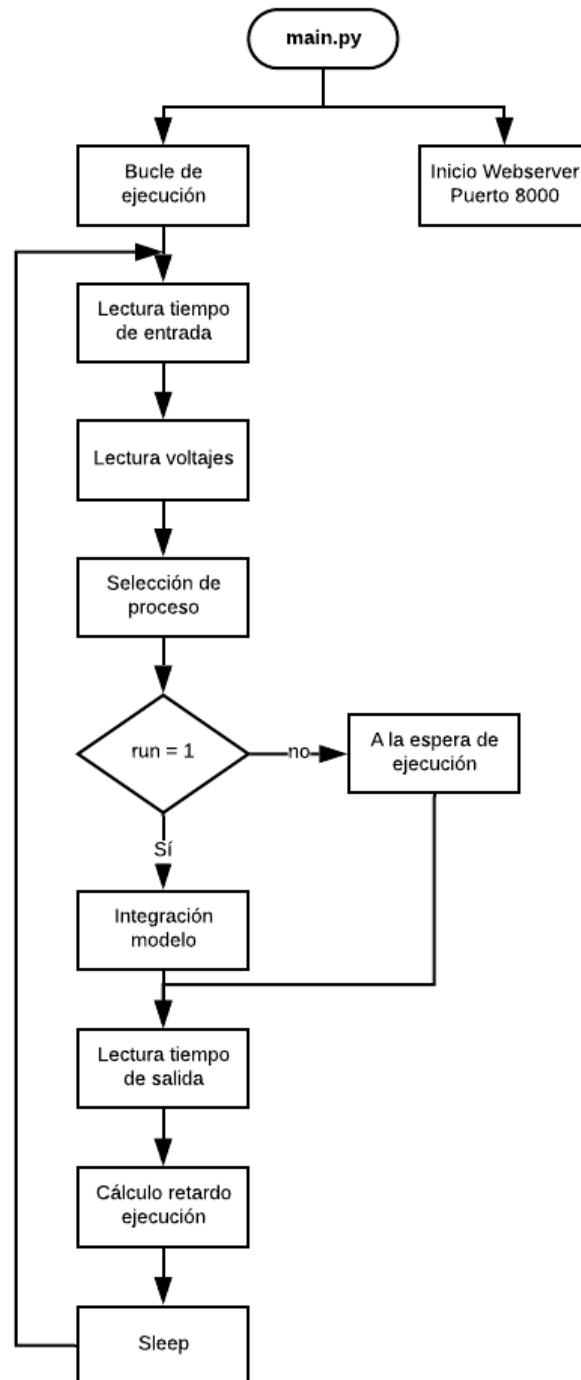


Figura 22. Diagrama de flujo main.py

Las funciones para la adquisición y envío de datos de las entradas y salidas analógicas se encuentra definido en los ficheros mcp2434.py y MCP4728.py, estos ficheros cuentan con toda la información para lograr comunicar los circuitos integrados a través del protocolo I2C.

Para el servidor web se han definido dos ficheros que son los encargados de comunicar el frontend con el backend, en este caso el fichero eventos.py cuenta con las funciones de GET y POST para el envío y recepción de información, además de la definición de las variables globales que se pretenden recibir o enviar.

3.3.2. Plataforma Web

Para la interfaz del usuario se ha definido una estructura que será capaz de mostrar toda la información relevante del proceso a simular (ver Figura 23), normalizar esta estructura para todos los procesos trae ventajas significativas sobre todo al momento de la experiencia del usuario ya que es de fácil navegación y equilibrada.

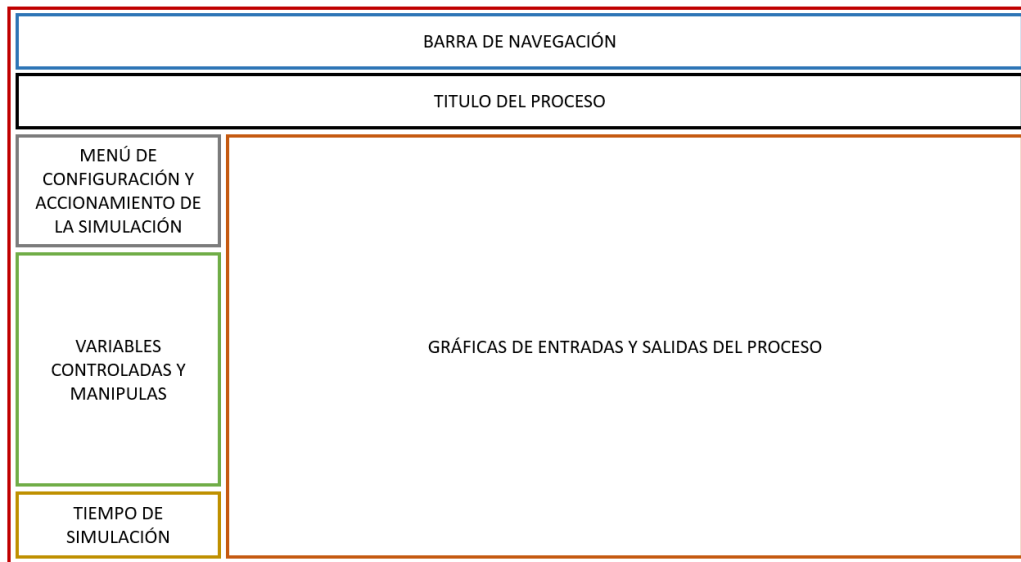


Figura 23. Diseño de la distribución de pantalla.

A continuación, se describe el detalle de cada una de las secciones que contiene la plataforma web:

- Barra de navegación: Menú en el cual se podrá navegar a través de todas las interfaces de los procesos a simular.

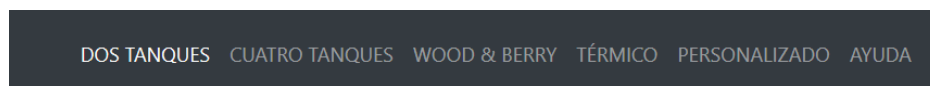


Figura 24. Barra de navegación.

- Título del proceso: Nombre del proceso a simular.
- Menú configuración: Cuenta con un menú desplegable en el cual se tiene por objetivo configurar los distintos parámetros de los procesos a simular, además que se puede observar el esquema del proceso para tener una mejor percepción en cuanto a sus entradas y salidas.

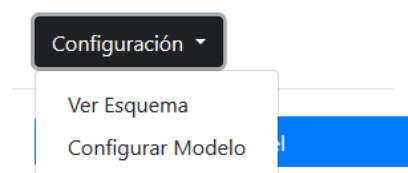


Figura 25. Menú configuración proceso.

- Accionamiento de la simulación: Botones capaces de ejecutar, parar o reiniciar las variables del proceso a simular.



Figura 26. Menú accionamiento de proceso

- Variables: En esta sección se encuentra la información de las variables controladas y manipuladas.

| Variables Controladas | |
|-----------------------|-------|
| Altura H1 | 5.000 |
| Altura H2 | 5.000 |

| Variables Manipuladas | |
|-----------------------|-------|
| Voltaje | 2.118 |
| % Valvula | 0.197 |

Figura 27. Información de las variables del proceso.

- Tiempo simulación: Sección en la que se muestra el tiempo desde el momento de ejecución de la simulación, se encuentra en segundos.
- Gráficas: Para logra una correcta percepción del control del proceso es necesario contar con gráficas que muestren en tiempo real el comportamiento tanto de las entradas como salidas del sistema dinámico.

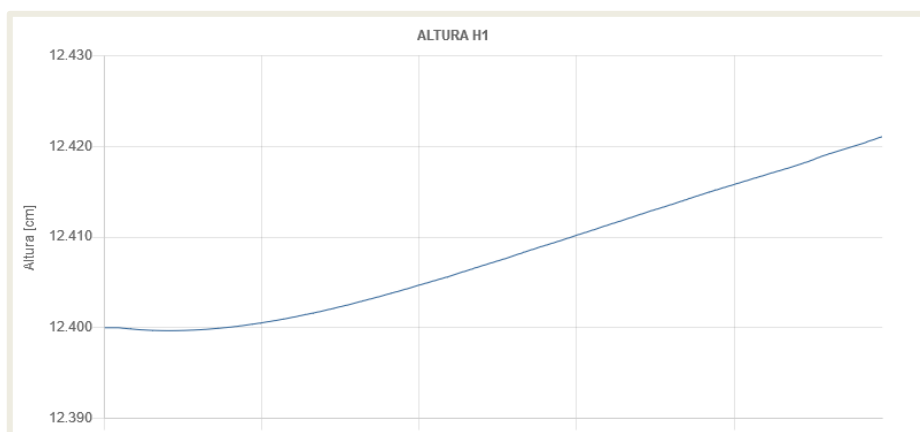


Figura 28. Sección de gráficas del proceso.

- Personalizado: Solo para el modelo personalizado difiera en cierto detalle en cuanto a la estructura, para este tipo de proceso se ha desarrollado una interfaz en la cual el usuario puede ingresar los coeficientes de las funciones de transferencia que rigen su comportamiento ubicándolas de mayor a menor exponente en modo de vector.

Por ejemplo, en el modelo que se detalla a continuación de dos entradas y dos salidas que ha sido representado por sus funciones de transferencia, la forma correcta de ingresarlo en la plataforma se detalla en la Figura 29.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{s+2} & \frac{2}{s^2+2s+4} \\ -0.4 & \frac{s-2}{s^2+0.5s+0.6} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

| | |
|------------------------------------|--|
| G11 | G12 |
| <input type="text" value="1"/> | <input type="text" value="2"/> |
| <input type="text" value="1 2"/> | <input type="text" value="1 2 4"/> |
| G21 | G22 |
| <input type="text" value="-0.4"/> | <input type="text" value="1 -2"/> |
| <input type="text" value="3 2 0"/> | <input type="text" value="1 0.5 0.6"/> |

Figura 29. Configuración modelo personalizado ejemplo 2x2

A continuación, se detalla las capturas de pantalla de cada una de las interfaces de la plataforma web:



Figura 30. Menú principal interfaz web

Desarrollo de un simulador Hardware-In-The-Loop de procesos dinámicos multivariables basado en Raspberry Pi

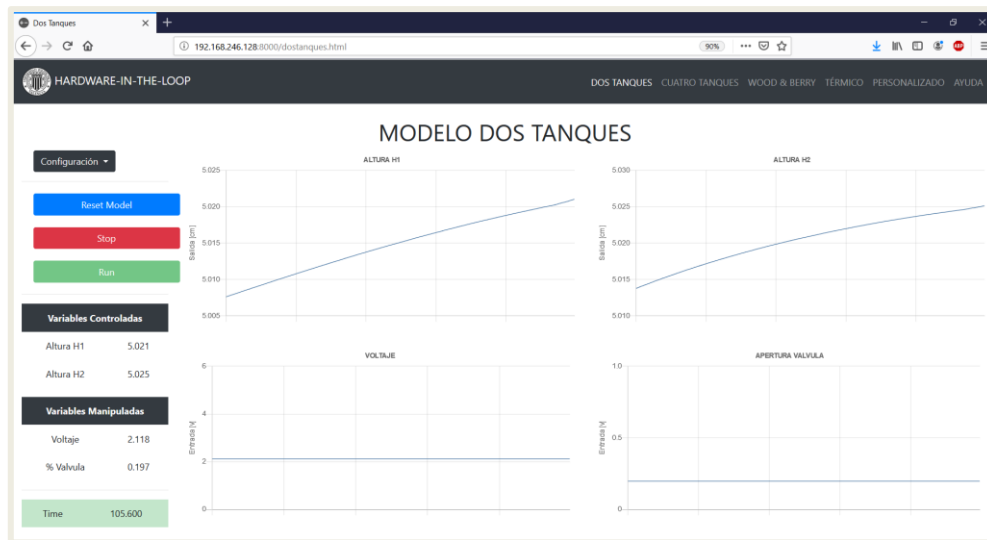


Figura 31. Interfaz del modelo de Dos Tanques Acoplados



Figura 32. Interfaz del modelo de Cuatro Tanques Acoplados.

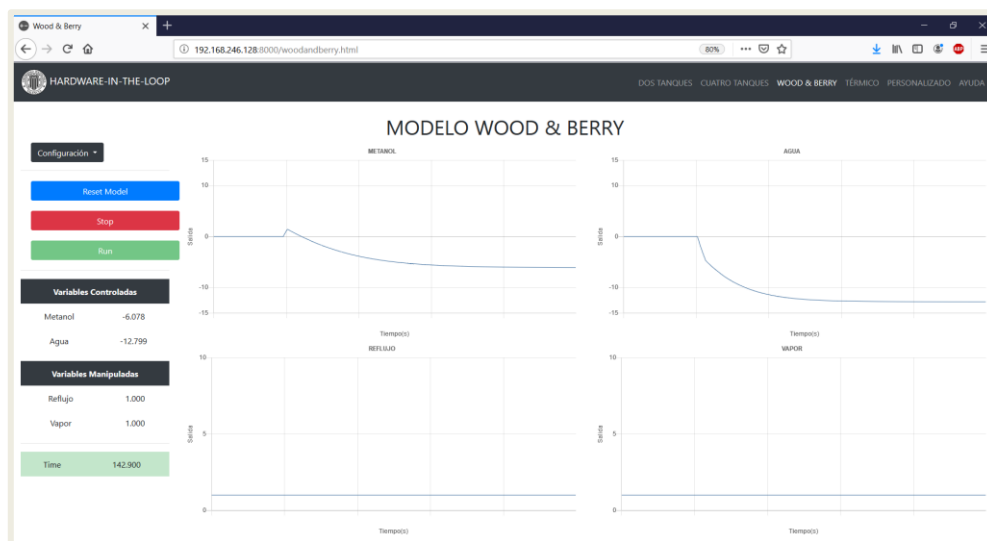


Figura 33. Interfaz del modelo Wood & Berry.

Desarrollo de un simulador Hardware-In-The-Loop de procesos dinámicos multivariables basado en Raspberry Pi

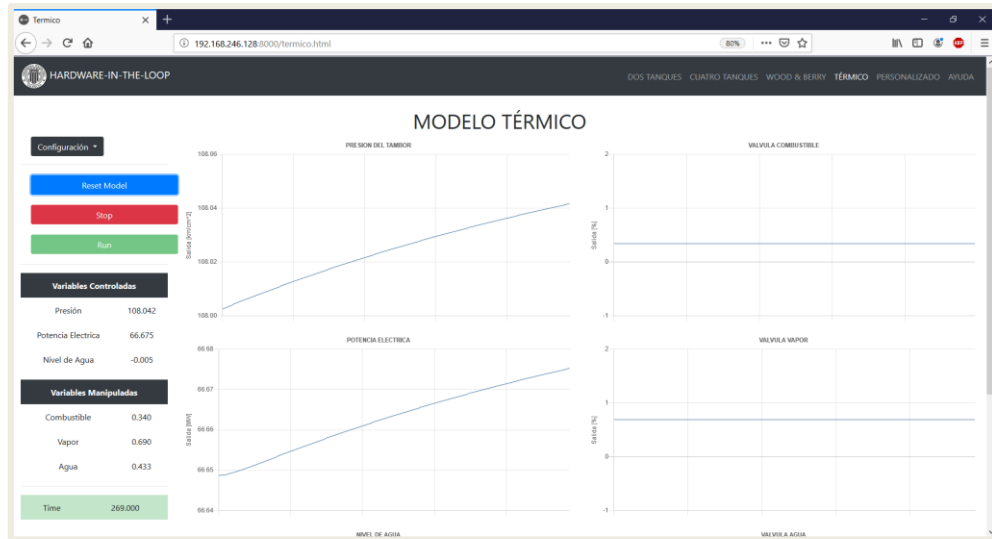


Figura 34. Interfaz del modelo Térmico

Configurar Modelo

Número de entradas:

Número de salidas:

Periodo de muestreo:

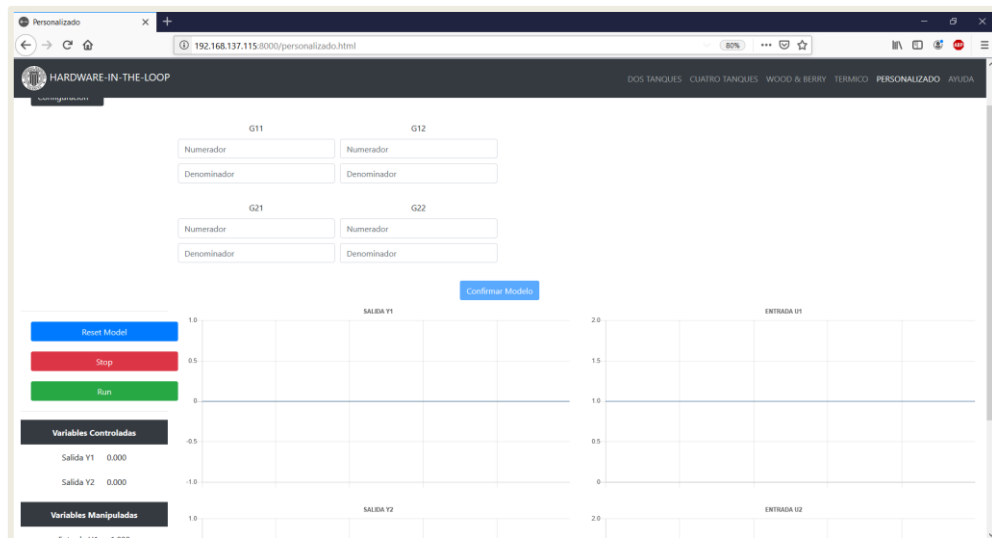


Figura 35. Interfaz del modelo Personalizado.

4. PRUEBAS DE CONCEPTO DEL SISTEMA DESARROLLADO

Para las pruebas de concepto se ha hecho uso del software LabVIEW para generar el controlador y cerrar el lazo del sistema HIL, además junto al software se ha utilizado la tarjeta de adquisición de datos NI 6001.

LabVIEW es el acrónimo de Laboratory Virtual Instrument Engineering Workbench. Es un lenguaje y, a la vez, una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico pensado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido. (UPV/EHU, s.f.)



Figura 36. Esquema de implementación de HIL.

En la Figura 36 se puede observar cómo se cierra el lazo de control teniendo de por medio LabVIEW como herramienta de control y la Raspberry Pi para simular los procesos dinámicos multivariables.

Se debe tener en cuenta que muchas veces por factores externos a la electrónica propia de los dispositivos como la puesta a tierra o la estática en el medio que se está desarrollando la simulación se suele tener pequeñas variaciones en los voltajes de entradas y salidas a pesar de compartir la misma masa entre las tarjetas, se deberá realizar una corrección de voltajes tanto en la Raspberry Pi como en la tarjeta de National Instruments mediante una ecuación que corrija este problema, regresión lineal esto se puede lograr una ecuación que corrija este problema.

En la Tabla 14 se muestran las medidas que se han generado con la tarjeta NI 6001 y los valores que se han obtenido con la Raspberry Pi, con el propósito de conseguir una ecuación que lo corrija, esto se lo realiza a través de software. Siendo $m = 1.007298$ y $b = 0.08589$ lo valores obtenidos.

$$v_{\text{corregidos}} = 1.007298 * v_{in} + 0.08589$$

Mediante la ecuación antes mostrada se puede asegurar una mejor respuesta de la simulación.

Tabla 14. Datos para corregir voltajes recibidos en la TAD

| y | x | Corrección Voltajes |
|-----------|-------------|---------------------|
| Salida NI | Lectura RPI | $y_{corr} = mx + b$ |
| -9.00 | -9.02 | -9.00 |
| -8.00 | -8.03 | -8.00 |
| -7.00 | -7.03 | -7.00 |
| -6.00 | -6.04 | -6.00 |
| -5.00 | -5.05 | -5.00 |
| -4.00 | -4.06 | -4.00 |
| -3.00 | -3.06 | -3.00 |
| -2.00 | -2.07 | -2.00 |
| -1.00 | -1.08 | -1.00 |
| 0.00 | -0.09 | 0.00 |
| 1.00 | 0.91 | 1.00 |
| 2.00 | 1.90 | 2.00 |
| 3.00 | 2.89 | 3.00 |
| 4.00 | 3.89 | 4.00 |
| 5.00 | 4.88 | 5.00 |
| 6.00 | 5.87 | 6.00 |
| 7.00 | 6.86 | 7.00 |
| 8.00 | 7.86 | 8.00 |
| 9.00 | 8.85 | 9.00 |

4.1. Pruebas Dos Tanques

El modelo con el que se ha decidido realizar las pruebas de concepto es el proceso de dos tanques acoplados, ya que la tarjeta NI 6001 tiene como característica cuatro entradas y dos salidas analógicas respectivamente por lo que para este caso en particular es adecuado para poner en marcha el simulador.

Se ha usado el toolbox de “Control & Simulation” que presenta LabVIEW para generar los controladores PID de manera que el mismo software se encarga de verificar el error y calcular la acción de control, es necesario ubicar correctamente las señales de entradas, salidas y setpoint (ver Figura 37).

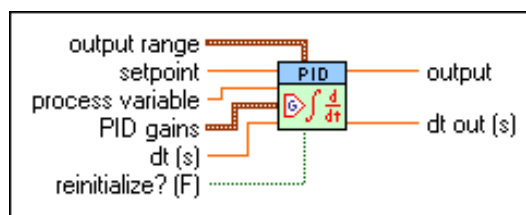


Figura 37. PID VI (National Instruments, s.f.)

Se debe tener en cuenta que los parámetros del tiempo integral (T_i) y el tiempo derivativo (T_d) se encuentran en minutos (ver Figura 38), por lo que si en un principio se está trabajando con Simulink para generar los parámetros del controlador se debe aplicar esta transformación una vez validado los resultados.

Particularmente en este caso se han configurado dos controladores PI por lo que T_d quedará en cero, de igual manera y por seguridad el cuadro de configuración permite delimitar los rangos de salida.

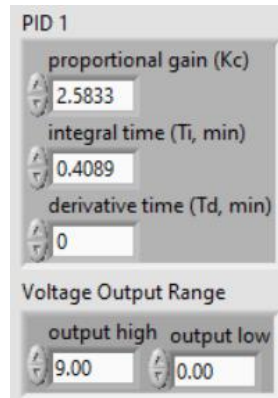


Figura 38. Cuadro de control de parámetros PID

En la Tabla 15 se puede observar los parámetros con los que se ha trabajado en Simulink donde las constantes de tiempo se encuentran en segundos y los parámetros que recoge LabVIEW ya transformados en minutos.

Tabla 15. Parámetros de los controladores PID

| PID 1 | | | | PID 2 | | | |
|----------|---------|-----------|--------|----------|---------|-----------|--------|
| SIMULINK | | LABVIEW | | SIMULINK | | LABVIEW | |
| P | 2.58331 | Kc | 2.5833 | P | 0.13129 | Kc | 0.1313 |
| I | 0.04076 | Ti | 0.4089 | I | 0.03697 | Ti | 0.4508 |
| D | 0.0000 | Td | 0.0000 | D | 0.00000 | Td | 0.0000 |

Para observar el comportamiento de la simulación se ha realizado un HMI que muestra en cada instante cual es la evolución tanto de las variables manipuladas como las controladas y verificar si el proceso llega al setpoint previsto (ver Figura 39).

Desarrollo de un simulador Hardware-In-The-Loop de procesos dinámicos multivariables basado en Raspberry Pi



Figura 39. HMI Proceso Dos Tanques LabVIEW

Con el objetivo de guardar los datos para graficarlos posteriormente en Matlab se ha hecho uso del toolbox File I/O que ha ayudado a generar un documento de texto plano con todas las variables controladas y manipuladas generadas durante el proceso de simulación, no hay que olvidar que el tiempo de muestreo debe ser similar al de la simulación HIL para no perder información.

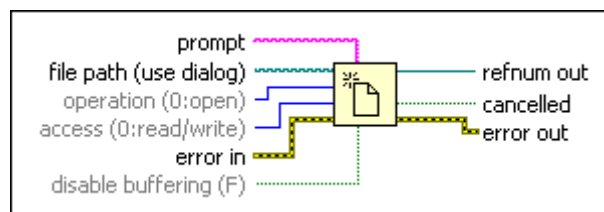


Figura 40. Toolbox File I/O de LabVIEW (National Instruments, s.f.)

Una vez configurado los parámetros del controlador se ha procedido a realizar la simulación y se han encontrado datos muy favorables, como se puede observar en la Figura 41 las respuestas obtenidas del simulador son muy parecidas a las obtenidas en Simulink, claramente se puede apreciar un pequeño rizado en la respuesta del simulador debido al ruido de medida que se suele obtener en procesos reales.

Es interesante observar que a pesar que se ha configurado los controladores con los mismos valores, en el caso del simulador HIL se obtiene un pequeño retardo en cuanto a la respuesta, esto se debe a la velocidad de procesamiento del simulador, sincronismo y la diferencia que existe entre un proceso completamente simulado y uno aproximado a la realidad.

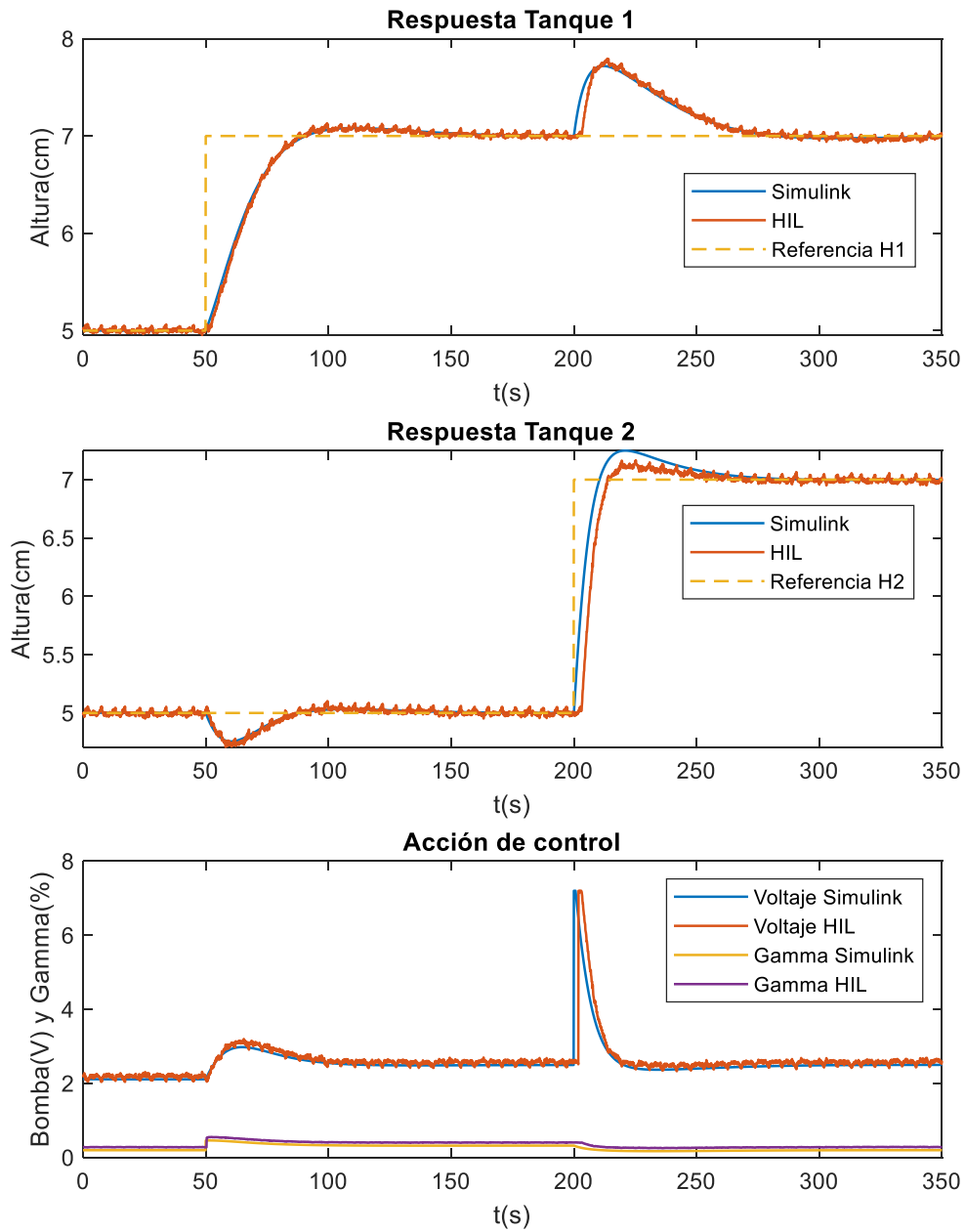


Figura 41. Validación de resultados del Simulador vs. Simulink

5. CONCLUSIONES Y TRABAJOS FUTUROS

El presente trabajo de fin de máster ha logrado su objetivo principal, demostrando que el simulador Hardware-In-The-Loop desarrollado cumple con todas las características de diseño propuestas en un principio, y a pesar de utilizar un computador con un procesamiento limitado como es la Raspberry Pi las respuestas obtenidas llegan a ser muy prometedoras.

Implementar un simulador HIL avanzado de procesos multivariables y de bajo coste es sin lugar a duda un proyecto que tiene un impacto sumamente importante en el campo académico, ya que los estudiantes podrán experimentar de una manera distinta la realidad del control multivariable.

El uso del lenguaje científico Python como núcleo de desarrollo del presente trabajo ha tenido ventajas muy significativas, entre ellas, las poderosas librerías con las que cuentan sus repositorios y la facilidad con la que se las puede implementar en cualquier tipo de aplicación. Como se ha podido observar durante el desarrollo de esta memoria, las librerías Numpy y Scipy han sido claves para lograr el objetivo y el tratamiento de los modelos matemáticos, pero hay que mencionar que la librería Python Control Systems aplicada durante el desarrollo del modelo personalizado le ha dado una versatilidad importante al trabajo logrando realizar modelos con hasta cuatro entradas y cuatro salidas.

Sería interesante que para futuros proyectos del simulador se puedan migrar las tarjetas Raspberry Pi 3B+ a su nueva versión, ya que cuenta hasta con cuatro gigabytes de memoria RAM lo cual puede ser un empuje importante para el simulador HIL, sobre todo para su plataforma.

Adicionalmente los controladores que se han estudiado en las pruebas de concepto han sido PID de la teoría de control clásica, sin embargo, nada impide que se implementen estrategias de control más avanzadas y se estudien sus bondades en el control de procesos multivariables simulados.

No cabe duda que el enfrentar este tipo de proyectos que tienen tanto parte electrónica, informática y de automática lo han hecho un trabajo muy completo en cuanto a generar conocimiento, ya que al afectar directamente estas tres áreas el estudiante se ve obligado a lograr soluciones no convencionales.

6. BIBLIOGRAFÍA

- Chi, J. (s.f.). What is Hardware in the loop (HIL) simulation? Recuperado el 10 de 09 de 2019, de <https://www.youtube.com/watch?v=BrmqyVuyegc>
- Halfacree, G. (2018). *THE OFFICIAL Raspberry Pi Beginner's Guide* (Primera ed.). Cambridge, England: Raspberry Pi PRESS.
- Lussón, A., Bausa, I., Barcaz, Beltrán, R., & Hernández, Y. (2015). "Alternativas para el Control Multivariable". *Inge@UAN*, 5(10), 21-27.
- Montoya, A. (2012). "DISEÑO DE UNA COLUMNA DE DESTILACIÓN PARA RECUPERACIÓN DE UNA SUSTANCIA TERMOSENSIBLE". Proyecto de fin de carrera, Chile: Pontificia Universidad Católica de Valparaíso.
- Mulka, V. (2016). "HARDWARE IN THE LOOP SIMULATION FOR DC MOTOR". Thesis, West Yorkshire, England: University of Huddersfield.
- National Instruments. (s.f.). *ni.com*. Recuperado el 16 de 09 de 2019, de http://zone.ni.com/reference/en-XX/help/371361R-01/lvpid/pid_vi/
- National Instruments. (s.f.). *ni.com*. Recuperado el 16 de 09 de 2019, de https://zone.ni.com/reference/en-XX/help/371361R-01/glang/open_create_replace_file/
- Tlachi, E., Muñoz, G., & Maya, L. (2017). Control de nivel de tanques acoplados mediante GPC y control PI descentralizado. *Congreso Nacional de Control Automático*. Monterrey, Nuevo León, México.
- Tollervey, N. H. (2015). *Python in Education* (First ed.). Sebastopol, CA, USA: O'Reilly Media, Inc.
- UPV/EHU. (s.f.). Recuperado el 16 de 09 de 2019, de Tutorial MyRio: <https://www.ehu.eus/es/web/tutorial-myrio/sarrera>

DESARROLLO DE UN SIMULADOR
HARDWARE-IN-THE-LOOP DE PROCESOS
DINÁMICOS MULTIVARIABLES BASADO EN
RASPBERRY PI

MANUAL DE USUARIO

ÍNDICE CONTENIDO

| | |
|--|---|
| 1. INTRODUCCIÓN..... | 1 |
| 2. PUESTA EN MARCHA DEL SIMULADOR..... | 1 |
| 3. ACCESO A LA INTERFAZ..... | 3 |

ÍNDICE FIGURAS

| | |
|--|---|
| Figura 1. Autenticación VNC..... | 1 |
| Figura 2. Ficheros HIL..... | 2 |
| Figura 3. Acceso a ruta de ficheros..... | 2 |
| Figura 4. Ejecución de simulación y espera de proceso a seleccionar..... | 2 |
| Figura 5. Acceso a interfaz mediante IP y puerto definido..... | 3 |
| Figura 6. Menú principal interfaz..... | 3 |

1. INTRODUCCIÓN

En el presente documento se define el manual de usuario con el cuál trabajará el simulador Hardware-In-The-Loop, es importante seguir las recomendaciones que se exponen en la presente guía con el fin de lograr los siguientes objetivos:

- Acceder al escritorio remoto de Raspbian
- Lanzar el simulador desde la Raspberry Pi
- Acceder correctamente hacia la interfaz.
- Interpretar los datos obtenidos de la simulación.

2. PUESTA EN MARCHA DEL SIMULADOR

Para iniciar el simulador se debe acceder a la Raspberry Pi, sea mediante la forma más habitual conectando los periféricos para utilizarla como un computador normal o a través del servidor VNC (Virtual Network Computing) que ya ha sido habilitado previamente.

Si la segunda opción es la elegida se debe tener en cuenta que se ha dejado estática la IP siendo esta la 192.168.137.115, de esta manera se logrará acceder a través de un escritorio remoto.

Es importante tener en cuenta que el computador anfitrión que desee acceder a la Raspberry Pi debe tener instalado el VNC Viewer.

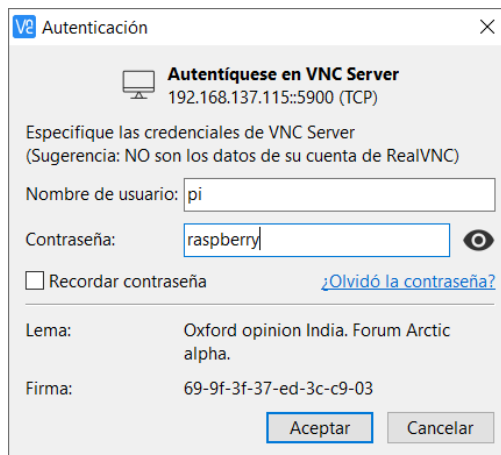


Figura 1. Autenticación VNC

Una vez establecida la conexión se deberá utilizar el usuario y contraseña que por defecto viene en el sistema operativo Raspbian, usuario: pi, contraseña: raspberrypi.

Al acceder al escritorio de la Raspberry se puede observar que se encuentra la carpeta TFM_Marcelo en donde se encuentra todo el núcleo programático del simulador HIL (ver Figura 2).

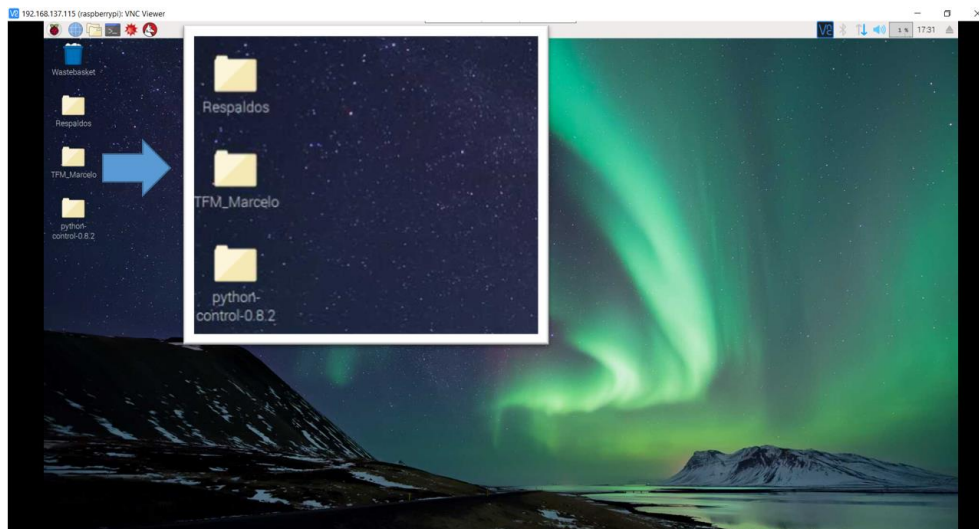


Figura 2. Ficheros HIL

A continuación, se debe abrir una terminal y acceder a la ruta donde se encuentran los ficheros (ver Figura 3) para posteriormente con el comando “*python main.py*” empezar a ejecutar el simulador (ver Figura 4).

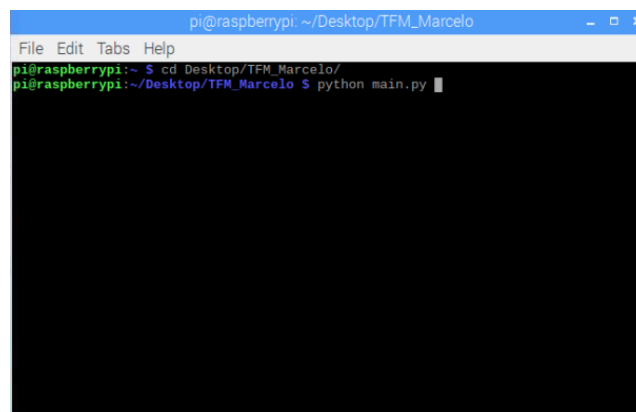


Figura 3. Acceso a ruta de ficheros

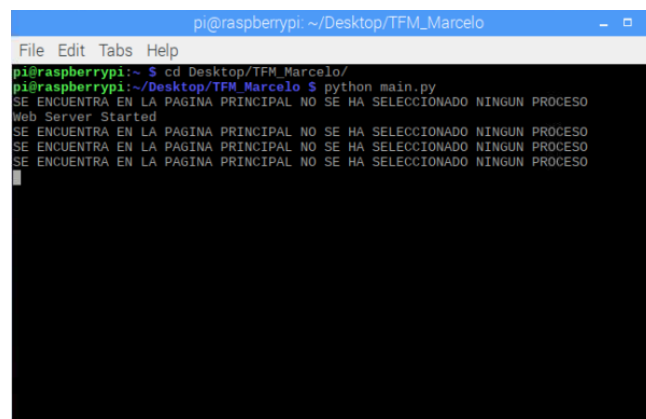


Figura 4. Ejecución de simulación y espera de proceso a seleccionar

3. ACCESO A LA INTERFAZ

Para acceder a la interfaz es necesario usar un explorador web y en la barra de navegación colocar la IP antes mencionada teniendo en cuenta que el puerto a utilizar será el 8000 (ver **Figura 5**).

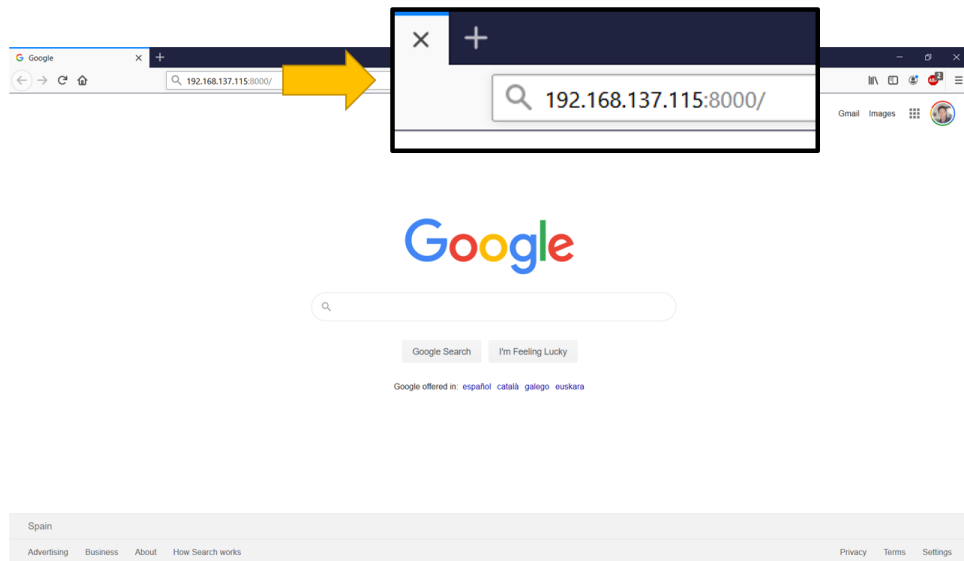


Figura 5. Acceso a interfaz mediante IP y puerto definido

Una vez que se cuente con el acceso se podrá navegar a través de los cinco modelos descritos en la memoria.

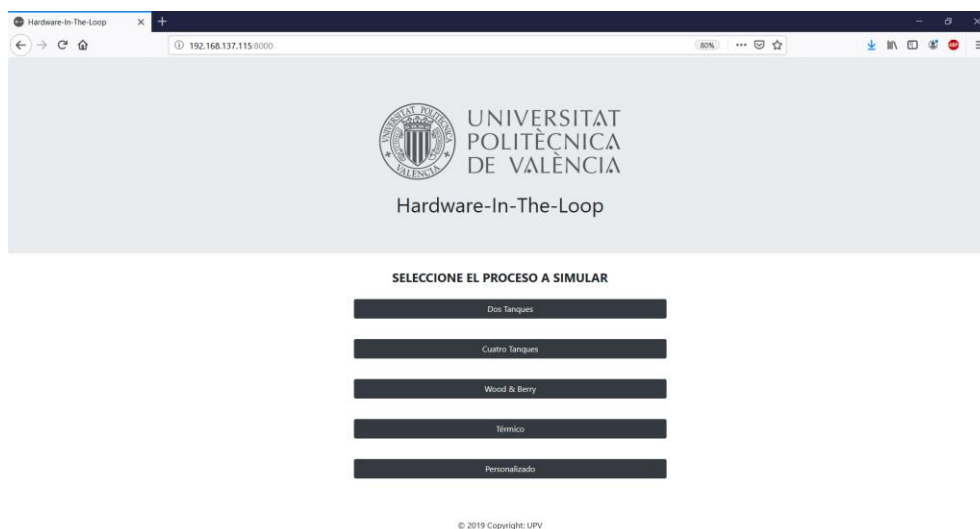


Figura 6. Menú principal interfaz

Nota: Es sumamente importante tener en cuenta los rangos de trabajo definidos, tanto de los modelos matemáticos de los procesos como de la tarjeta de adquisición de datos con el objetivo de lograr una correcta simulación, toda esta información se encuentra descrita en la memoria del proyecto.

DESARROLLO DE UN SIMULADOR
HARDWARE-IN-THE-LOOP DE PROCESOS
DINÁMICOS MULTIVARIABLES BASADO EN
RASPBERRY PI

PRESUPUESTO

ÍNDICE CONTENIDO

| | |
|-------------------------------|---|
| 1. INTRODUCCIÓN..... | 1 |
| 2. PRECIOS UNITARIOS..... | 1 |
| 3. PRECIOS DESCOMPUESTOS..... | 2 |
| 4. PRECIOS FINALES..... | 4 |

ÍNDICE TABLAS

| | |
|---|---|
| Tabla 1. Precios unitarios de mano de obra | 1 |
| Tabla 2. Precios unitarios del software empleado..... | 1 |
| Tabla 3. Precios unitarios del software empleado..... | 1 |
| Tabla 4. Unidad de obra N° 1 | 2 |
| Tabla 5. Unidad de obra N° 2 | 2 |
| Tabla 6. Unidad de obra N° 3 | 2 |
| Tabla 7. Unidad de obra N° 4 | 3 |
| Tabla 8. Unidad de obra N° 5 | 3 |
| Tabla 9. Unidad de obra N° 6 | 3 |

1. INTRODUCCIÓN

El siguiente documento tiene la finalidad de analizar el presupuesto para el desarrollo del proyecto. Se ha considerado el estudio de cada uno de los precios unitarios dentro de cada actividad para lograr un ajuste que tenga concordancia con la realidad del mercado.

Para el cálculo del coste por hora se tiene en consideración que el año tiene 220 días laborables y cada día una jornada laborable de 8 horas diarias.

2. PRECIOS UNITARIOS

Tabla 1. Precios unitarios de mano de obra

| Código | Empleado | Salario (€/mes) | Salario (€/hora) |
|----------|--|-----------------|------------------|
| MO.GMAII | Graduado del Máster en Automática e Informática Industrial | 4,000.00 € | 25.00 € |

Tabla 2. Precios unitarios del software empleado

| Código | Detalle | Unidades | Precio (€) | Horas de funcionamiento | Coste (€/hora) |
|--------|-----------------------|----------|------------|-------------------------|----------------|
| SF.MA | Matlab 2019 | 1 | 800.00 € | 1760 | 0.45 € |
| SF.LV | NI LabVIEW 2018 | 1 | 1,080.00 € | 1760 | 0.61 € |
| SF.WD | Windows 10 | 1 | 30.00 € | 1760 | 0.02 € |
| SF.MO | Microsoft Office 2016 | 1 | 69.00 € | 1760 | 0.04 € |
| SF.PY | Python 2.7 | 1 | 0.00 € | 1760 | 0.00 € |
| SF.RB | Raspbian | 1 | 0.00 € | 1760 | 0.00 € |
| SF.VS | Visual Studio Code | 1 | 0.00 € | 1760 | 0.00 € |

Tabla 3. Precios unitarios del hardware empleado

| Código | Detalle | Unidades | Precio(€) | Horas de funcionamiento | Coste (€/hora) |
|--------|------------------------|----------|-----------|-------------------------|----------------|
| HW.OP | Ordenador portátil | 1 | 850.00 € | 1760 | 0.48 € |
| HW.MP | Monitor y periféricos | 1 | 500.00 € | 1760 | 0.28 € |
| HW.TD | TAD | 1 | 43.00 € | 1760 | 0.02 € |
| HW.RP | Raspberry Pi 3 Model B | 1 | 36.00 € | 1760 | 0.02 € |
| HW.NI | TAD NI DAQ 6001 | 1 | 432.00 € | 1760 | 0.25 € |

3. PRECIOS DESCOMPUESTOS

Tabla 4. Unidad de obra N° 1

| Investigación preliminar | | | | | |
|--------------------------|----------|--|-------------|--------------|-------------------|
| Código | Unidades | Descripción | Rendimiento | Precio | Importe |
| MO.GMAIL | h | Graduado del Máster en Automática e Informática Industrial | 40.00 | 25.00 € | 1,000.00 € |
| HW.OP | h | Ordenador portatil | 40.00 | 0.48 € | 19.32 € |
| SF.WD | h | Windows 10 | 40.00 | 0.02 € | 0.68 € |
| | | | | TOTAL | 1,020.00 € |

Tabla 5. Unidad de obra N° 2

| Simulación de modelos matemáticos Matlab/Simulink | | | | | |
|---|----------|--|-------------|--------------|-------------------|
| Código | Unidades | Descripción | Rendimiento | Precio | Importe |
| MO.GMAIL | h | Graduado del Máster en Automática e Informática Industrial | 40.00 | 25.00 € | 1,000.00 € |
| HW.OP | h | Ordenador portatil | 40.00 | 0.48 € | 19.32 € |
| SF.WD | h | Windows 10 | 40.00 | 0.02 € | 0.68 € |
| SF.MA | h | Matlab 2019 | 40.00 | 0.45 € | 18.18 € |
| | | | | TOTAL | 1,038.18 € |

Tabla 6. Unidad de obra N° 3

| Desarrollo Front End | | | | | |
|----------------------|----------|--|-------------|--------------|-------------------|
| Código | Unidades | Descripción | Rendimiento | Precio | Importe |
| MO.GMAIL | h | Graduado del Máster en Automática e Informática Industrial | 80.00 | 25.00 € | 2,000.00 € |
| HW.OP | h | Ordenador portatil | 80.00 | 0.48 € | 38.64 € |
| SF.WD | h | Windows 10 | 80.00 | 0.02 € | 1.36 € |
| SF.VS | h | Visual Studio Code | 80.00 | 0.00 € | 0.00 € |
| | | | | TOTAL | 2,040.00 € |

Tabla 7. Unidad de obra N° 4

| Desarrollo Back End | | | | | |
|---------------------|----------|--|-------------|--------------|-------------------|
| Código | Unidades | Descripción | Rendimiento | Precio | Importe |
| MO.GMAIL | h | Graduado del Máster en Automática e Informática Industrial | 80.00 | 25.00 € | 2,000.00 € |
| HW.OP | h | Ordenador portatil | 80.00 | 0.48 € | 38.64 € |
| SF.WD | h | Windows 10 | 80.00 | 0.02 € | 1.36 € |
| SF.VS | h | Visual Studio Code | 80.00 | 0.00 € | 0.00 € |
| SF.PY | h | Python 2.7 | 80.00 | 0.00 € | 0.00 € |
| SF.RB | h | Raspbian | 80.00 | 0.00 € | 0.00 € |
| | | | | TOTAL | 2,040.00 € |

Tabla 8. Unidad de obra N° 5

| Montaje del Simulador Hardware-In-The-Loop y prueba de concepto | | | | | |
|---|----------|--|-------------|--------------|-------------------|
| Código | Unidades | Descripción | Rendimiento | Precio | Importe |
| MO.GMAIL | h | Graduado del Máster en Automática e Informática Industrial | 40.00 | 25.00 € | 1,000.00 € |
| HW.OP | h | Ordenador portatil | 40.00 | 0.48 € | 19.32 € |
| HW.MP | h | Monitor y periféricos | 40.00 | 0.28 € | 11.36 € |
| HW.RP | h | Raspberry Pi 3 Model B | 40.00 | 0.02 € | 0.82 € |
| HW.TD | h | TAD | 40.00 | 0.02 € | 0.98 € |
| HW.NI | h | TAD NI DAQ 6001 | 40.00 | 0.25 € | 9.82 € |
| SF.WD | h | Windows 10 | 40.00 | 0.02 € | 0.68 € |
| SF.PY | h | Python | 40.00 | 0.00 € | 0.00 € |
| SF.RB | h | Raspbian | 40.00 | 0.00 € | 0.00 € |
| SF.LV | h | NI LABView 2018 | 40.00 | 0.61 € | 24.55 € |
| | | | | TOTAL | 1,042.98 € |

Tabla 9. Unidad de obra N° 6

| Documentación proyecto | | | | | |
|------------------------|----------|--|-------------|--------------|-------------------|
| Código | Unidades | Descripción | Rendimiento | Precio | Importe |
| MO.GMAIL | h | Graduado del Máster en Automática e Informática Industrial | 40.00 | 25.00 € | 1,000.00 € |
| HW.OP | h | Ordenador portatil | 40.00 | 0.48 € | 19.32 € |
| SF.WD | h | Windows 10 | 40.00 | 0.02 € | 0.68 € |
| SF.MO | h | Microsoft Office 2016 | 40.00 | 0.04 € | 1.57 € |
| | | | | TOTAL | 1,020.00 € |

4. PRECIOS FINALES

| Código | Descripción | Coste | Importe |
|--------|---|------------|--------------------|
| 1 | Investigación preliminar | 1,020.00 € | |
| 2 | Simulación de modelos matemáticos Matlab/Simulink | 1,038.18 € | |
| 3 | Desarrollo Front End | 2,040.00 € | |
| 4 | Desarrollo Back End | 2,040.00 € | |
| 5 | Montaje del Simulador Hardware-In-The-Loop y prueba de concepto | 1,042.98 € | |
| 6 | Documentación proyecto | 1,020.00 € | |
| | | | |
| | Presupuesto de ejecución del material (PEM) | | 8,201.16 € |
| | Gasto generales 13% (del PEM) | | 1,066.15 € |
| | Beneficio industrial 6% (del PEM) | | 492.07 € |
| | | | |
| | Subtotal | | 9,759.38 € |
| | IVA 21% | | 2,049.47 € |
| | | | |
| | TOTAL PRESUPUESTO | | 11,808.85 € |

El presupuesto total del proyecto asciende a la cantidad en Euros:

ONCE MIL OCHOCIENTOS OCHO CON OCHENTA Y CINCO.