

Memoria Trabajo Practico 2021 / 2022

Trabajo del Máster Universitario en Ingeniería Informática

Victor Fagúndez Poyo

Métodos de Simulación y Modelado

E.T.S de Ingeniería Informática

Curso 2021/2022

Historial de Revisiones

<i>Versión</i>	Fecha	Descripción
<i>0.0.0</i>	19.12.21	Creación del documento
<i>1.0.0</i>	09.01.22	Finalización de la versión inicial del documento

Contenido

1.	Ejercicio 1	1
1.1.	Enunciado[1].....	1
1.2.	Respuestas	2
1.2.1.	¿Qué analogías pueden establecerse entre el modelado basado en diagramas de bloques y el paradigma de la simulación analógica?	2
1.2.2.	¿Qué es el paradigma de modelado físico? ¿Qué tipo de modelos matemáticos se obtienen de aplicar el paradigma del modelado físico?.....	2
1.2.3.	¿Qué diferencias hay entre el paradigma de la simulación analógica y el paradigma del modelado físico?	3
1.2.4.	Observe la Figura 1.1 y comente su contenido basándose en el artículo.	3
2.	Ejercicio 2	4
2.1.	Enunciado	4
2.2.	Ecuaciones del modelo del sistema.....	7
2.3.	Causalidad computacional	10
2.3.1.	Clasificación de las variables	10
2.3.1.1.	Sustituir las derivadas respecto al tiempo por variables mudas	10
2.3.1.2.	Clasificar las variables en conocidas y desconocidas	12
2.3.2.	Matriz de incidencia.....	12
2.3.3.	Singularidad estructural	12
2.3.4.	Asignación de la causalidad computacional	13
2.4.	Diagrama de flujo	18
2.5.	Script en R	20
2.5.1.	Código.....	20
2.5.2.	Representación grafica.....	25
3.	Ejercicio 3	29
3.1.	Enunciado	29
3.2.	Modelo atómico	30
3.3.	Librería	32
3.4.	Representación grafica.....	35
4.	Ejercicio 4	38
4.1.	Enunciado	38
4.2.	Código	40
4.3.	Representación Grafica.....	41
5.	Bibliografía	43

Tabla de Ilustraciones

Ilustración 1. Evolución del modelado y simulación de tiempo continuo.....	1
Ilustración 2. Sistema de depósito cilíndrico a analizar	4
Ilustración 3.Evolución en el tiempo del valor consigna para la altura del liquido	5
Ilustración 4. Diagrama de flujo del sistema de cilindro y válvulas.....	18
Ilustración 5. Estado de la válvula 1	25
Ilustración 6. Estado de la válvula 2.....	25
Ilustración 7. Flujo de salida 1.....	26
Ilustración 8. Flujo de salida 2.....	26
Ilustración 9. Flujo de entrada	27
Ilustración 10. Altura de referencia	27
Ilustración 11. Altura real.....	28
Ilustración 12. Representación de la altura del líquido junto con su valor consigna	35
Ilustración 13. Flujo de Entrada	35
Ilustración 14. Flujos de salida.....	36
Ilustración 15. Variable que describe la fase de la válvula 1	36
Ilustración 16. Variable que describe la fase de la válvula 2	37
Ilustración 17. Sección del bloque de madera (dibujo no a escala).....	38
Ilustración 18. Curvas de Temperatura.....	41
Ilustración 19. Flujo de calor.....	42

Tabla de Tablas

Tabla 1. Magnitudes físicas relevantes del sistema mostrado en la ilustración 2	7
Tabla 2. Clasificación de variables en conocidas y desconocidas.....	12

1. Ejercicio 1

1.1. Enunciado[1]

Lea el artículo citado a continuación, que puede descargar de la página web de la asignatura, y conteste a las preguntas.

Åström, K.J., Elmqvist, H., Mattsson, S.E. Evolution of continuous-time modeling and simulation. The 12th European Simulation Multiconference, ESM'98, June 16–19, 1998, Manchester, UK.

1. ¿Qué analogías pueden establecerse entre el modelado basado en diagramas de bloques y el paradigma de la simulación analógica?
2. ¿Qué es el paradigma de modelado físico? ¿Qué tipo de modelos matemáticos se obtienen de aplicar el paradigma del modelado físico?
3. ¿Qué diferencias hay entre el paradigma de la simulación analógica y el paradigma del modelado físico?
4. Observe la Figura 1.1 y comente su contenido basándose en el artículo.

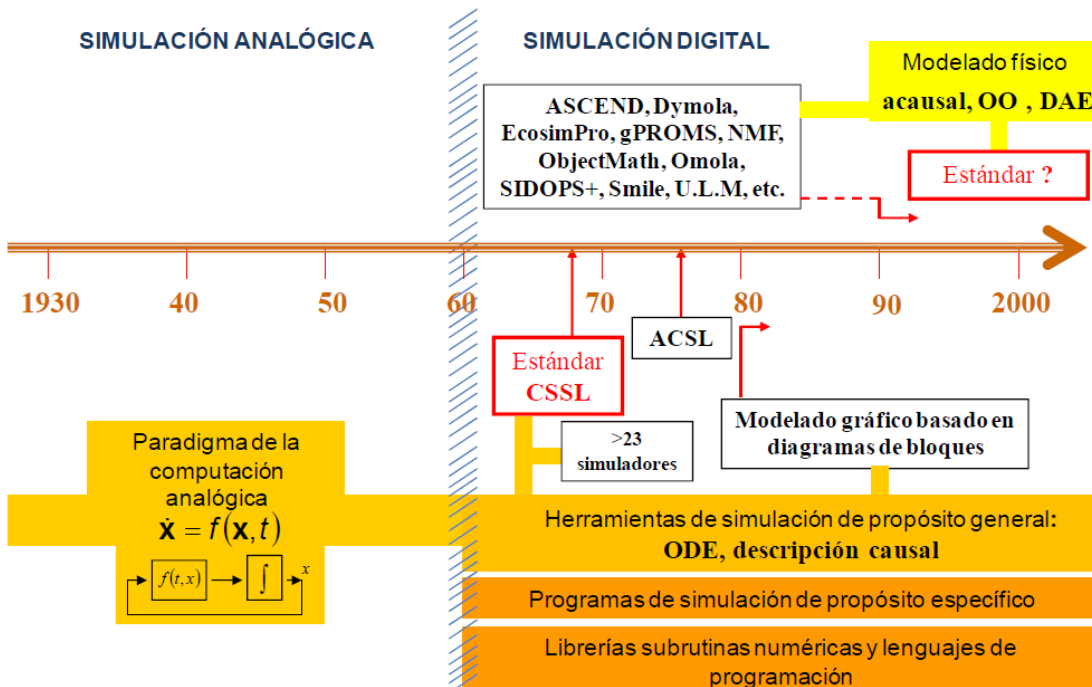


Ilustración 1. Evolución del modelado y simulación de tiempo continuo

1.2. Respuestas

1.2.1. ¿Qué analogías pueden establecerse entre el modelado basado en diagramas de bloques y el paradigma de la simulación analógica?

El paradigma de la simulación analógica se basa en la idea de modelar un sistema en términos de ecuaciones diferenciales ordinarias y después hacer un dispositivo físico que obedezca a las ecuaciones. Mientras tanto el paradigma de modelado basado en diagramas de bloques solía usarse para representar estos modelos físicos en términos de integradores, sumadores y potenciómetros en los primeros días de la simulación analógica.

1.2.2. ¿Qué es el paradigma de modelado físico? ¿Qué tipo de modelos matemáticos se obtienen de aplicar el paradigma del modelado físico?

El paradigma del modelado físico es una metodología para el modelado de los sistemas físicos, que propone plantear el modelo de la forma siguiente:

1. Definir la estructura del sistema y dividir el sistema en partes.
2. Definir la interacción entre las partes.
3. Definir el comportamiento interno de cada parte, independientemente de las demás, mediante ecuaciones: en términos de balances de materia, energía y momento, y de relaciones constitutivas.

El modelo que típicamente se obtiene mediante la aplicación de este paradigma es un modelo de tiempo continuo, descrito mediante ecuaciones algebraicas y diferenciales (modelo DAE).

1.2.3. ¿Qué diferencias hay entre el paradigma de la simulación analógica y el paradigma del modelado físico?

La idea del paradigma de simulación analógica es modelar el sistema en términos de ecuaciones diferenciales ordinarias y luego construir un dispositivo físico que obedezca estas ecuaciones.

1.2.4. Observe la Figura 1.1 y comente su contenido basándose en el artículo.

La Ilustración 1 representa la evolución de los diferentes paradigmas de simulación desde los años 30 del siglo XX hasta comienzos del siglo XXI.

La simulación en la primera mitad del siglo XX se basa en el paradigma de la computación analógica, este consistía en modelar un sistema en términos de ecuaciones diferenciales ordinarias, para posteriormente reproducir estas ecuaciones en un modelo físico o electrónico.

A partir de la llegada de los ordenadores a partir de los años 60 se introduce la simulación digital. Con la llegada del estándar CSSL (1967) se unificaron conceptos y estructuras del lenguaje entre los programas de simulación disponibles.

Más adelante en 1976 se presentó el programa ACSL basado en el estándar CSSL que se convirtió en estándar de facto en la simulación durante un largo periodo.

El modelado basado en diagramas de bloques se comenzó a popularizar a partir de la generalización de los ordenadores personales con rasterización de gráficos, en la década de los 80.

2. Ejercicio 2

2.1. Enunciado

Consideremos el sistema mostrado en la figura, que está compuesto por un depósito cilíndrico, una tubería a través de la cual entra un caudal de líquido F_{in} al depósito, y dos tuberías de desagüe con sendas válvulas, a través de las cuales sale del depósito un caudal de líquido $F_{out,1}$ y $F_{out,2}$ respectivamente. El área de la base del depósito, A , es igual a 2 m^2 .

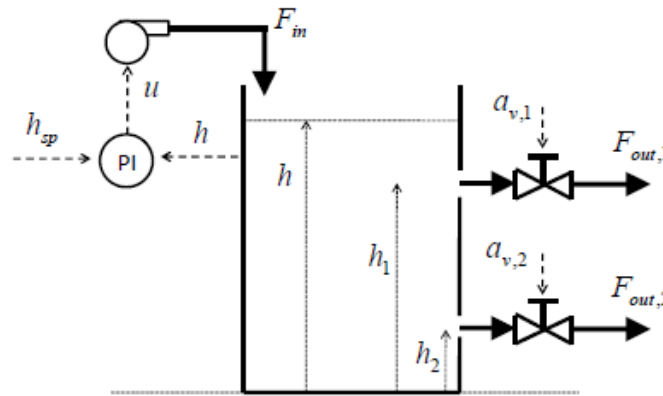


Ilustración 2. Sistema de depósito cilíndrico a analizar

Una válvula puede encontrarse en dos fases: abierta ($a_v = 1$) y cerrada ($a_v = 0$). En el primer caso (abierta) permite el paso de líquido y en el segundo (cerrada) lo impide. La fase de la primera válvula cambia cada $T = 30$ segundos, estando inicialmente cerrada. Mientras la primera válvula está abierta, la segunda válvula está cerrada y viceversa.

La primera tubería está situada a una altura $h_1 = 4 \text{ m}$ y la segunda a una altura $h_2 = 1.5 \text{ m}$. Sólo circula líquido por una tubería si, estando su válvula abierta, la altura del líquido en el depósito (h) es mayor que la altura a la que está situada la tubería. Cuando circula líquido a través de las tuberías, el caudal se calcula de la forma siguiente:

$$F_{out,1} = K * \sqrt{h - h_1}$$

$$F_{out,2} = K * \sqrt{h - h_2}$$

siendo K un parámetro de valor conocido. El valor de K es 0.5 , expresado en unidades del sistema internacional.

El caudal de entrada de líquido, F_{in} , es proporcionado por una bomba que es manipulada por un controlador PI. El caudal de la bomba (F_{in}) es proporcional a la tensión de control de la bomba (u) en el rango de valores entre 0 y 20 voltios. La relación constitutiva de la bomba es la siguiente:

$$F_{in} = B * \min(20, \max(0, u))$$

Donde B es un parámetro cuyo valor es: $B = 0.3 \text{ m}^3 / (\text{s} * \text{V})$

El controlador PI está descrito mediante las ecuaciones siguientes:

$$e = h_{sp} - h$$

$$\frac{dI}{dt} = e$$

$$u = k_p * e + \frac{1}{k_I} * I$$

donde los parámetros del controlador valen: $k_p = 5 \text{ V/m}$, $k_I = 10 \text{ m} \cdot \text{s/V}$. La evolución en el tiempo del valor de consigna para la altura de líquido en el depósito, h_{sp} , es conocida. Se muestra en la figura siguiente h_{sp} frente al tiempo, para el intervalo entre el instante inicial y $t = 240 \text{ s}$.

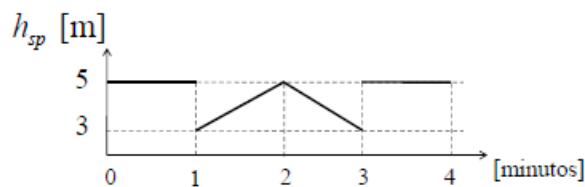


Ilustración 3. Evolución en el tiempo del valor consigna para la altura del líquido

La altura inicial de líquido en el depósito es 0.5 m.

1. Escriba las ecuaciones del modelo del sistema. El modelo debe describir la evolución de la altura de líquido (h) y de su valor de consigna (h_{sp}), del caudal de entrada (F_{in}), de los caudales de salida ($F_{out,1}$, $F_{out,2}$), de las variables del controlador (e , u , I), y de las variables que describen las fases de las válvulas ($av,1$, $av,2$).
2. Asigne la causalidad computacional. Indique cuántos grados de libertad tiene el modelo.

3. Escriba el diagrama de flujo del algoritmo para la simulación de este modelo. Emplee el método de integración de Euler explícito. La condición de finalización de la simulación es que el tiempo alcance el valor 240 s.
4. Programe el algoritmo anterior en lenguaje R y ejecute la simulación. Represente gráficamente frente al tiempo las variables siguientes: la altura de líquido, su valor de consigna, los flujos de entrada y salida, y las variables que describen las fases de las válvulas. Explique qué criterio ha seguido para escoger el tamaño del paso de integración.

2.2. Ecuaciones del modelo del sistema

Las magnitudes físicas que intervienen en el modelo se muestran en la Tabla 1 que se muestra a continuación.

Tabla 1. Magnitudes físicas relevantes del sistema mostrado en la ilustración 2

Símbolo	Significado	Unidades	Valor
F_{in}	Caudal de líquido de entrada	kg/s	
$F_{out,1}, F_{out,2}$	Caudales de líquido de salida	kg/s	
A	Sección de la base del depósito	m^2	2.0
h_1	Altura de la primera tubería	m	4.0
h_2	Altura de la segunda tubería	m	1.5
h	Nivel de líquido en el deposito	m	
K	Parámetro de la tubería		0.5
B	Parámetro de la bomba	$m^3/(s * V)$	0.3
k_p	Parámetro proporcional del controlador	V/m	5.0
K_I	Parámetro integral del controlador	$(m * s)/V$	10.0
e	Error calculado en el controlador	m	
h_{sp}	Valor de consigna para la altura del líquido en el deposito	m	
u	Tensión de control de la bomba	V	
a_{v1}, a_{v2}	Estado de la válvula		
m	Masa del líquido en el deposito	kg	
t	Tiempo	s	

En este modelo consideramos que la sección de la base del depósito (A), las alturas de las tuberías (h_1, h_2), así como los parámetros de proporcionalidad de la bomba y el controlador y la densidad del líquido son parámetros. Las variables del modelo ($m, h, F_{in}, F_{out,1}, F_{out,2}, e, I, u, h_{sp}, a_{v1}, a_{v2}$) son calculadas en las ecuaciones (1) – (11) siguientes.

$$h_{sp} = \begin{cases} 5 \text{ m} & \text{si } t < 60 \text{ s} \\ \frac{t + 30}{30} & \text{si } 60 \text{ s} \leq t < 120 \text{ s} \\ \frac{-t + 30}{30} & \text{si } 120 \text{ s} \leq t < 180 \text{ s} \\ 5 \text{ m} & \text{si } t \geq 180 \text{ s} \end{cases} \quad (1)$$

$$F_{in} = B * \min(20, \max(0, u)) \quad (2)$$

$$F_{out1} = \begin{cases} 0 & \text{si } h \leq h_1 \\ K * \sqrt{h - h_1} & \text{si } h > h_1 \end{cases} \quad (3)$$

$$F_{out2} = \begin{cases} 0 & \text{si } h \leq h_2 \\ K * \sqrt{h - h_2} & \text{si } h > h_2 \end{cases} \quad (4)$$

$$e = h_{sp} - h \quad (5)$$

$$\frac{dI}{dt} = e \quad (6)$$

$$u = k_p * e + \frac{1}{k_I} * I \quad (7)$$

$$a_{v1} = 0.5 * \operatorname{sgn}\left(-\operatorname{sen}\left(\frac{\pi * t}{30}\right)\right) + 0.5 \quad (8)$$

$$a_{v2} = 0.5 * \operatorname{sgn}\left(\operatorname{sen}\left(\frac{\pi * t}{30}\right)\right) + 0.5 \quad (9)$$

$$A * \frac{dh}{dt} = F_{in} - (F_{out1} + F_{out2}) \quad (10)$$

La Ec. (1) expresa la evolución en el tiempo del valor de la consigna para el nivel del líquido. La Ec. (2) es la relación constitutiva de la fuente y las Ecs. (3) – (4) describen la dependencia del flujo masico de salida a través de las 2 válvulas del sistema. Las Ecs. (5) – (7) describen la ley de control del controlador PI. Las Ecs. (8) – (9) describen el proceso de apertura y cierre de las válvulas. Asumiendo que la densidad del líquido es constante, el balance de masa en el depósito se escribe como en la Ec. (10) donde el área A del depósito es un parámetro de valor conocido.

Las condiciones iniciales quedan especificadas indicando el valor de la altura de líquido que hay inicialmente en el depósito e indicando que la integral del error vale inicialmente 0.

2.3. Causalidad computacional

El objetivo de la asignación de la causalidad computacional es decidir de qué ecuación se evalúa cada una de las variables desconocidas y en qué orden debe realizarse dicha evaluación.

2.3.1. Clasificación de las variables

Como paso previo a la asignación de la causalidad computacional, se realizan las siguientes dos acciones:

1. *Sustituir las derivadas respecto al tiempo por variables mudas.* Aplicando el criterio de llamar a la variable muda con el mismo nombre que la variable que aparece derivada, anteponiendo el prefijo “der”.
2. *Clasificar las variables en conocidas y desconocidas.* Las variables desconocidas son aquellas que deben calcularse evaluando las ecuaciones del modelo. Se consideran conocidas:
 - La variable tiempo (t)
 - Las constantes y parámetros del modelo, ya que sus valores no cambian durante el curso de la simulación y son conocidos al comienzo de ésta.

Se consideran desconocidas:

- Las variables mudas, introducidas de la forma descrita anteriormente en sustitución de las derivadas respecto al tiempo.
- Las restantes variables del modelo. Es decir, aquellas variables que, no siendo variables de estado, dependen para su cálculo en el instante de evaluación del valor de otras variables.

2.3.1.1. Sustituir las derivadas respecto al tiempo por variables mudas

Sustituimos las derivadas respecto al tiempo por variables mudas.

$$\frac{dh}{dt} \rightarrow derh \quad (11)$$

$$\frac{dl}{dt} \rightarrow derl \quad (12)$$

Además, en las Ecs. (1), (3) y (4) con dos o más ramas, pueden ser equivalentemente descritas mediante las Ecs. (13), (15) y (16). En la Ec. (13) α , β , γ son variables mudas

de tiempo discreto, donde α vale 1 si $t < 60s$ o $t \geq 180s$ sino 0. β vale 1 si $60s \leq t < 120s$ sino 0. γ vale 1 si $120s \leq t < 180s$ sino 0. Realizando las sustituciones se obtiene el modelo siguiente:

$$h_{sp} = 5 * \alpha + \left(\frac{t + 30}{30}\right) * \beta + \left(\frac{-t + 30}{30}\right) * \gamma \quad (13)$$

$$F_{in} = B * \min(20, \max(0, u)) \quad (14)$$

$$F_{out1} = K * a_{v1} * \sqrt{h - h_1} \quad (15)$$

$$F_{out2} = K * a_{v2} * \sqrt{h - h_2} \quad (16)$$

$$e = h_{sp} - h \quad (17)$$

$$derI = e \quad (18)$$

$$u = k_p * e + \frac{1}{k_I} * I \quad (19)$$

$$a_{v1} = 0.5 * sgn\left(-sen\left(\frac{\pi * t}{30}\right)\right) + 0.5 \quad (20)$$

$$a_{v2} = 0.5 * sgn\left(sen\left(\frac{\pi * t}{30}\right)\right) + 0.5 \quad (21)$$

$$A * derh = F_{in} - (F_{out1} + F_{out2}) \quad (22)$$

2.3.1.2. Clasificar las variables en conocidas y desconocidas

Supongamos que las dos variables que aparecen derivadas son variables de estado. De esta forma, las variables se clasifican de la manera siguiente:

- Parámetros y Constantes: $A, h_1, h_2, K, B, k_p, k_I$
- Variables de estado: h, I
- Variables algebraicas: $F_{in}, F_{out1}, F_{out2}, h_{sp}, e, u, a_{v1}, a_{v2}$

Tabla 2. Clasificación de variables en conocidas y desconocidas

Conocidas	t $A, h_1, h_2, K, B, k_p, k_I$ h, I
Desconocidas	$F_{in}, F_{out1}, F_{out2}, h_{sp}, e, u, a_{v1}, a_{v2}$ $derh, derI$

Dado que las variables de estado son calculadas mediante la integración numérico de sus derivadas, las variables de estado de este sistema son h, I y por tanto el **número de grados de libertad** es 2.

2.3.2. Matriz de incidencia

Se muestra a continuación la matriz de incidencia original del modelo, construida de las Ecs. (13) - (22) y de la clasificación anterior de las variables.

	h_{sp}	F_{in}	F_{out1}	F_{out2}	e	u	$derh$	$derI$	a_{v1}	a_{v2}
$f_1(h_{sp})$	X	0	0	0	0	0	0	0	0	0
$f_2(F_{in}, u)$	0	X	0	0	0	X	0	0	0	0
$f_3(F_{out1}, a_{v1}, h)$	0	0	X	0	0	0	0	0	X	0
$f_4(F_{out2}, a_{v2}, h)$	0	0	0	X	0	0	0	0	0	X
$f_5(e, h, h_{sp})$	X	0	0	0	X	0	0	0	0	0
$f_6(derI, e)$	0	0	0	0	X	0	0	X	0	0
$f_7(u, e, I)$	0	0	0	0	X	X	0	0	0	0
$f_8(a_{v1})$	0	0	0	0	0	0	0	0	X	0
$f_9(a_{v2})$	0	0	0	0	0	0	0	0	0	X
$f_{10}(derh, F_{in}, F_{out1}, F_{out2})$	0	X	X	X	0	0	X	0	0	0

(23)

2.3.3. Singularidad estructural

El siguiente paso consiste en comprobar que el modelo no sea estructuralmente singular. Para ello, se comprueba que:

1. *El número de ecuaciones y de variables desconocidas (incógnitas) es el mismo.*
Este modelo se compone de 10 ecuaciones y tiene 10 variables desconocidas ($h_{sp}, F_{in}, F_{out1}, F_{out2}, e, u, derh, derl, a_{v1}, a_{v2}$). Por lo tanto, se cumple este paso.
2. *Cada variable desconocida puede emparejarse con una ecuación en que aparezca y con la cual no se haya emparejado ya otra variable desconocida.*

Para ello, se buscan permutaciones de las variables (columnas) de la matriz de incidencia de modo que la matriz resultante, con las variables permutadas de orden, tenga todos sus elementos diagonales distintos de cero. Si no se consigue, se dice que es un sistema DAE singular.

Ejecutando estas permutaciones sobre nuestra matriz de incidencia (23) obtenemos la matriz a la que le hemos reordenado las columnas (24) en la que podemos observar que todos sus elementos diagonales son distintos de cero.

Por lo tanto, estamos ante un sistema DAE **no singular**.

	h_{sp}	F_{in}	F_{out1}	F_{out2}	e	$derl$	u	a_{v1}	a_{v2}	$derh$
$f_1(h_{sp})$	X	0	0	0	0	0	0	0	0	0
$f_2(F_{in}, u)$	0	X	0	0	0	0	X	0	0	0
$f_3(F_{out1}, a_{v1}, h)$	0	0	X	0	0	0	0	X	0	0
$f_4(F_{out2}, a_{v2}, h)$	0	0	0	X	0	0	0	0	X	0
$f_5(e, h, h_{sp})$	X	0	0	0	X	0	0	0	0	0
$f_6(derl, e)$	0	0	0	0	X	X	0	0	0	0
$f_7(u, e, l)$	0	0	0	0	X	0	X	0	0	0
$f_8(a_{v1})$	0	0	0	0	0	0	0	X	0	0
$f_9(a_{v2})$	0	0	0	0	0	0	0	0	X	0
$f_{10}(derh, F_{in}, F_{out1}, F_{out2})$	0	X	X	X	0	0	0	0	0	X

(24)

2.3.4. Asignación de la causalidad computacional

Ahora aplicaremos el algoritmo de partición para encontrar las permutaciones de las filas (ecuaciones) y las permutaciones de las columnas (variables) de la matriz de incidencia que hagan que ésta quede convertida en una matriz triangular inferior por bloques BLT (Block Lower Triangular). Esto es, en una matriz con elementos distintos de cero formando bloques cuadrados, del menor tamaño posible, en la diagonal; y con todos los elementos situados por encima de la diagonal y no pertenecientes a los bloques diagonales, iguales a cero.

El procedimiento que seguiremos para realizar la partición manualmente consiste en aplicar repetidamente las dos reglas siguientes:

Regla 1: Si una ecuación contiene una única variable desconocida aun no evaluada, entonces debe emplearse esa ecuación para evaluar dicha variable.

Regla 2: si una variable desconocida que aún no ha sido evaluada aparece en una única ecuación del modelo, entonces la variable debe ser evaluada de esa ecuación.

Ahora aplicaremos las 2 reglas anteriores a nuestro modelo cuya matriz de incidencia es (23).

1. La ecuación f_1 tiene una única variable desconocida: la variable h_{sp} . Por tanto, deberá usarse la ecuación f_1 para evaluar h_{sp} . Como h_{sp} interviene en otras ecuaciones además de en f_1 , es necesario conocer su valor para poder calcular otras variables desconocidas. Por tanto, se pasa la ecuación f_1 a la primera fila de la matriz y la variable h_{sp} a la primera columna.

	h_{sp}	F_{in}	F_{out1}	F_{out2}	e	u	$derh$	$derI$	a_{v1}	a_{v2}
$f_1(h_{sp})$	X	0	0	0	0	0	0	0	0	0
$f_2(F_{in}, u)$	0	X	0	0	0	X	0	0	0	0
$f_3(F_{out1}, a_{v1}, h)$	0	0	X	0	0	0	0	0	X	0
$f_4(F_{out2}, a_{v2}, h)$	0	0	0	X	0	0	0	0	0	X
$f_5(e, h, h_{sp})$	X	0	0	0	X	0	0	0	0	0
$f_6(derI, e)$	0	0	0	0	X	0	0	X	0	0
$f_7(u, e, I)$	0	0	0	0	X	X	0	0	0	0
$f_8(a_{v1})$	0	0	0	0	0	0	0	0	X	0
$f_9(a_{v2})$	0	0	0	0	0	0	0	0	0	X
$f_{10}(derh, F_{in}, F_{out1}, F_{out2})$	0	X	X	X	0	0	X	0	0	0

(25)

2. La ecuación f_8 tiene una única variable desconocida: la variable a_{v1} . Por tanto, deberá usarse la ecuación f_8 para evaluar a_{v1} . Como a_{v2} interviene en otras ecuaciones además de en f_8 , es necesario conocer su valor para poder calcular otras variables desconocidas. Por tanto, se pasa la ecuación f_8 a la segunda fila de la matriz y la variable a_{v2} a la segunda columna. Lo mismo ocurre con la ecuación f_9 y la variable a_{v2} que pasan a la tercera columna y fila.

	h_{sp}	a_{v1}	a_{v2}	F_{in}	F_{out1}	F_{out2}	e	u	$derh$	$derI$
$f_1(h_{sp})$	X	0	0	0	0	0	0	0	0	0
$f_8(a_{v1})$	0	X	0	0	0	0	0	0	0	0
$f_9(a_{v2})$	0	0	X	0	0	0	0	0	0	0
$f_2(F_{in}, u)$	0	0	0	X	0	0	0	X	0	0
$f_3(F_{out1}, a_{v1}, h)$	0	X	0	0	X	0	0	0	0	0
$f_4(F_{out2}, a_{v2}, h)$	0	0	X	0	0	X	0	0	0	0
$f_5(e, h, h_{sp})$	X	0	0	0	0	0	X	0	0	0
$f_6(derI, e)$	0	0	0	0	0	0	X	0	0	X
$f_7(u, e, I)$	0	0	0	0	0	0	X	X	0	0
$f_{10}(derh, F_{in}, F_{out1}, F_{out2})$	0	0	0	X	X	X	0	0	X	0

(26)

3. Como $derh$ y $derl$ intervienen en una única ecuación las ecuaciones f_{10} y f_6 . Deberá usarse esa ecuación para evaluarla. Como $derh$ y $derl$ no interviene en ninguna otra ecuación no es necesario conocer su valor para evaluar otras variables desconocidas. Por tanto, se pasan las ecuaciones f_{10} y f_6 a la última y penúltima fila y columna respectivamente.

	h_{sp}	a_{v1}	a_{v2}	F_{in}	F_{out1}	F_{out2}	e	u	$derl$	$derh$
$f_1(h_{sp})$	[X]	0	0	0	0	0	0	0	0	0
$f_8(a_{v1})$	0	[X]	0	0	0	0	0	0	0	0
$f_9(a_{v2})$	0	0	[X]	0	0	0	0	0	0	0
$f_2(F_{in}, u)$	0	0	0	X	0	0	0	X	0	0
$f_3(F_{out1}, a_{v1}, h)$	0	X	0	0	X	0	0	0	0	0
$f_4(F_{out2}, a_{v2}, h)$	0	0	X	0	0	X	0	0	0	0
$f_5(e, h, h_{sp})$	X	0	0	0	0	0	X	0	0	0
$f_7(u, e, I)$	0	0	0	0	0	0	X	X	0	0
$f_6(derl, e)$	0	0	0	0	0	0	X	0	[X]	0
$f_{10}(derh, F_{in}, F_{out1}, F_{out2})$	0	0	0	X	X	X	0	0	0	[X]

(27)

4. Asumiendo que la variable h_{sp} ya han sido evaluadas, la ecuación f_5 del cálculo del error ya solo tiene una variable desconocida no evaluada: la variable e . Ubicando esta ecuación en la cuarta fila y la variable e en la cuarta columnas se obtiene a continuación (28).

	h_{sp}	a_{v1}	a_{v2}	e	F_{in}	F_{out1}	F_{out2}	u	$derl$	$derh$
$f_1(h_{sp})$	[X]	0	0	0	0	0	0	0	0	0
$f_8(a_{v1})$	0	[X]	0	0	0	0	0	0	0	0
$f_9(a_{v2})$	0	0	[X]	0	0	0	0	0	0	0
$f_5(e, h, h_{sp})$	X	0	0	[X]	0	0	0	0	0	0
$f_2(F_{in}, u)$	0	0	0	0	X	0	0	X	0	0
$f_3(F_{out1}, a_{v1}, h)$	0	X	0	0	0	X	0	0	0	0
$f_4(F_{out2}, a_{v2}, h)$	0	0	X	0	0	0	X	0	0	0
$f_7(u, e, I)$	0	0	0	X	0	0	0	X	0	0
$f_6(derl, e)$	0	0	0	X	0	0	0	0	[X]	0
$f_{10}(derh, F_{in}, F_{out1}, F_{out2})$	0	0	0	0	X	X	X	0	0	[X]

(28)

5. Asumiendo que la variable e ya ha sido evaluada, la ecuación f_7 del cálculo del voltaje ya solo tiene una variable desconocida no evaluada: la variable u . Ubicando esta ecuación en la quinta fila y la variable u en la quinta columna se obtiene a continuación (29).

	h_{sp}	a_{v1}	a_{v2}	e	u	F_{in}	F_{out1}	F_{out2}	$derl$	$derh$
$f_1(h_{sp})$	[X]	0	0	0	0	0	0	0	0	0
$f_8(a_{v1})$	0	[X]	0	0	0	0	0	0	0	0
$f_9(a_{v2})$	0	0	[X]	0	0	0	0	0	0	0
$f_5(e, h, h_{sp})$	X	0	0	[X]	0	0	0	0	0	0
$f_7(u, e, I)$	0	0	0	X	[X]	0	0	0	0	0
$f_2(F_{in}, u)$	0	0	0	0	X	X	0	0	0	0
$f_3(F_{out1}, a_{v1}, h)$	0	X	0	0	0	0	X	0	0	0
$f_4(F_{out2}, a_{v2}, h)$	0	0	X	0	0	0	0	X	0	0
$f_6(derl, e)$	0	0	0	X	0	0	0	0	[X]	0
$f_{10}(derh, F_{in}, F_{out1}, F_{out2})$	0	0	0	0	0	X	X	X	0	[X]

(29)

6. Una vez la variable u ya ha sido evaluada, las ecuaciones f_2, f_3 y f_4 pueden evaluar al solo tener una variable desconocida: $F_{in}, F_{out1}, F_{out2}$ respectivamente. Quedando la matriz de incidencia en forma **BLT** (29).

	h_{sp}	a_{v1}	a_{v2}	e	u	F_{in}	F_{out1}	F_{out2}	$derI$	$derh$
$f_1(h_{sp})$	[X]	0	0	0	0	0	0	0	0	0
$f_8(a_{v1})$	0	[X]	0	0	0	0	0	0	0	0
$f_9(a_{v2})$	0	0	[X]	0	0	0	0	0	0	0
$f_5(e, h, h_{sp})$	X	0	0	[X]	0	0	0	0	0	0
$f_7(u, e, I)$	0	0	0	X	[X]	0	0	0	0	0
$f_2(F_{in}, u)$	0	0	0	0	X	[X]	0	0	0	0
$f_3(F_{out1}, a_{v1}, h)$	0	X	0	0	0	0	[X]	0	0	0
$f_4(F_{out2}, a_{v2}, h)$	0	0	X	0	0	0	0	[X]	0	0
$f_6(derI, e)$	0	0	0	X	0	0	0	0	[X]	0
$f_{10}(derh, F_{in}, F_{out1}, F_{out2})$	0	0	0	0	0	X	X	X	0	[X]

(30)

Por tanto, las ecuaciones del modelo ordenadas y con la causalidad computacional señalada son:

$$[h_{sp}] = 5 * \alpha + \left(\frac{t + 30}{30}\right) * \beta + \left(\frac{-t + 30}{30}\right) * \gamma \quad (31)$$

$$[a_{v1}] = 0.5 * \operatorname{sgn}\left(-\operatorname{sen}\left(\frac{\pi * t}{30}\right)\right) + 0.5 \quad (32)$$

$$[a_{v2}] = 0.5 * \operatorname{sgn}\left(\operatorname{sen}\left(\frac{\pi * t}{30}\right)\right) + 0.5 \quad (33)$$

$$[e] = h_{sp} - h \quad (34)$$

$$[u] = k_p * e + \frac{1}{k_I} * I \quad (35)$$

$$[F_{in}] = B * \min(20, \max(0, u)) \quad (36)$$

$$[F_{out1}] = K * a_{v1} * \sqrt{h - h_1} \quad (37)$$

$$[F_{out2}] = K * a_{v2} * \sqrt{h - h_2} \quad (38)$$

$$[derI] = e \quad (39)$$

$$A * [derh] = F_{in} - (F_{out1} + F_{out2}) \quad (40)$$

Despejando en cada ecuación la variable a evaluar, se obtiene el modelo ordenado y resuelto:

$$[h_{sp}] = 5 * \alpha + \left(\frac{t + 30}{30}\right) * \beta + \left(\frac{-t + 30}{30}\right) * \gamma \quad (41)$$

$$[a_{v1}] = 0.5 * \operatorname{sgn}\left(-\operatorname{sen}\left(\frac{\pi * t}{30}\right)\right) + 0.5 \quad (42)$$

$$[a_{v2}] = 0.5 * \operatorname{sgn}\left(\operatorname{sen}\left(\frac{\pi * t}{30}\right)\right) + 0.5 \quad (43)$$

$$[e] = h_{sp} - h \quad (44)$$

$$[u] = k_p * e + \frac{1}{k_I} * I \quad (45)$$

$$[F_{in}] = B * \min(20, \max(0, u)) \quad (46)$$

$$[F_{out1}] = K * a_{v1} * \sqrt{h - h_1} \quad (47)$$

$$[F_{out2}] = K * a_{v2} * \sqrt{h - h_2} \quad (48)$$

$$[derI] = e \quad (49)$$

$$[derh] = \frac{F_{in} - (F_{out1} + F_{out2})}{A} \quad (50)$$

2.4. Diagrama de flujo

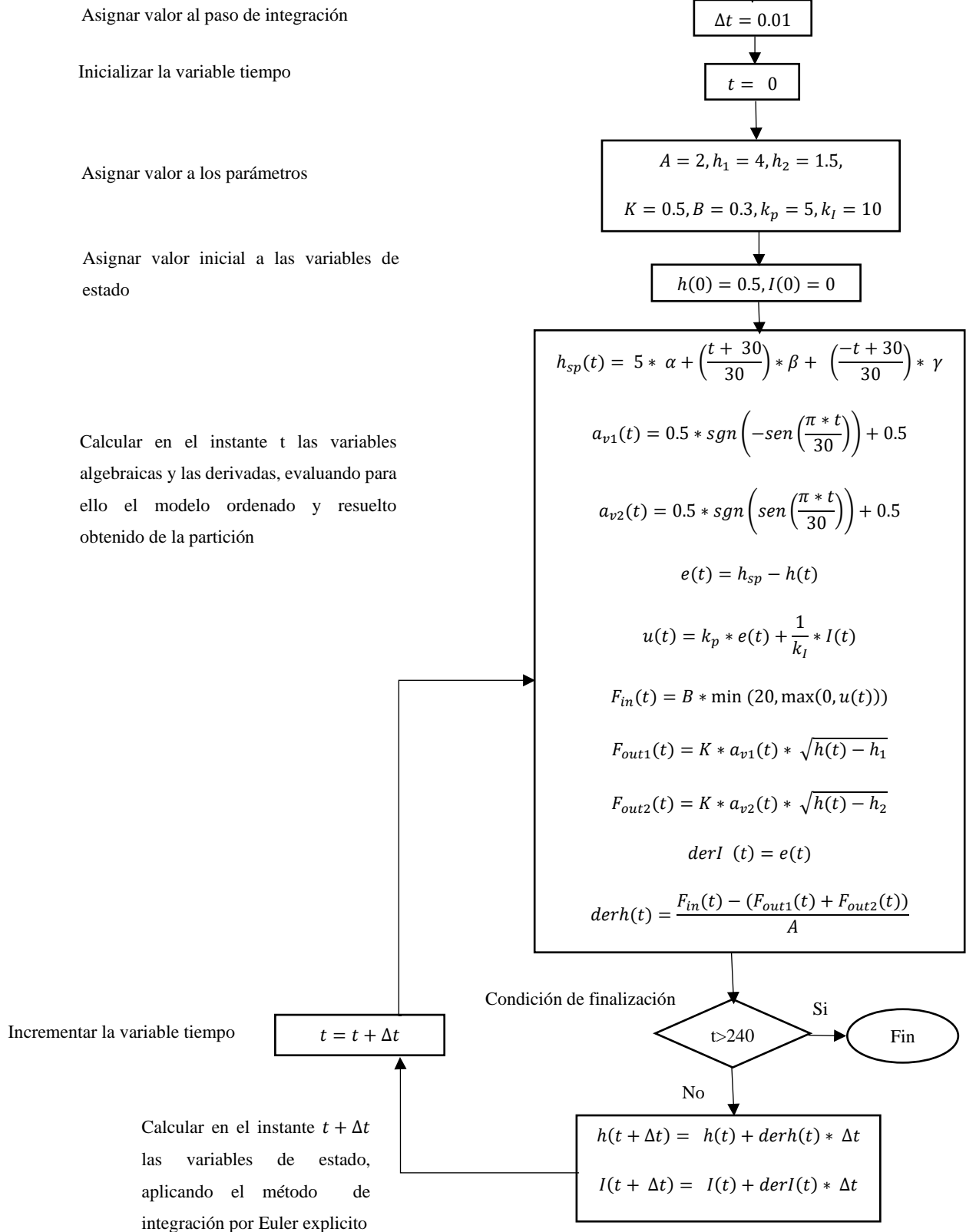


Ilustración 4. Diagrama de flujo del sistema de cilindro y válvulas

En la ilustración 4 se muestra el algoritmo para la simulación del modelo de tiempo continuo del sistema. El algoritmo de integración empleado es el método de Euler explícito. Puede comprobarse que la clasificación de las variables del modelo en parámetros, variables de estado y variables algebraicas constituye la base para la simulación del modelo:

- Al comenzar la simulación, se asigna valores a los parámetros. Estos valores permanecen constantes durante toda la simulación.
- Las variables de estado son calculadas mediante la integración numérica de sus derivadas.
- El valor de las variables algebraicas y las derivadas de las variables de estado se calcula en cada instante de tiempo, conocido el valor de los parámetros y el valor de las variables de estado en ese instante de tiempo, evaluando el modelo ordenado y resuelto resultante de aplicar el algoritmo de la partición.

2.5. Script en R

2.5.1. Codigo

```
t_fin <- 240 # Valor final del tiempo
h      <- 0.1 # Tamaño del paso de integración
t_com <- 0.5 # Intervalo de comunicación
# Parámetros
  A  <- 2
  h1 <- 4
  h2 <- 1.5
  K  <- 0.5
  B  <- 0.3
  kp <- 5
  kI <- 10
# Inicialización del tiempo
  t  <- 0
# Valor inicial del estado
  hr <- 0.5
  I  <- 0
# Inicialización resultados salida
  t_plot      <- numeric(0)
  hsp_plot    <- numeric(0)
  hr_plot     <- numeric(0)
  av1_plot    <- numeric(0)
  av2_plot    <- numeric(0)
  e_plot      <- numeric(0)
  u_plot      <- numeric(0)
  Fin_plot    <- numeric(0)
  Fout1_plot  <- numeric(0)
  Fout2_plot  <- numeric(0)
  t_ultimaCom <- -Inf
# Bucle de la simulación
termina <- FALSE
while ( !termina ) {
  termina <- (t + h/2 > t_fin)
  # Cálculo variables algebraicas en t
  #   Calculo de la altura de referencia hsp
  if(t<60){
    hsp<-5
  }else if(t>=60 && t<120){
    hsp<-(t+30)/30
  }else if(t>=120&&t<180){
    hsp<-(-t+270)/30
  }else{
```

```

    hsp<-5
}
#   Calculo del estado de la valvula 1
av1<-0.5*sign(-sin(pi*t/30))+0.5
#   Calculo del estado de la valvula 1
av2<-0.5*sign(sin(pi*t/30))+0.5
#   Calculo del error de control
e<-(hsp-hr)
#   Calculo de la salida del controlador
u<-(kp*e+I/kI)
#   Calculo del Flujo de entrada
Fin<-B*min(20,max(0,u))
#   Calculo del Flujo de salida 1
if(av1<=0){
    Fout1<-0
}else{
    if(hr<=h1){
        Fout1<-0
    }else{
        Fout1<-K*av1*sqrt(hr-h1)
    }
}
#   Calculo del Flujo de salida 2
if(av2<=0){
    Fout2<-0
}else{
    if(hr<=h2){
        Fout2<-0
    }else{
        Fout2<-K*av2*sqrt(hr-h2)
    }
}
#   Calculo de derI
derI<-e1
#   Calculo de derh
derh<-(Fin-Fout1-Fout2)/A
# Almacenar resultados de salida
if ( t - t_ultimaCom + h/2 > t_com | termina ) {
t_plot      <- c(t_plot, t)
hsp_plot    <- c(hsp_plot, hsp)
hr_plot     <- c(hr_plot, hr)
av1_plot    <- c(av1_plot, av1)
av2_plot    <- c(av2_plot, av2)
e_plot      <- c(e_plot, e)
u_plot      <- c(u_plot, u)

```

```

Fin_plot    <- c(Fin_plot, Fin)
Fout1_plot  <- c(Fout1_plot, Fout1)
Fout2_plot  <- c(Fout2_plot, Fout2)

t_ultimaCom <- t
}
# Valor en t+h de los estados
hr <- hr + h*derh
I  <- I + h*derI
# Actualización del valor del tiempo
t  <- t + h
}
# Representación gráfica
# Para poder usar minor.tick es necesario instalar el paquete Hmisc
# Configuración de la representación gráfica
par( mfrow = c(2,1) )
# hsp vs t
par( mai = c(0.4,1,0.8,0.2), cex = 1.2 )
plot(t_plot, hsp_plot, lty = 1, type = "l",
xlim = c(0,t_fin), ylim = c(0,5),
yaxt = "n", xlab = "", ylab = "hsp [m]", lwd = 2 )
axis(2, las = 2)
minor.tick(nx = 2, ny = 5, tick.ratio = 0.5)
abline(v = seq(0,t_fin,10*t_com), h = seq(0,5,0.2),
lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)
title("Altura de referencia", cex = 1)
# hr vs t
par( mai = c(0.4,1,0.8,0.2), cex = 1.2 )
plot(t_plot, hr_plot, lty = 1, type = "l",
xlim = c(0,t_fin), ylim = c(0,5),
yaxt = "n", xlab = "", ylab = "hr [m]", lwd = 2 )
axis(2, las = 2)
minor.tick(nx = 2, ny = 5, tick.ratio = 0.5)
abline(v = seq(0,t_fin,10*t_com), h = seq(0,5,0.2),
lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)
title("Altura real", cex = 1)

dev.new()
par( mfrow = c(2,1) )

# Fin vs t
par( mai = c(0.4,1,0.8,0.2), cex = 1.2 )
plot(t_plot, Fin_plot, lty = 1, type = "l",

```

```

xlim = c(0,t_fin), ylim = c(0,5),
yaxt = "n", xlab = "", ylab = "Fin [kg/s]", lwd = 2 )
axis(2, las = 2)
minor.tick(nx = 2, ny = 5, tick.ratio = 0.5)
abline(v = seq(0,t_fin,10*t_com), h = seq(0,5,0.2),
lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)
title("Flujo de entrada", cex = 1)

dev.new()
par( mfrow = c(2,1) )
# Fout1 vs t
par( mai = c(0.4,1,0.8,0.2), cex = 1.2 )
plot(t_plot, Fout1_plot, lty = 1, type = "l",
xlim = c(0,t_fin), ylim = c(0,5),
yaxt = "n", xlab = "", ylab = "Fout1 [kg/s]", lwd = 2 )
axis(2, las = 2)
minor.tick(nx = 2, ny = 5, tick.ratio = 0.5)
abline(v = seq(0,t_fin,10*t_com), h = seq(0,5,0.2),
lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)
title("Flujo de salida 1", cex = 1)

# Fout2 vs t
par( mai = c(0.4,1,0.8,0.2), cex = 1.2 )
plot(t_plot, Fout2_plot, lty = 1, type = "l",
xlim = c(0,t_fin), ylim = c(0,5),
yaxt = "n", xlab = "", ylab = "Fout2 [kg/s]", lwd = 2 )
axis(2, las = 2)
minor.tick(nx = 2, ny = 5, tick.ratio = 0.5)
abline(v = seq(0,t_fin,10*t_com), h = seq(0,5,0.2),
lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)
title("Flujo de salida 2", cex = 1)

dev.new()
par( mfrow = c(2,1) )
# av1 vs t
par( mai = c(0.4,1,0.8,0.2), cex = 1.2 )
plot(t_plot, av1_plot, lty = 1, type = "l",
xlim = c(0,t_fin), ylim = c(0,5),
yaxt = "n", xlab = "", ylab = "av1", lwd = 2 )
axis(2, las = 2)
minor.tick(nx = 2, ny = 5, tick.ratio = 0.5)
abline(v = seq(0,t_fin,10*t_com), h = seq(0,5,0.2),

```

```
lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)
title("Estado de la valvula 1", cex = 1)

# av2 vs t
par( mai = c(0.4,1,0.8,0.2), cex = 1.2 )
plot(t_plot, av2_plot, lty = 1, type = "l",
xlim = c(0,t_fin), ylim = c(0,5),
yaxt = "n", xlab = "", ylab = "av2", lwd = 2 )
axis(2, las = 2)
minor.tick(nx = 2, ny = 5, tick.ratio = 0.5)
abline(v = seq(0,t_fin,10*t_com), h = seq(0,5,0.2),
lwd = 0.5, lty = 3)
abline(h = 0, lwd = 1, lty = 1)
title("Estado de la valvula 2", cex = 1)
```

Se ha seleccionado un tamaño del paso igual a 0.1 s. Para ello, se ha repetido la simulación empleando diferentes tamaños del paso, encontrándose que el error cometido escogiendo 0.1 s es admisible para el propósito de este estudio. Se muestra a continuación el código R que implementa el algoritmo mostrado en la Figura 1.3, con un intervalo de comunicación de 0.5 s. En las siguientes ilustraciones se muestran las gráficas obtenidas al ejecutar el código R.

2.5.2. Representación grafica

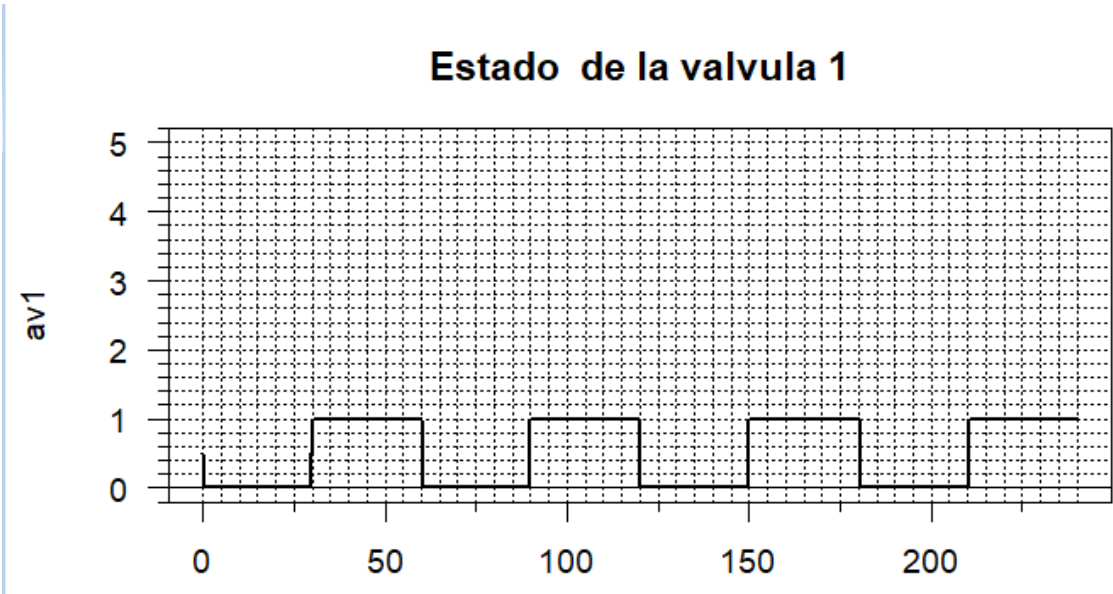


Ilustración 5. Estado de la válvula 1

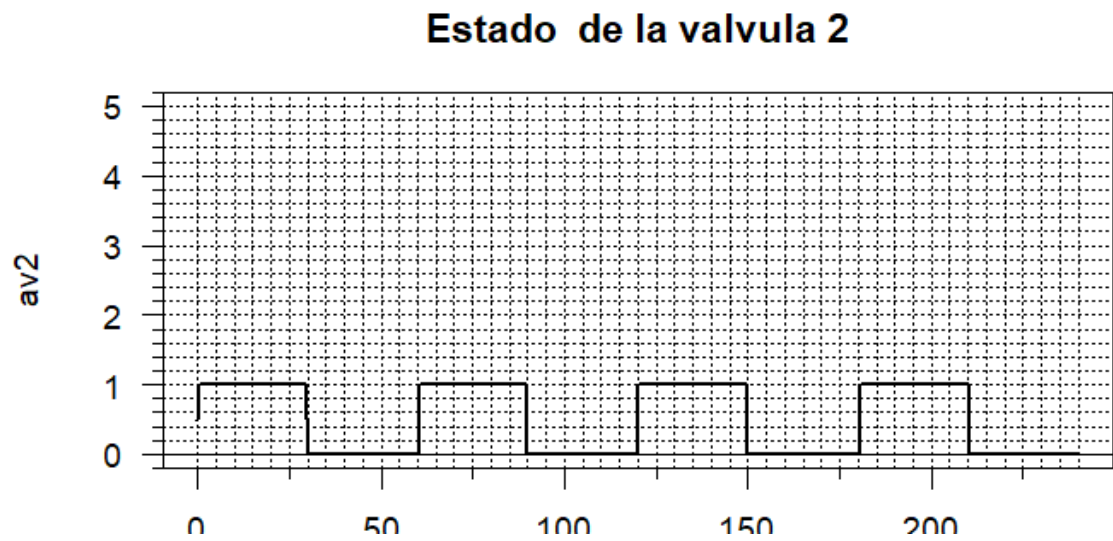


Ilustración 6. Estado de la válvula 2

Flujo de salida 1

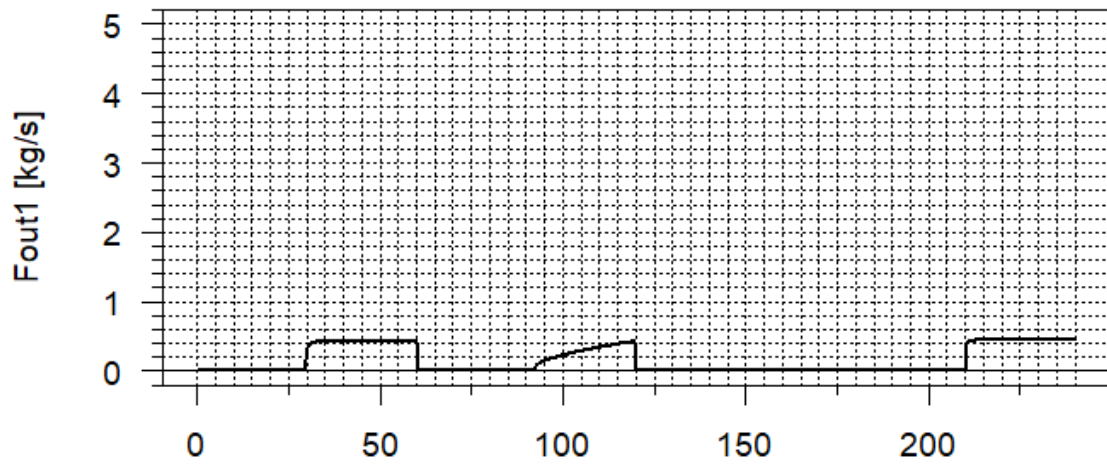


Ilustración 7. Flujo de salida 1

Flujo de salida 2

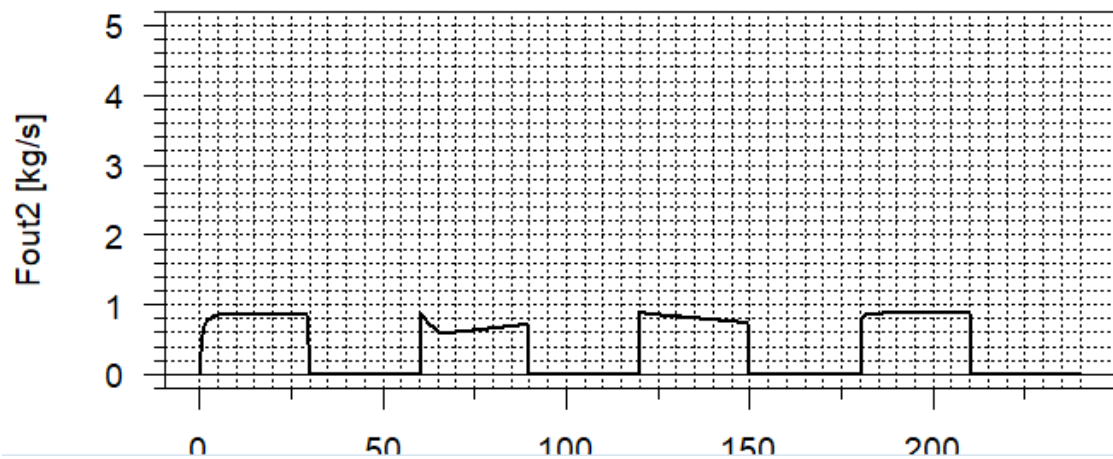


Ilustración 8. Flujo de salida 2

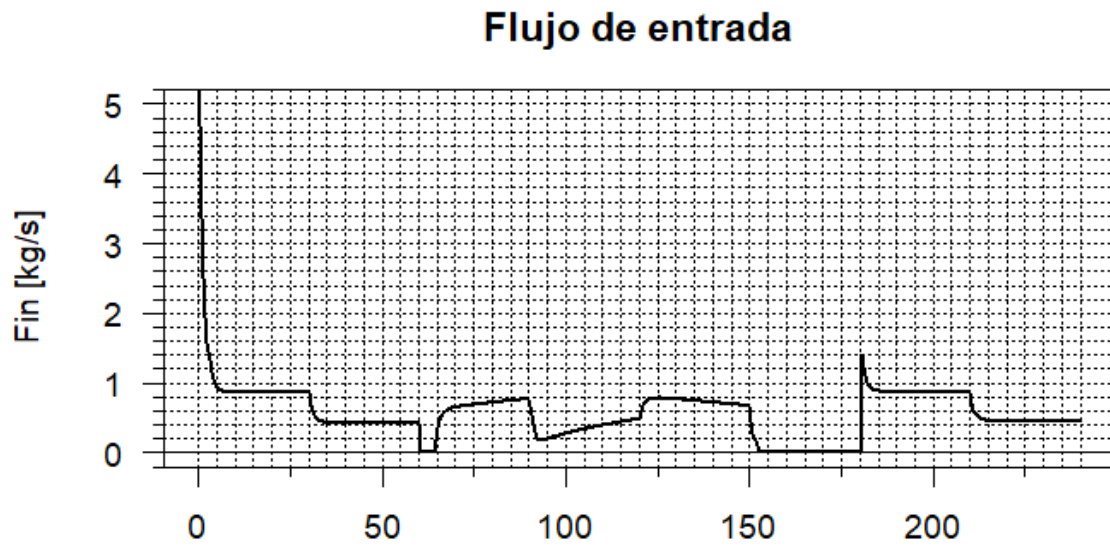


Ilustración 9. Flujo de entrada

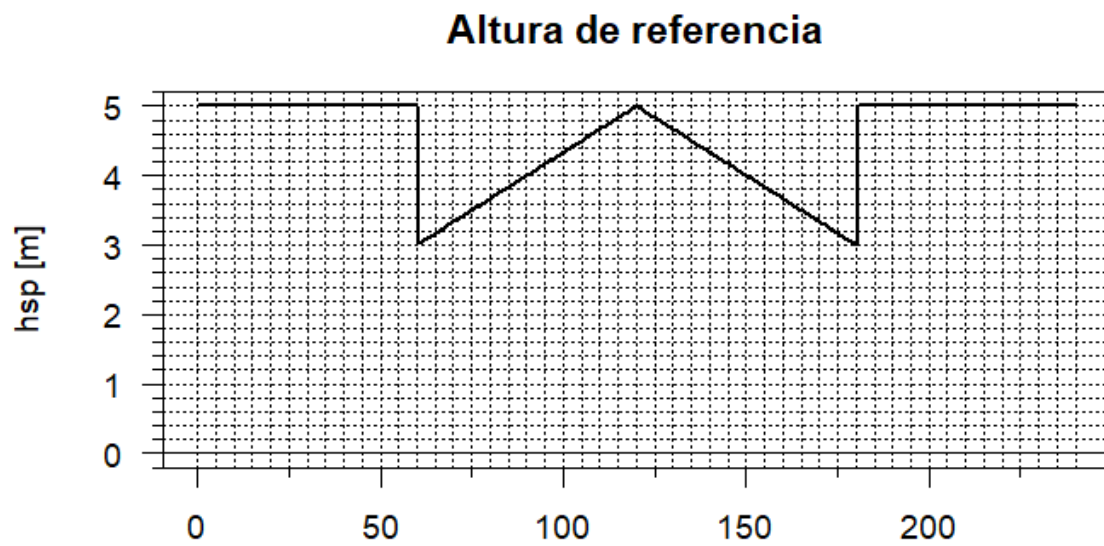


Ilustración 10. Altura de referencia

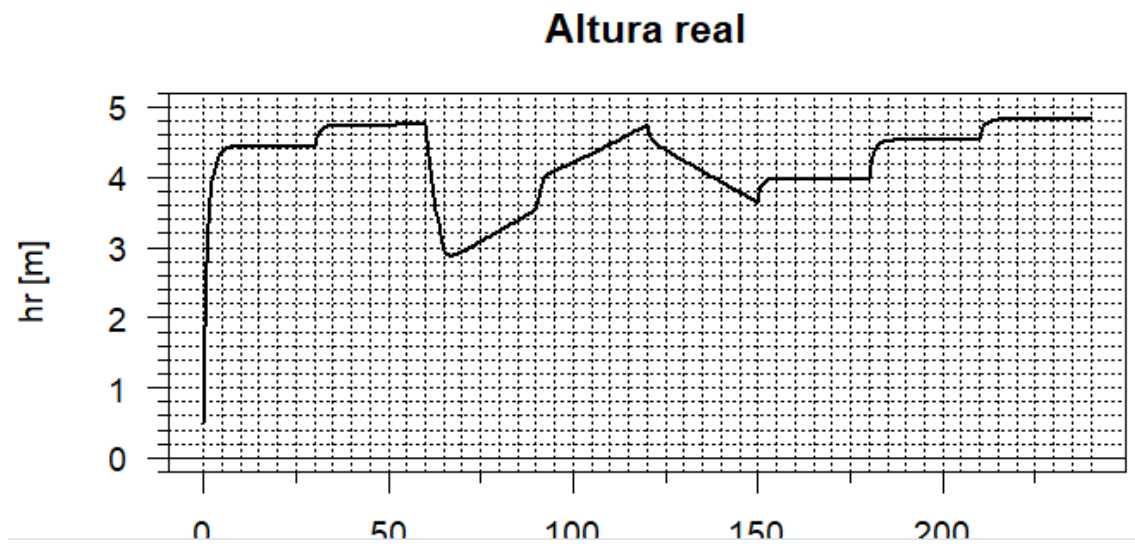


Ilustración 11. Altura real

3. Ejercicio 3

3.1. Enunciado

Describa en lenguaje modélica el sistema del ejercicio anterior, de las dos maneras siguientes:

1. Como un modelo atómico, que viene descrito por las ecuaciones que usted ha planteado al contestar a la pregunta anterior.
2. Programe una librería que contenga los componentes necesarios para componer el sistema descrito en el Ejercicio 2. A continuación, defina dicho sistema como un modelo compuesto, instanciando y conectando componentes de la librería que ha creado.

Asigne a los parámetros los valores indicados en el ejercicio anterior y simule los dos modelos anteriores durante 240 s.

Represente gráficamente frente al tiempo las variables siguientes: la altura de líquido, su valor de consigna, los flujos de entrada y salida, y las variables que describen las fases de las válvulas.

3.2. Modelo atómico

La descripción en lenguaje Modelica del sistema se muestra en el código siguiente:

```

model ControlDeposito
  import SI = Modelica.SIunits;
  import Modelica.Math.*;
  import Modelica.Blocks.Math.*;

  SI.Position      e      "href-h, calculado en el controlador";
  SI.MassFlowRate  Fin     "Flujo masico de entrada al deposito";
  SI.MassFlowRate  Fout1   "Flujo masico de salida del deposito 1";
  SI.MassFlowRate  Fout2   "Flujo masico de salida del deposito 2";
  Integer          av1     "Estado de la valvula 1 0=cerrada 1=abierta";
  Integer          av2     "Estado de la valvula 2 0=cerrada 1=abierta";
  Real             I  (unit="m.s", start=0, fixed=true) "Integral del error";
  SI.Height        h  (start=0.5, fixed=true) "Nivel en el depósito";
  SI.Height        hsp   "Consigna para el nivel";
  constant Real    pi = 2 * Modelica.Math.asin(1.0); // 3.14159265358979;
  parameter SI.Area A = 2      "Sección del deposito";
  parameter SI.Height h1 = 4    "Altura de la salida 1";
  parameter SI.Height h2 = 1.5  "Altura de la salida 2";
  parameter Real    K  (unit="mm") =5      "Parametro de la tuberia";
  parameter Real    B  (unit="m3/(s.V)") =0.3 "Parametro de la bomba";
  parameter Real    kf (unit="kg/(s.V)") = 100 "Parametro de la fuente";
  parameter Real    kI (unit="m.s/V") = 10 "Param. integral controlador";
  parameter Real    kP (unit="V/m") = 5     "Param. proporcional controlador";
  SI.Voltage        u      "Voltaje de salida del controlador";

  equation
    A*der(h) = Fin - (Fout1+Fout2);
    Fin      = B * min(20,max(0,u));
    Fout1    = if av1 <= 0
                then 0
                else
                  if h<=h1
                    then 0
                    else K*av1*sqrt(h-h1);
    Fout2    = if av2 <= 0
                then 0
                else
                  if h<=h2
                    then 0
                    else K*av2*sqrt(h-h2);

    av1      = 0.5*sign(-sin(pi*time/30))+0.5;
    av2      = 0.5*sign(sin(pi*time/30))+0.5;
    e        = hsp - h;

```

```
der(I)  = e;  
u       = kP*e + I/kI;  
hsp     = if time < 60  
          then 5  
          else if time >=60 and time < 120  
                then (time+30)/30  
          else if time >= 120 and time <180  
                then (-time+270)/30  
          else 5;  
end ControlDeposito;
```

3.3. Librería

```

package Deposito
  import SI = Modelica.SIunits;
  import Modelica.Math.*;
  import Modelica.Blocks.Math.*;

  package Interfaces
    connector Liquido
      SI.MassFlowRate Fm "Flujo masico";
    end Liquido;
    connector EValvula
      Integer s "Estado de la valvula";
    end EValvula;
    connector Signal
      Real s;
    end Signal;
    connector Altura
      SI.Height h;
    end Altura;
  end Interfaces;

  package Componentes
    model Valvula
      constant Real pi = 2 * Modelica.Math.asin(1.0); // 3.14159265358979;
      Interfaces.EValvula av1;
      Interfaces.EValvula av2;
      equation
        av1.s = 0.5*sign(-sin(pi*time/30))+0.5;
        av2.s = 0.5*sign(sin(pi*time/30))+0.5;
      end Valvula;

      model Fuente
        Interfaces.Signal u;
        Interfaces.Liquido Fin;
        parameter Real B (unit="m3/(s.V)") "Parametro de la bomba";
        equation
          Fin.Fm = B * min(20,max(0,u.s));
        end Fuente;

      model Deposito
        Interfaces.Liquido Fin;
        Interfaces.Liquido Fout1;
        Interfaces.Liquido Fout2;
        Interfaces.Altura h;
      end Deposito;
    end Componentes;
  end Deposito;

```

```

parameter SI.Area   A      "Sección del deposito";

equation
der(h.h) = (Fin.Fm - Fout1.Fm - Fout2.Fm)/A;
end Deposito;

model Salida
  Interfaces.Liquido Fout;
  Interfaces.Altura h; //altura del liquido
  parameter SI.Height hv; //altura de la valvula
  Interfaces.EValvula av; //estado de la valvula
  parameter Real      K (unit="mm")          "Parametro de la tuberia";
  algorithm
    Fout.Fm := if av.s<=0
                then 0
            else
                if h.h<=hv
                    then 0
                else K*av.s*sqrt(h.h-hv);
            end if;
  end Salida;

model ConsignaNivel
  Interfaces.Altura hsp;
  equation
    hsp.h = if time < 60
                then 5
            else if time >=60 and time < 120
                then (time+30)/30
            else if time >= 120 and time <180
                then (-time+270)/30
            else 5;
  end ConsignaNivel;

model ControladorPI
  Interfaces.Signal u;
  Interfaces.Altura hsp;
  Interfaces.Altura h;
  parameter Real    kP "Parametro proporcional del controlador";
  parameter Real    kI "Param. integral controlador";
  protected
    SI.Position      e "href-h,calculado en el controlador";
    Real             I (unit="m.s", start=0, fixed=true) "Integral del error";
  equation
    e = hsp.h - h.h;
    der(I) = e;

```

```

        u.s      = kP*e + I/kI;
    end ControladorPI;

end Componentes;

model Control
    import Deposito.Componentes.*;
    parameter Real      kf (unit="kg/(s.V)") = 100 "Parametro de la fuente";
    parameter Real      kI (unit="m.s/V") = 10 "Param. integral controlador";
    parameter Real      kP (unit="V/m") = 5 "Param. proporcional controlador";
    parameter SI.Height h1 = 4.0 "Altura de la salida 1";
    parameter SI.Height h2 = 1.5 "Altura de la salida 2";
    parameter Real      B (unit="m3/(s.V)") = 0.3 "Parametro de la bomba";
    parameter Real      K (unit="mm") = 5 "Parametro de la tuberia";
    parameter SI.Area    A = 2 "Sección del deposito";
    Deposito      deposito(A=A);
    Fuente        fuente (B=B);
    ControladorPI LC (kI=kI,kP=kP);
    ConsignaNivel nivelSP;
    Salida        salida1 (hv=h1,K=K);
    Salida        salida2 (hv=h2,K=K);
    Valvula       valvulas;
    equation
        connect(nivelSP.hsp, LC.hsp); //nivel con controlador
        connect(deposito.Fin,fuente.Fin);
        connect(deposito.Fout1,salida1.Fout);
        connect(deposito.Fout2,salida2.Fout);
        connect(salida1.av,valvulas.av1);
        connect(salida1.h,deposito.h);
        connect(salida2.av,valvulas.av2);
        connect(salida2.h,deposito.h);
        connect(LC.u,fuente.u); //controlador con la fuente
        connect(LC.h,deposito.h);
    end Control;
end Deposito;

```

3.4. Representación grafica

En las ilustraciones siguientes se representa la altura de líquido, su valor de consigna, los flujos de entrada y salida, y las variables que describen las fases de las válvulas durante un tiempo (t) de 240s.

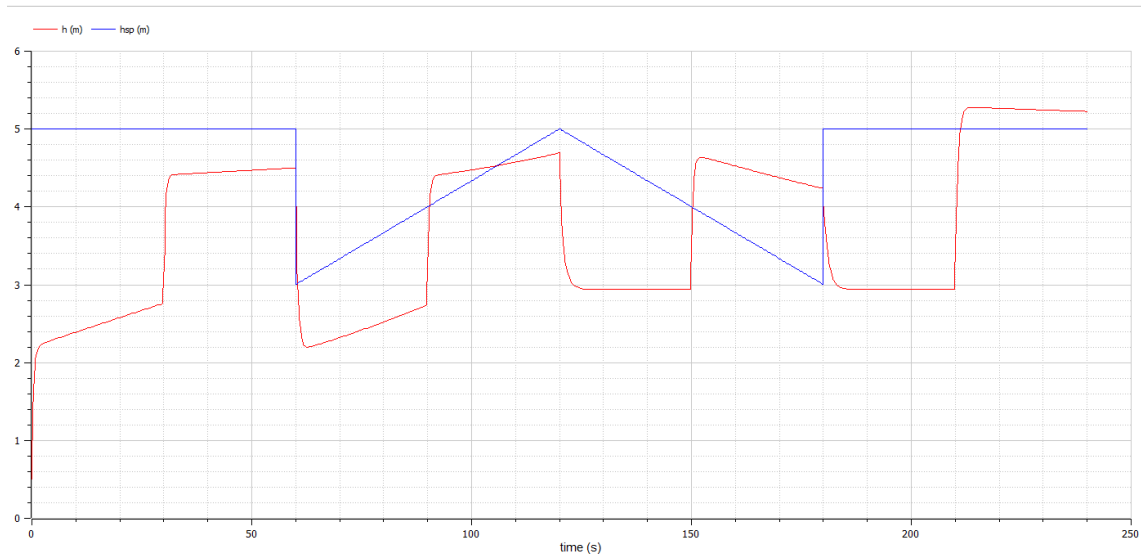


Ilustración 12. Representación de la altura del líquido junto con su valor consigna

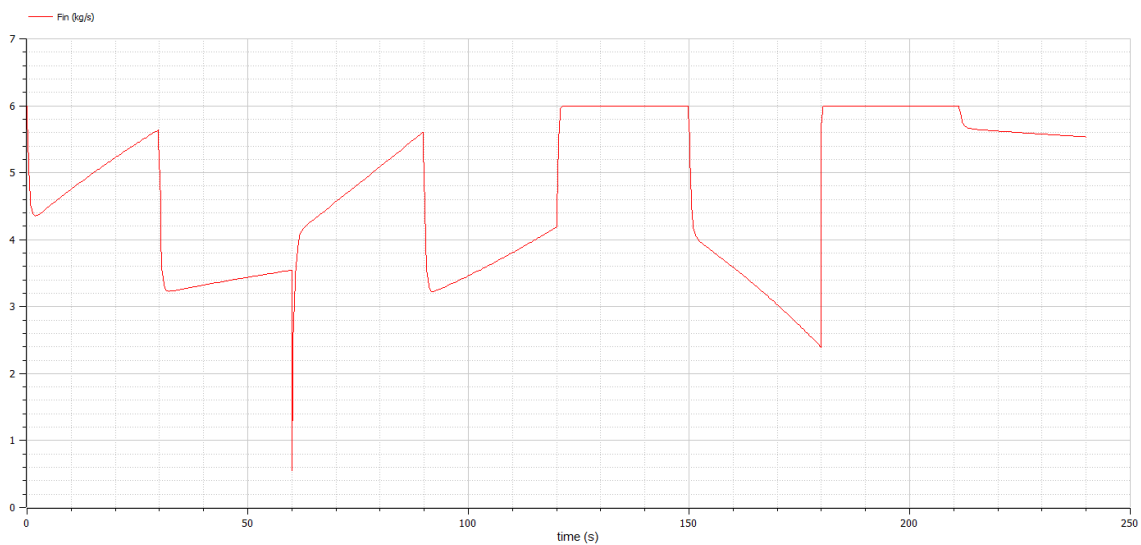


Ilustración 13. Flujo de Entrada

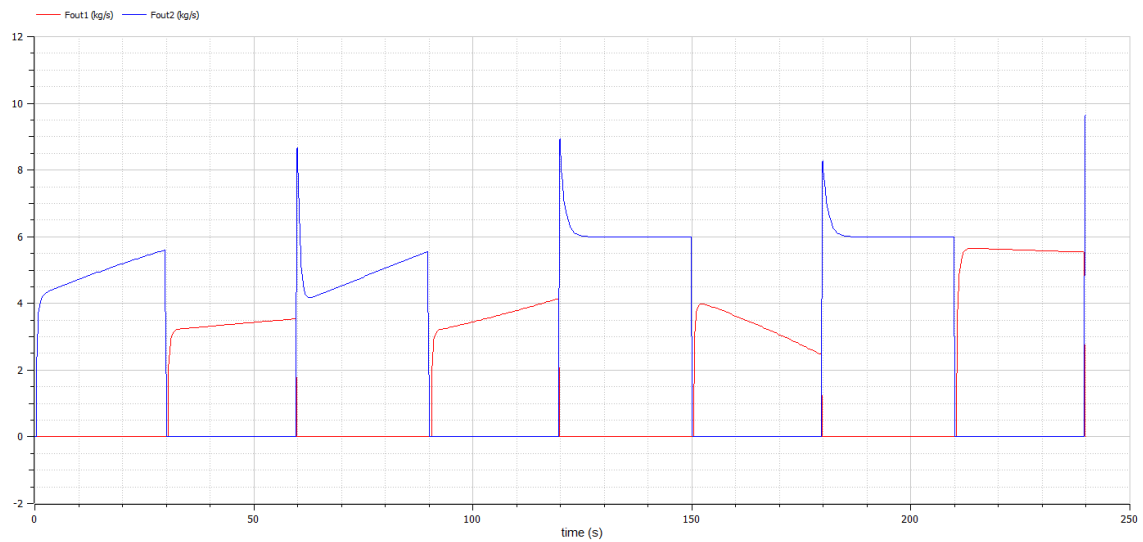


Ilustración 14. Flujos de salida

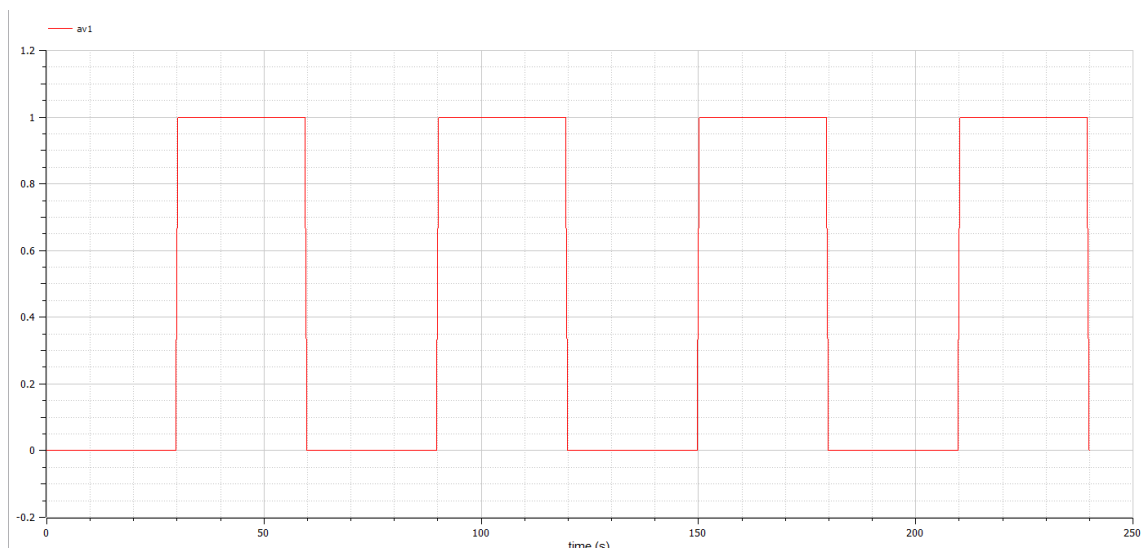


Ilustración 15. Variable que describe la fase de la válvula 1

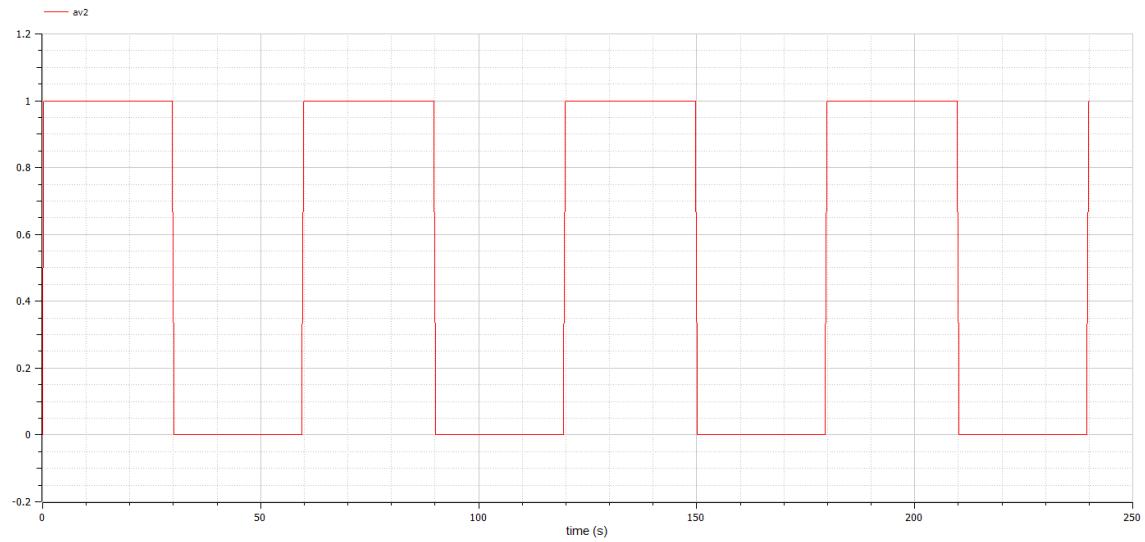


Ilustración 16. Variable que describe la fase de la válvula 2

4. Ejercicio 4

4.1. Enunciado

Escriba un script en FlexPDE para obtener la temperatura y los vectores correspondientes al flujo de calor del sistema descrito a continuación.

El sistema está formado por un bloque de madera de pino. El bloque de madera de pino tiene una conducción térmica anisótropa, siendo su constante de conductividad diferente en las direcciones x e y . La conductividad térmica en la dirección del eje X es $k_x = 0.15 \text{ W}/(\text{m} \cdot \text{K})$ y en la dirección del eje Y es $k_y = 0.36 \text{ W}/(\text{m} \cdot \text{K})$.

En la superficie superior del bloque de madera hay un calefactor eléctrico que tiene una anchura de 0.02 m . El calefactor produce calor a una tasa constante de $500 \text{ W}/\text{m}^2$. La superficie inferior del bloque de madera se mantiene a una temperatura constante de 300 K . El resto de lados del bloque de madera están bien aislados.

El sistema se trata como un problema bidimensional. En la figura se muestra una sección transversal, en la cual el bloque tiene una forma cuadrada de lado 0.2 m . La línea negra más gruesa indica la superficie que está calefactada.

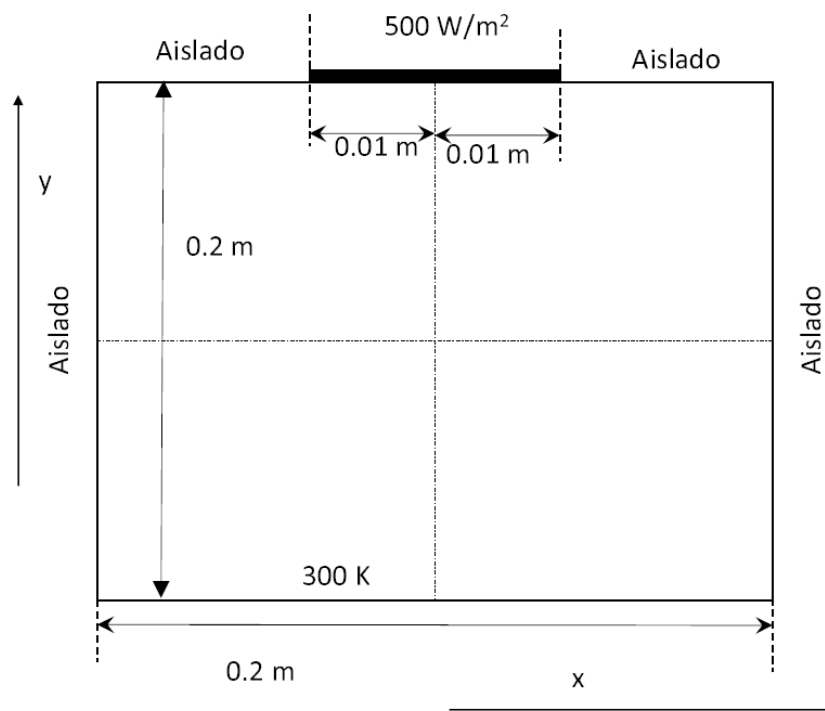


Ilustración 17. Sección del bloque de madera (dibujo no a escala)

Este fenómeno se describe mediante la ecuación diferencial siguiente:

$$\frac{\partial}{\partial x} \left(-k_x * \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(-k_y * \frac{\partial T}{\partial y} \right) = 0$$

donde T es la temperatura.

Escriba el código del script de FlexPDE para obtener un gráfico de las líneas de temperatura y un gráfico vectorial del flujo de calor. Las componentes x e y del flujo de calor son, respectivamente, las siguientes:

$$q_x = -k_x * \frac{\partial T}{\partial x}$$

$$q_y = -k_y * \frac{\partial T}{\partial y}$$

4.2. Código

```

TITLE 'Bloque Pino' { Flujo de calor en un bloque Pino }
SELECT
errlim=1e-3 spectral_colors
VARIABLES temp
DEFINITIONS
Lx=0.1 Ly = 0.1 Cx = 0.01 Cy = 0.01 !Posiciones
kx = 0.15 !Conductividad termica en X
ky = 0.36 !Conductividad termica en Y
fluxd_x = -kx*dx(temp) fluxd_y=-ky*dy(temp)
fluxd=vector(fluxd_x,fluxd_y)
INITIAL VALUES
temp = 273
EQUATIONS
dx( fluxd_x)+ dy( fluxd_y)=0
BOUNDARIES
region 'dominio' {Define el dominio del problema}
START (-Lx,Ly) NATURAL(temp) = 0
LINE TO (-Lx,-Ly) VALUE(temp) = 300
LINE TO (Lx, -Ly) NATURAL(temp) = 0
LINE TO (Lx, Ly) NATURAL(temp)= 0 LINE TO CLOSE
START(-Cx,Ly) LINE TO (Cx,Ly) {Region calefactor}
VALUE(temp) =500 LINE TO CLOSE

PLOTS

CONTOUR(temp)
VECTOR(fluxd)
end
END

```

4.3. Representación Grafica

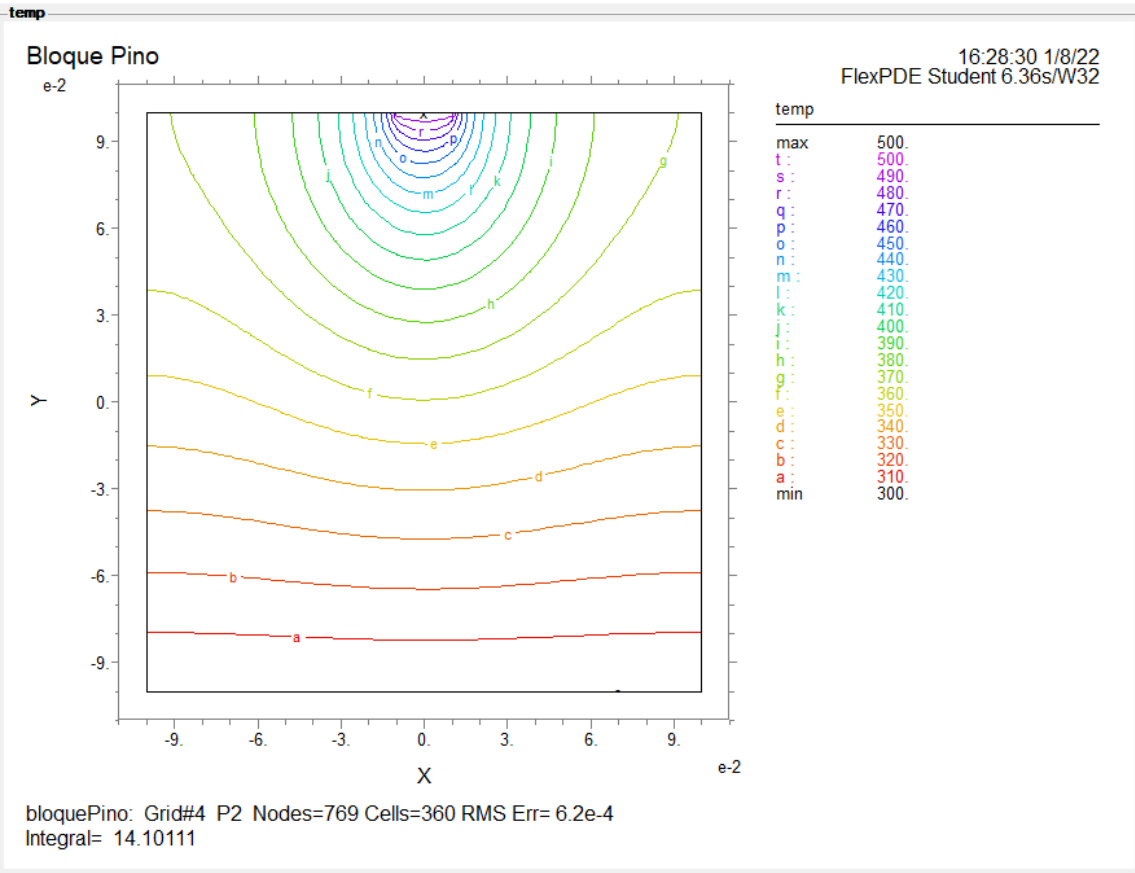


Ilustración 18. Curvas de Temperatura

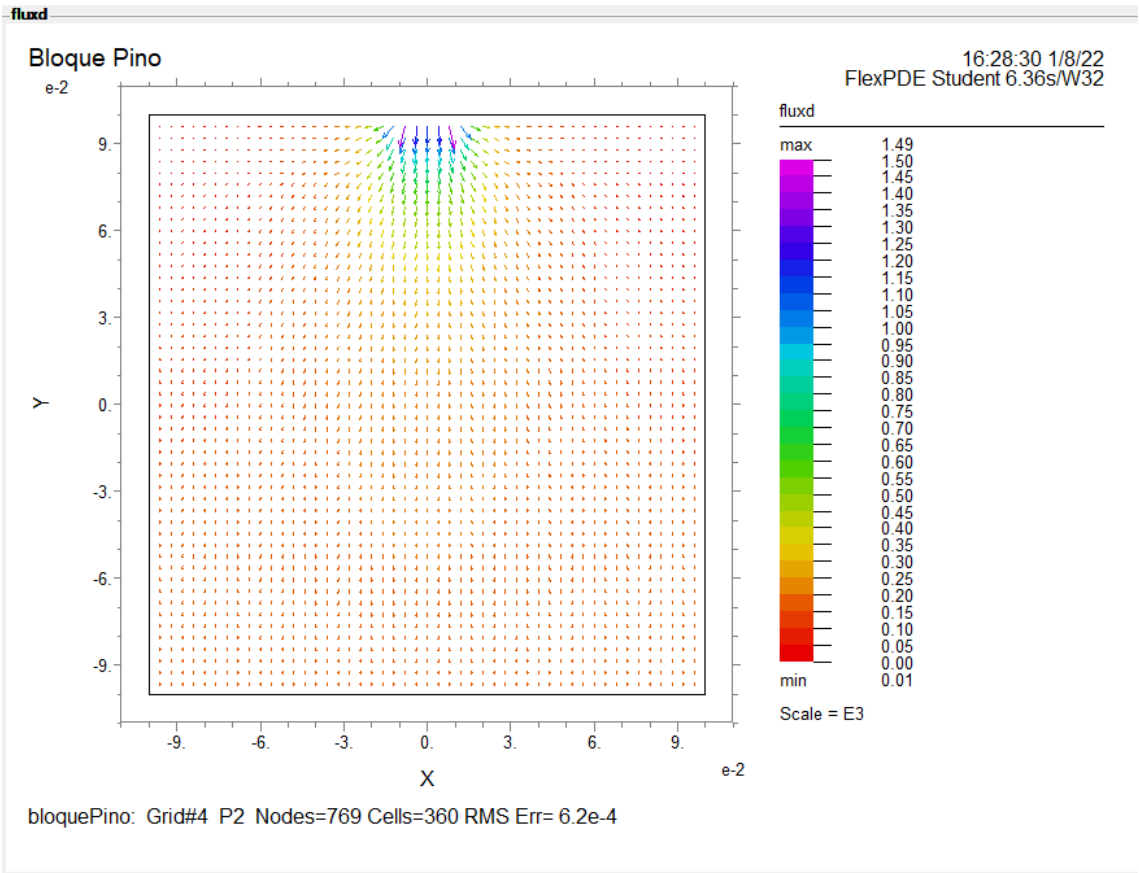


Ilustración 19. Flujo de calor

5. Bibliografía

- [1] Dpto. de Informática y Automática, “Trabajo Práctico Métodos de Simulación y Modelado - Convocatoria ordinaria del curso 2021/22,” 2021.