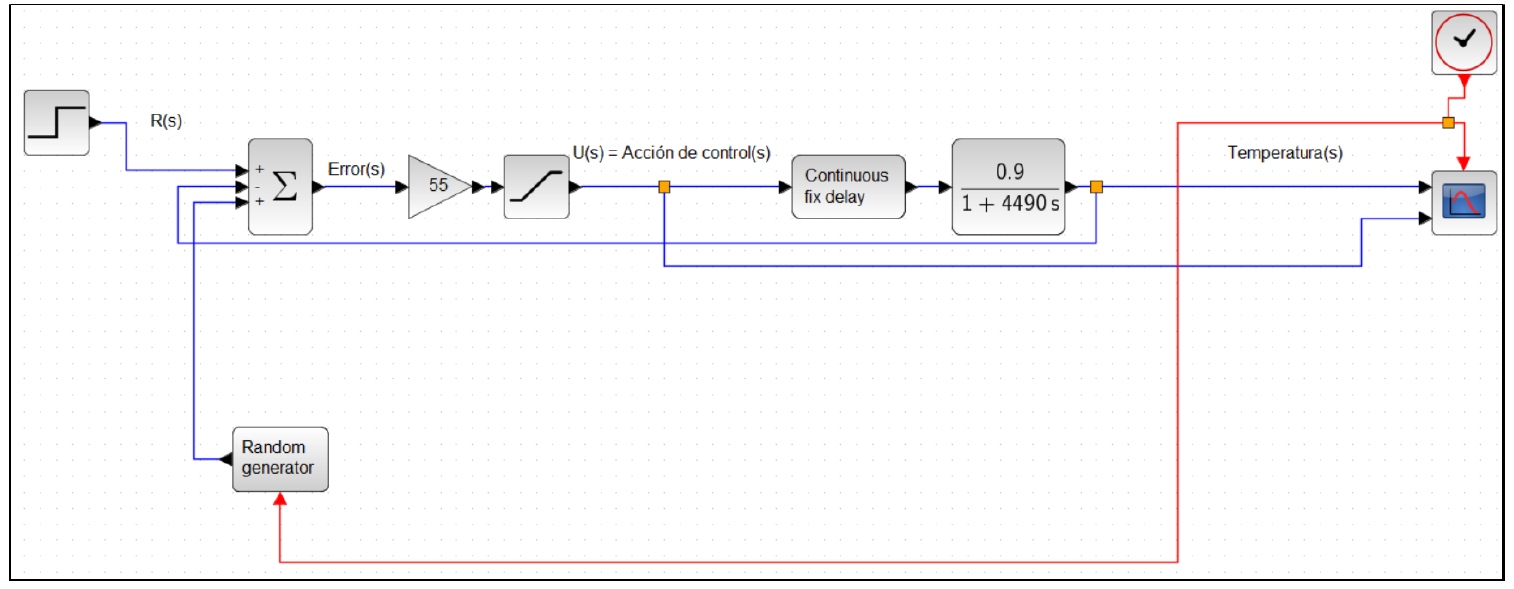
* Objetivo
* Breve descripción del modelo: ¿orden? ¿retardo? ¿alinealidades? ¿ruido?...
* Elección de los parámetros Kp y/o Kp+Ti
* Mejoras en software y hardware (si las hubo)
* Práctica en sí. Comparación con simulación. ¿Las mejoras sirvieron?
* Problemas presentados.
* Conclusiones

Objetivo

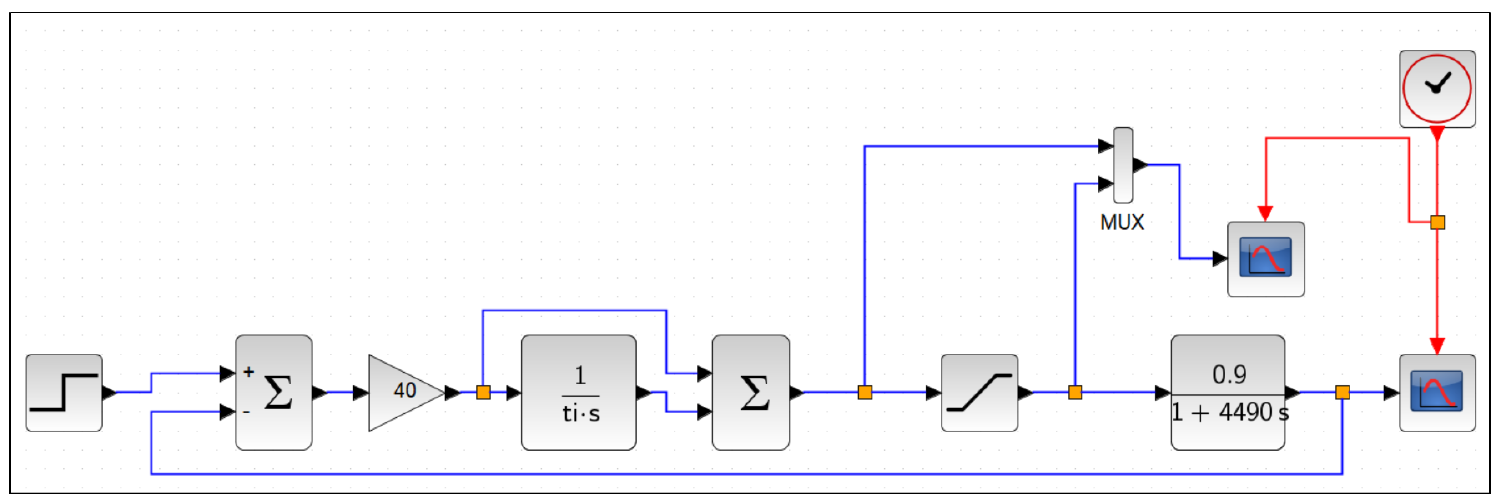
Nuestro objetivo es establecer la temperatura del agua a 85°C. En el controlador proporcional nuestro requerimiento de diseño es utilizar un Kp que otorgue un error a estado estacionario de 2%. Utilizando el teorema de valor final hayamos Kp = 55 aproximadamente para que la temperatura del agua en estado estacionario sea de 83°C que se encuentra dentro del rango de 2%.

Breve descripción del modelo: ¿orden? ¿retardo? ¿alinealidades? ¿ruido?...

El modelo de control proporcional se agrego en xcos un bloque de retardo, también un bloque de ruido, y como alinealidad agregamos un bloque de saturación que limita la potencia de la pava a 2200W.



En el modelo de control pi, se agrego alinealidad de saturación.



Elección de los parámetros Kp y/o Kp+Ti

Para el controlador proporcional-integral, se eligió ganancia K = 40 para no tener sobrevalor

en la señal de salida. Al elegir un K, notamos que a medida que aumentaba el mismo el

sistema se volvía más rápido, pero generaba sobrevalor. Por otro lado, si elegimos K muy

pequeño el sistema se vuelve muy lento. Por lo tanto, se definió K de 40 debido a que

notamos que es el más estable. Luego de hacer varias pruebas se eligió un tiempo de integración de 50 minutos.

Con estos parámetros logramos un sistema más estable y eficaz, tratando de que nuestro

sistema no sea muy lento y no genere un sobrevalor muy alto.

Otras opciones al elegir Kp y Ti.

| Caso | Ti [min] | Kp [W/°C] | T crecimiento 10% a 90% | T establecido 2% |
| --- | --- | --- | --- | --- |
| 1 | 100 min | 2 | 124,56 min | 288,87 min |
| 2 | 45 min | 2 | 68 min | 203 min |
| 3 | 100 min | 0.32 | 816,48 min | 1380,47 min |
| 4 | 45min | 0.32 | 303,68 min | 474,70 min |

Problemas presentados.

Problema 1

Tuvimos el incoveniente en que el circuito del SSR (rele de estado solido) esta diseñado para apagarse cuando la señal senoidal se encuentra a los 0W. Por lo tanto cuando enviábamos la orden de apagar, teníamos que esperar a que la señal senoidal llegue a 0W para apagarse el rele efectivamente

Problema 2

El controlador del sensor de temperatura para medir fue programado con un delay bloqueante de 0.6s Entonces la potencia aplicada de la Pava no será de 2200W sino menor.

Mejoras en software y hardware (si las hubo)

Mejora 1 - Software

Tuvimos el incoveniente en que el circuito del SSR (rele de estado solido) esta diseñado para apagarse cuando la señal senoidal se encuentra a los 0W. Por lo tanto cuando enviábamos la orden de apagar, teníamos que esperar a que la señal senoidal llegue a 0W para apagarse el rele efectivamente. Para solucionarlo, debemos hacer que la frecuencia del pulso sea mayor que al periodo de la senoidal. Entonces se programó PWM (Modulo de ancho de pulso ) de 5s en Arduino. De esta manera logramos una mayor precisión al desear apagar el rele en el momento que deseamos.

Mejora 2 - Software

En vez de limitar la potencia a 2200W , se limito por código a 1936W. Esto es debido a que el controlador del sensor de temperatura para medir fue programado con un delay bloqueante de 0.6s. Entonces la pava no estará encendida 5s, sino , 4,4s. Por lo tanto se calculo la potencia correspondiente para el periodo de 4.4s.

Práctica en sí. Comparación con simulación. ¿Las mejoras sirvieron?

En la simulación podemos notar que utilizando el controlador proporcional el agua llega a la

temperatura de los 83°C en 527s. En cambio, en el controlador proporcional-integral tarda

423s en llegar a dicha temperatura.

Por lo tanto, el control proporcional es más lento al llegar la temperatura del agua a 83°C en

comparación al controlador proporcional-integral.

Además, el controlador proporcional-integral logra que la temperatura del agua se mantenga

a los 85°C desde los 590s.

En la experiencia, identificamos que el agua tarda 285s en llegar a los 83°C para el

controlador proporcional. Tarda menos tiempo que en la simulación, debido a que en la

simulación agregamos bloque de retardo que afecta en la velocidad del sistema.

En el controlador proporcional-integral se tarda 310s en llegar a los 83°C. Esto quiere decir

que el controlador proporcional actuó más rápido que el proporcional-integral.

Sin embargo, con el controlador proporcional-integral logramos llegar a la temperatura

objetivo de 85°C en 830s.

Conclusiones

Si deseamos que el agua esté a la temperatura exacta que deseamos. Entonces es

preferible usar el controlador proporcional-integral, pero debemos tener en cuenta de la

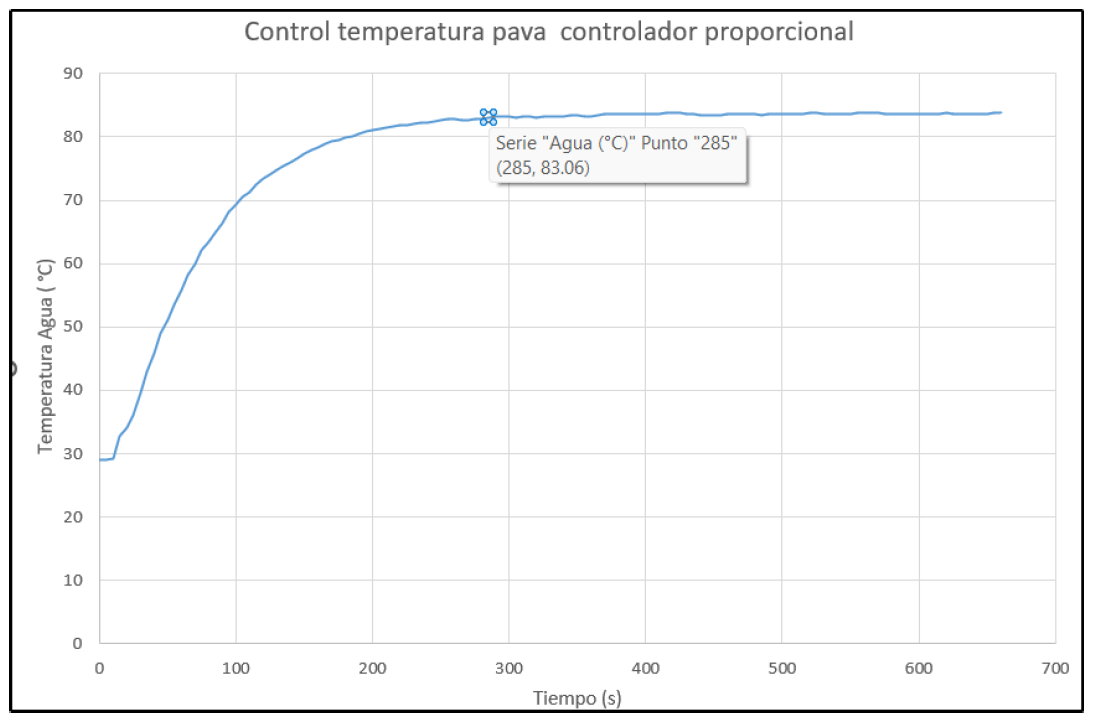
experiencia que debemos esperar más tiempo para lograr dicha temperatura.

Si preferimos rapidez en vez de precisión, entonces elegimos el controlador proporcional ya

que , si bien no llega a los 85°C, podemos utilizar el agua en una temperatura menor a la

que deseamos, en este caso 83°C.

Proporcional (Temperatura agua):



PI (Temperatura agua con perturbacion):

