SISTEMAS DE CONTROL I
Trabajo final integrador
Sistema de control de temperatura para un CPU

Docentes:
AGUERO CLAUDIO (Titular)
PEDRONI JUAN PABLO (Adjunto)
Alumnos:
Bernaus Julieta, Campos Mariano

1 de diciembre de 2024

Resumen

Este proyecto tiene como objetivo diseñar e implementar un sistema de control a lazo cerrado para regular la temperatura de un CPU mediante el ajuste dinámico de la velocidad de un ventilador. El sistema busca mantener la temperatura dentro de límites seguros, optimizando el desempeño y la vida útil del CPU.

Índice

1. Definición del problema

1.1. Principio de funcionamiento del sistema

El sistema propuesto es un controlador a lazo cerrado para regular la temperatura de un CPU mediante el ajuste de la velocidad de un ventilador. El principio de funcionamiento se basa en la retroalimentación: se mide constantemente la temperatura del CPU, se compara con un valor deseado (setpoint), y se ajusta dinámicamente la velocidad del ventilador para mantener la temperatura en los límites establecidos.

1.2. Variable a controlar

La variable a controlar es la temperatura del CPU,(T_{CPU} en grados Celsius, °C). El objetivo es mantenerla dentro de un rango seguro para evitar el sobrecalentamiento, con un setpoint ajustable dependiendo de la carga del sistema.

1.3. Medición de la variable de salida

Si bien algunos CPU tienen incorporados sensores de temperaturas internos,para nuestro caso utilizamos un sensor de temperatura LM35.

- Sensor:precisión de ± 0.5°C.
- Acondicionamiento de señal:El LM35 podría requerir un ADC si fuera necesario utiliza un microcontrolador(Se determina en las secciones posteriores).

1.4. Ejecución de la acción de control

La acción de control se ejecutará mediante un ventilador de corriente continua (DC) cuyo motor será controlado con señales PWM (modulación por ancho de pulso). La señal PWM ajustará las revoluciones por minuto (RPM) del ventilador, proporcionalmente a la señal de control generada por el controlador.

1.5. Variables del sistema

Variable	Unidad	Descripción
T_{CPU}	°C	Temperatura del CPU.
V_{Fan}	RPM	Velocidad del ventilador.
Señal de control (u)	% (Duty Cycle)	Señal generada por el controlador (PWM).
T_{Amb}	°C	Temperatura ambiente (perturbación).

Cuadro 1: Variables del sistema con sus unidades y descripciones.

1.6. Posibles perturbaciones

- Variaciones en la temperatura ambiente T_{Amb} El sistema debe compensar los cambios en la temperatura externa.
- Carga del CPU: A mayor carga, se genera más calor, lo que afecta directamente la variable controlada, la T_{CPU}.
- Fluctuaciones de voltaje en el suministro eléctrico: Pueden alterar el funcionamiento del ventilador(No se tiene en cuenta para el diseño).

1.7. No linealidades involucradas

El sistema de control que regula la temperatura del CPU a través del ventilador debe abordar múltiples fuentes de no linealidad:

- Ventilador: La relación entre el ciclo de trabajo PWM y la velocidad del ventilador es no lineal, especialmente a bajas RPM.
- Sensores de temperatura: Los sensores tienen respuestas no lineales que deben ser compensadas para obtener mediciones precisas.
- Transferencia de calor: El comportamiento térmico del CPU y su interacción con el sistema de enfriamiento (ventilador) es no lineal.

1.8. Niveles de señal de entrada y salida

Existen dos principales señales, donde es de especial interés conocer sus rangos dinámicos, la T_{CPU} y por otro lado la señal de control PWM.Respecto a la primer señal tenemos que la temperatura máxima que puede alcanzar un CPU depende de varios factores, como el modelo específico del procesador, el sistema de refrigeración utilizado, y las condiciones ambientales.Se debe tener en cuenta que:

- Temperatura nominal de operación: Un CPU típico a máxima carga generalmente puede alcanzar temperaturas entre 70°C y 90°C. Este rango depende del tipo de CPU (por ejemplo, Intel o AMD), el proceso de fabricación, y las condiciones de refrigeración.

- Temperatura crítica o límite superior: Los fabricantes de CPUs suelen establecer una temperatura máxima segura alrededor de los 100°C a 105°C. Si el CPU alcanza estas temperaturas, el sistema activará medidas de protección, como el thermal throttling (reducción de la velocidad de reloj) o el apagado del sistema para evitar daños.

De lo mencionado anteriormente se concluye que la T_{CPU} oscila entre 30°C y 100°C, y por tanto teniendo en cuenta el datasheet del sensor LM35, el rango dinámico de la señal resulta de $300[mV]$ hasta $1[V]$
Para la segunda señal de interés tenemos la señal PWM, esta varia segun el ciclo de trabajo desde $0\,\%$ hasta el $100\,\%$, esto se traduce en un rango dinámico de $0[V]$ (300[RPM]) hasta $5[V]$(5000[RPM]).

2. Análisis de la planta

En esta sección se plantea un diagrama de bloques para modelar el sistema en su totalidad, se identifican sus partes principales(controlador, planta, sensor), las variables de entrada, salida y perturbaciones, ademas se detalla como interactúan estas entre si.

2.1. Diagrama de bloques del sistema

El diagrama de bloques de este sistema incluiría los siguientes elementos:

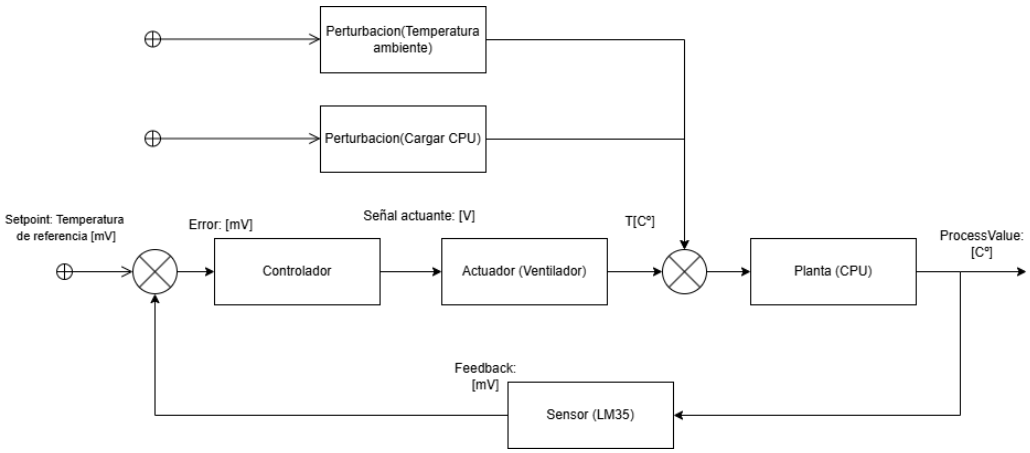

Figura 1: Diagrama de bloques del sistema

- Entrada (señal de referencia):La temperatura deseada o setpoint (se puede suponer una temperatura objetivo o de referencia a mantener para el CPU, T_{ref}.
- Controlador: El controlador compara la temperatura medida con la temperatura de referencia y genera una señal de control en forma de PWM para el ventilador.
- Actuador: El ventilador actúa como el actuador. Recibe la señal PWM del controlador y ajusta su velocidad en función de la carga térmica del CPU.
- Perturbaciones:La temperatura ambiente y la carga del CPU son las principales perturbaciones que afectan la temperatura del CPU. Estas son fuentes de interferencia en el sistema.
- Sensor:El sensor mide la temperatura real del CPU, T_{CPU} y esta señal es retroalimentada al controlador.
- Planta: El CPU es el elemento a controlar, las variables de entrada son las perturbaciones que afectan su temperatura, y la refrigeración qu aporta el ventilador, la salida es el process value T_{CPU}

2.2. Modelado matemático de los componentes

2.2.1. Sensor de temperatura

En el caso de la función de transferencia del sensor de temperatura LM35,esta se puede obtener de forma teórica, ya que en el datasheet se encuentra la expresión $V_{OUT} = F(T_{Amb})$, para la aplicación típica (FIGURE 1. Basic Centigrade Temperature Sensor), la expresión resulta:

$$V_{OUT}[V] = 0[V] + 0{,}01[V].T_{Amb}[C] \tag{1}$$

La función de transferencia de interés, se obtienen pasando la ecuación al dominio operacional y despejando la expresión $Voltaje/Temperatura$, esta resulta:

$$FdT_{Sensor} = \frac{V(s)}{T(s)} = 0{,}01 \tag{2}$$

2.2.2. Ventilador

Para obtener la función de transferencia del ventilador se opto por hacerlo de forma experimental, ya que la relación entrada-salida de la señal PWM y temperatura, no se encontraba disponible en ninguna bibliografía.El procedimiento es el siguiente:

- Armar el sistema de refrigeración con el ventilador y el habitáculo donde se va a encontrar el CPU.
- Hacer variar la señal de entrada PWM del ventilador y medir la temperatura resultante en el habitáculo.
- Con los datos obtenidos, mediante un script en Matlab, obtener la función de transferencia

Nota: En nuestro caso los datos son simulados con un script de python.

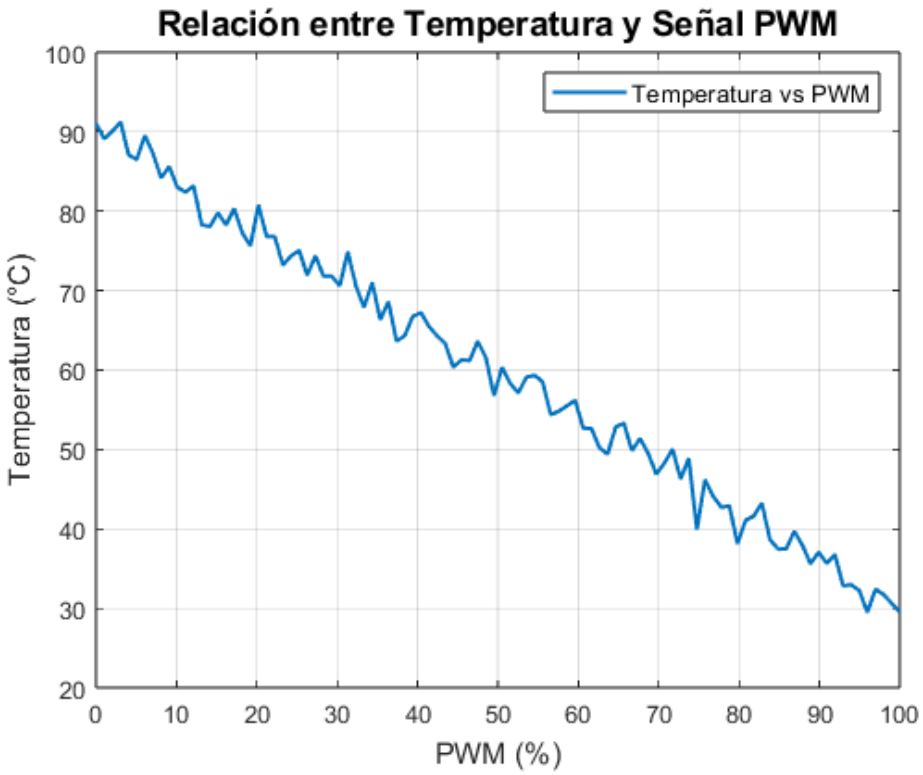

Figura 2: Datos recopilados

Con los datos obtenidos se realiza el siguiente análisis para obtener la función de transferencia:

```
    %Calculo de funcion de transferencias para el
vantilador
    %Importamos los datos obtenidos de las mediciones
    opts = delimitedTextImportOptions("NumVariables", 2);
    opts.DataLines = [2, Inf];
    opts.Delimiter = ",";
    opts.VariableNames = ["PWM", "TemperatureC"];
    opts.VariableTypes = ["double", "double"];
    opts.ExtraColumnsRule = "ignore";
    opts.EmptyLineRule = "read";
    data = readtable("D:\GD\Sistemas de control I\Trabajo-
Integrador-SCI\Datos\pwm_temperature_data.csv", opts)

    % Leer el archivo CSV como tabla
```

```
13      data = readtable("D:\GD\Sistemas de control I\Trabajo-
    Integrador-SCI\Datos\pwm_temperature_data.csv",opts);
14
15      % Separar las columnas en variables
16      PWM = data.("PWM"); % Columna PWM
17      Temperature = data.("TemperatureC"); % Columna
    Temperatura
18
19      % Crear el grafico
20      figure;
21      plot(PWM, Temperature, 'LineWidth', 1.5, 'MarkerSize',
    6, 'DisplayName', 'Temperatura vs PWM');
22      grid on;
23      xlabel('PWM (%)', 'FontSize', 12);
24      ylabel('Temperatura (C)', 'FontSize', 12);
25      title('Relacion entre Temperatura y Senal PWM', '
    FontSize', 14);
26      legend('show', 'Location', 'northeast', 'FontSize', 10)
    ;
27
28
29      % Crear entrada y salida como senales en tiempo (
    asumiendo un muestreo uniforme)
30      t = linspace(0, length(PWM)-1, length(PWM)); % Tiempo
    ficticio
31      u = PWM; %Entrada (PWM )
32      y = Temperature; % Salida (Temperatura)
33
34      % Ajustar un modelo de primer o segundo orden (dominio
    Laplace)
35      data_id = iddata(y, u, 1); % Crear datos de
    identificacion con un muestreo ficticio de 1s
36      model = tfest(data_id, 1, 0); % Ajustar un modelo de
    primer orden sin ceros
37
38      % Mostrar la funcion de transferencia estimada
39      disp('Funcion de transferencia estimada:');
40      disp(model);
41
42      % Graficar comparacion entre el modelo ajustado y los
    datos
43      figure;
44      compare(data_id, model);
45      title('Comparacion entre datos reales y modelo ajustado
    ', 'FontSize', 14);
```

```
46      xlabel('Tiempo (s)', 'FontSize', 12);
47      ylabel('Temperatura (C)', 'FontSize', 12);
48      legend('Datos reales', 'Modelo ajustado', 'Location', '
    best', 'FontSize', 10);
49      grid on;
50
```

La función de transferencia obtenida resulta:

$$FdT_{Ventilador} = \frac{-0{,}0038}{s + 0{,}007} \tag{3}$$

Figura 3:

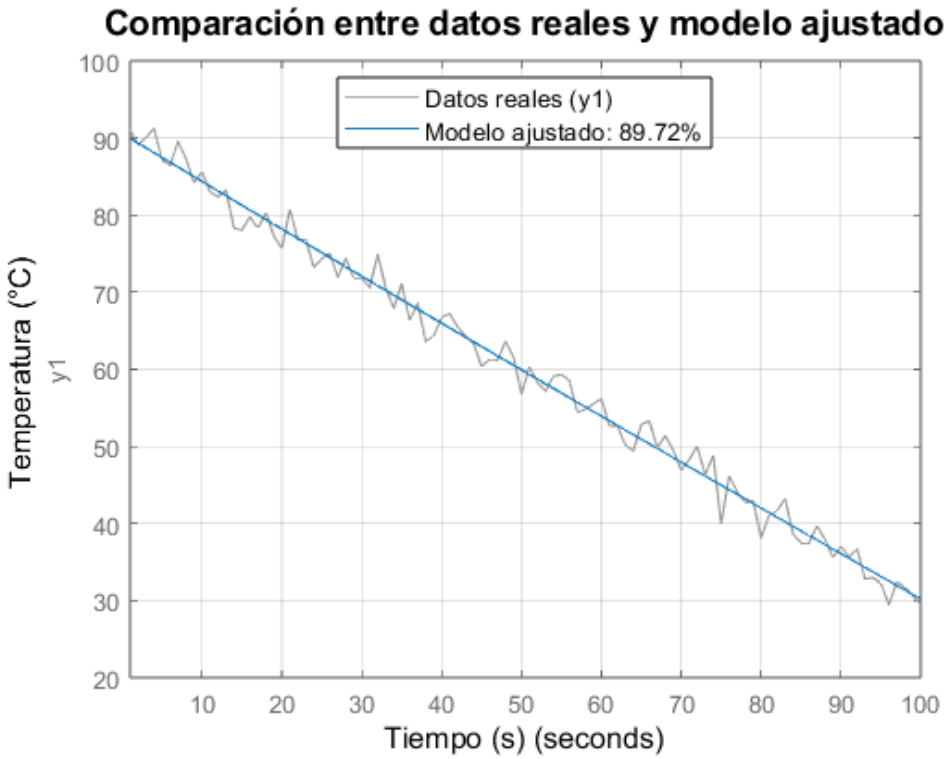

Figura 4: Ajuste del modelo a partir de los datos