

Tema	Simulación de sistemas dinámicos no lineales
Actividad	Trabajo independiente
Asignatura	Modelado y Simulación
Docente	Carlos Felipe Rengifo Rodas
Email	caferen@unicauca.edu.co
Programa	Ingeniería en Automática Industrial
Facultad	Ingeniería Electrónica y Telecomunicaciones
Institución	Universidad del Cauca
Fecha	05 de abril de 2021

1. Propósito del trabajo

El objetivo del presente trabajo es desarrollar un simulador con interfaz de usuario para un intercambiador de calor a contracorriente. En estos sistemas, un fluido caliente transfiere energía térmica a un fluido frío que circula en dirección opuesta. La transferencia de calor se da a través de una tubería metálica, razón por la cual los fluidos no entran en contacto (Figura 1).

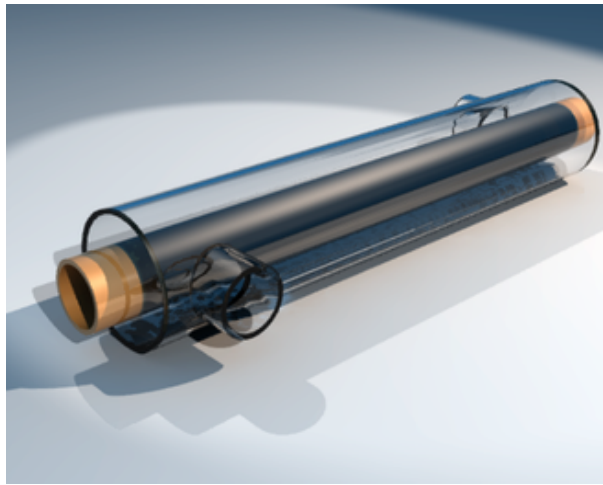


Figura 1: Intercambiador de calor a contracorriente. Imagen tomada de https://en.wikipedia.org/wiki/Heat_exchanger

2. Modelo del intercambiador de calor

En la Figura 2 se ilustra un intercambiador de calor a contracorriente, en el cual un líquido caliente fluye con caudal F_h para transferir calor a través de una tubería a un fluido frío que circula en la dirección opuesta con caudal F_c . Las temperaturas de entrada al intercambiador de los fluidos caliente y frío son T_{c0} y T_{h0} , respectivamente. En el sistema de la Figura 2, la temperatura varía con el tiempo y a lo largo de la tubería, por lo que el modelo resultante es una ecuación diferencial parcial, la cual no se puede resolver directamente utilizando *Simulink*. Para abordar este problema se utilizará una aproximación en la cual solo se consideran tres temperaturas a lo largo de la tubería (Figura 3). En dicho modelo se supone que las temperaturas y los volúmenes de los seis tanques son constantes, además se asume que las capacidades caloríficas y las densidades de los fluidos son independientes de la temperatura.

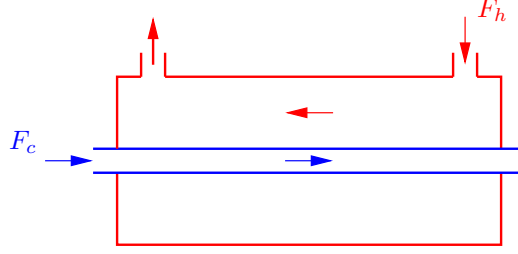


Figura 2: Intercambiador de calor a contracorriente.

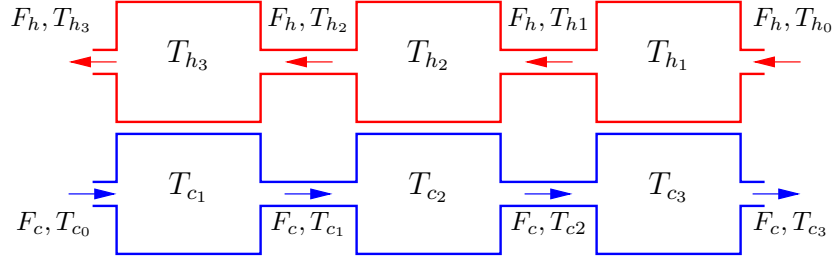


Figura 3: Modelo de parámetros concentrados del intercambiador de calor.

El modelo en espacio de estados del modelo de parámetros concentrados está dado por:

$$\begin{aligned}
 \rho_c V_c C_c \frac{dT_{c1}}{dt} &= \rho_c F_c C_c (T_{c0} - T_{c1}) + k (T_{h3} - T_{c1}) \\
 \rho_c V_c C_c \frac{dT_{c2}}{dt} &= \rho_c F_c C_c (T_{c1} - T_{c2}) + k (T_{h2} - T_{c2}) \\
 \rho_c V_c C_c \frac{dT_{c3}}{dt} &= \rho_c F_c C_c (T_{c2} - T_{c3}) + k (T_{h1} - T_{c3}) \\
 \rho_h V_h C_h \frac{dT_{h1}}{dt} &= \rho_h F_h C_h (T_{h0} - T_{h1}) + k (T_{c3} - T_{h1}) \\
 \rho_h V_h C_h \frac{dT_{h2}}{dt} &= \rho_h F_h C_h (T_{h1} - T_{h2}) + k (T_{c2} - T_{h2}) \\
 \rho_h V_h C_h \frac{dT_{h3}}{dt} &= \rho_h F_h C_h (T_{h2} - T_{h3}) + k (T_{c1} - T_{h3})
 \end{aligned} \tag{1}$$

Las constantes del fluido frío tienen el subíndice c que viene de la palabra *cold*, que en inglés significa frío. Dichas constantes son: la densidad $\rho_c = 1000 \text{ kg} \cdot \text{m}^{-3}$, el volumen de líquido en cada tanque $V_c = 0.2 \text{ m}^3$, la capacidad calorífica $C_c = 4180 \text{ J} \cdot \text{kg}^{-1} \cdot ^\circ\text{C}^{-1}$, el flujo $F_c = 0.05 \text{ m}^3 \cdot \text{min}^{-1}$, y la temperatura $T_{c0} = 20 ^\circ\text{C}$. Las constantes del fluido caliente tienen el subíndice h que viene de la palabra *hot*, que en inglés significa caliente. Dichas constantes son: la densidad $\rho_h = 1000 \text{ kg} \cdot \text{m}^{-3}$, el volumen de líquido en cada tanque $V_h = 0.2 \text{ m}^3$, la capacidad calorífica $C_h = 4180 \text{ J} \cdot \text{kg}^{-1} \cdot ^\circ\text{C}^{-1}$, el flujo $F_h = 0.11 \text{ m}^3 \cdot \text{min}^{-1}$, y la temperatura $T_{h0} = 90 ^\circ\text{C}$. La tasa de transferencia de calor del intercambiador es $k = 2 \times 10^5 \text{ J} \cdot ^\circ\text{C}^{-1} \cdot \text{min}^{-1}$.

3. Requerimientos

1. (10%) La interfaz de usuario debe contener un panel denominado *Parameters* (Figura 4), en el cual se especifican las constantes asociadas a los fluidos caliente y frío, además de la tasa de

transferencia de calor del intercambiador. Las cajas de dialogo deben contener los valores por defecto que toman los parámetros.

Figura 4: Panel donde se especifican las constantes del modelo (1).

2. (10%) La interfaz de usuario debe contener un panel denominado *Initial conditions* (Figura 5), en el cual se especifican las temperaturas iniciales de los fluidos caliente y frio. La temperatura indicada en la parte superior del panel se utiliza como condición inicial para las variables T_{h1} , T_{h2} , y T_{h3} , y la temperatura indicada en la parte inferior del panel se utiliza como condición inicial para las variables T_{c1} , T_{c2} , y T_{c3} . Las cajas de dialogo deben contener los valores por defecto que toman los parámetros.

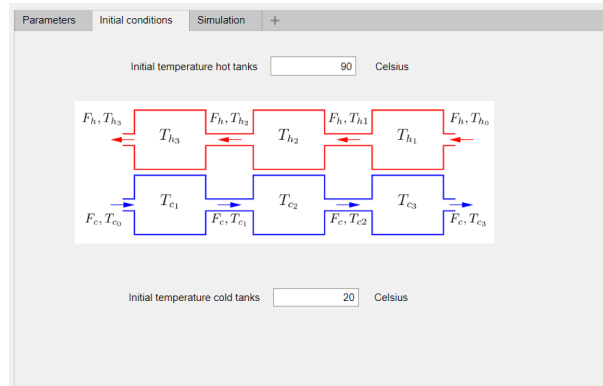


Figura 5: Panel donde se especifican las condiciones iniciales del modelo (1).

3. (10%) La interfaz de usuario debe contener un panel denominado *Simulation* (Figura 6) en el cual se especifican la duración de la simulación y las temperaturas que se van a graficar en función del tiempo. Al desplegar más de una temperatura se debe indicar el color con el cual esta se dibujo. Para ello, tome como base el código presentado en la Figura 7.
4. (30%) El sistema debe resolver el modelo (1) utilizando *Simulink* y las constantes especificadas en los paneles *Parameters* e *Initial conditions*. La Figura 8 presenta la estructura sugerida para el diagrama de bloques. Cada una de las seis ecuaciones diferenciales de (1) debe ser descrita en un subsistema independiente. Para que al llamar a *Simulink* este utilice las variables definidas al interior de una función, se debe asignar al parámetro *SrcWorkspace* el valor '*Current*', tal y como se indica en la Figura 9.

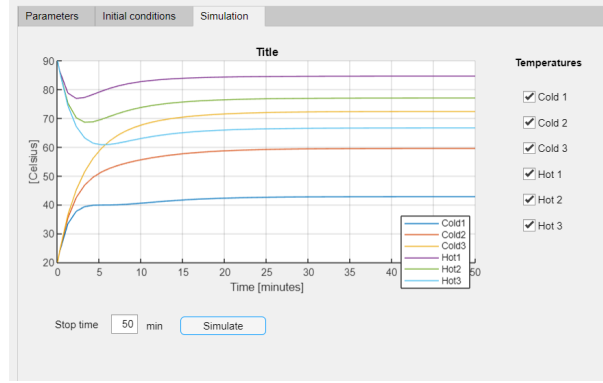


Figura 6: Panel donde se especifican el tiempo de simulación del modelo (1) y las variables a graficar.

```

1 mylegends      = {'A','B','C','D'};
2 flags          = [true; false; true; false];
3 figure;
4 hold('on');
5 if flags(1), senal1 = randn(100,1); plot(senal1); end
6 if flags(2), senal2 = randn(100,1); plot(senal2); end
7 if flags(3), senal3 = randn(100,1); plot(senal3); end
8 if flags(4), senal4 = randn(100,1); plot(senal4); end
9 legend(mylegends(flags));

```

Figura 7: Utilización de la función *legend* para un número variable de señales. La primera, segunda, tercera y cuarta señal tendrán los nombres A, B, C y D, respectivamente. El programa graficará únicamente las señales A y C, dado que solo las posiciones 1 y 3 del vector *flags* son verdaderas.

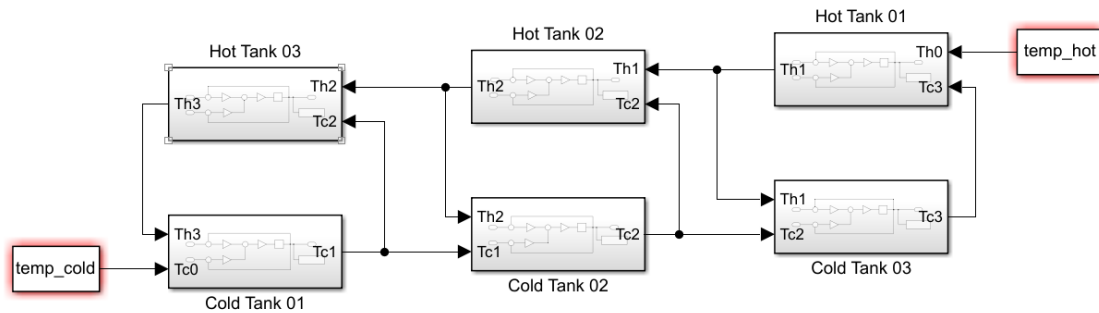


Figura 8: Diagrama de simulación del modelo (1).

El 40 % correspondiente a la sustentación se evaluará de acuerdo con la rúbrica de la Tabla 1. Durante la sustentación, los estudiantes deben demostrar que comprenden tanto el diagrama de bloques que representa al modelo (1), como el efecto de sus constantes en los resultados de simulación.

```

1 opt          = simset('SrcWorkspace','current');
2 results      = sim('test',tsim,opt);

```

Figura 9: Modificación del parámetro *SrcWorkspace* para que *Simulink* busque las variables del definidas al interior del modelo *test* dentro de la función desde la cual se invoca a la función *sim*.

Calificación	Sustentación
5	Los dos estudiantes responden satisfactoriamente y sin vacilación las preguntas realizadas por el docente.
4	Los estudiantes no responden de manera satisfactoria al menos una de las preguntas formuladas por el docente.
3	Los estudiantes no responden de manera satisfactoria al menos dos de las preguntas formuladas por el docente.
2	Uno de los estudiantes intenta responder las preguntas, pero en el otro se evidencian fuertes deficiencias en cuanto a comprensión del trabajo realizado.
1	La mayoría de las respuestas de los estudiantes son vagas o imprecisas.
0	Los estudiantes son incapaces de sustentar el trabajo que realizaron.

Cuadro 1: Rúbricas para la evaluación de la sustentación.

4. Fecha de entrega y sustentación

El código fuente de la aplicación se deberá entregar a más tardar el miércoles 28 de abril de 2021 a las 11:59 p.m., a través de la plataforma *Google Classroom*. Después de esta fecha y hora, se descontarán 2 décimas por cada hora de retardo en el envío. Las sustentaciones se realizarán el jueves 29 de abril en un horario previamente acordado con el docente.