

CONTROLE DE TEMPERATURA PID
Sistema de Controle

Luan Santos Gonçalves
Marco Aurélio Monteiro Lima
Pâmela Evelyn Carvalho
Raianny Magalhães Silva

Bambuí-MG
Junho/2019

1 INTRODUÇÃO

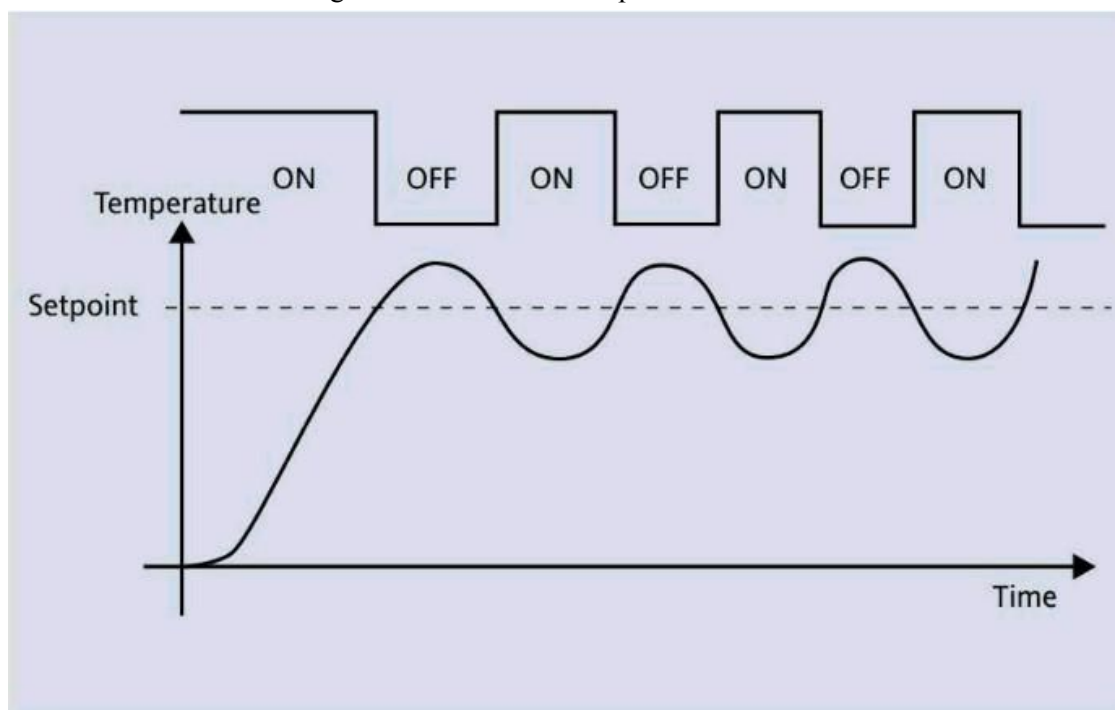
O controle de temperatura é definido como o acompanhamento contínuo da variação de temperatura, que pode ser realizado em um ambiente, produto, produções químicas ou em processos técnicos variados. Tal controle pode ser feito de forma manual, observando os valores mostrados em um termômetro durante um espaço de tempo, ou de forma automática, quando são utilizados sensores ou sistemas computadorizados. O monitoramento da temperatura é extremamente indicado em operações industriais, principalmente em áreas de saúde e alimentação (SILVEIRA, 2018).

De acordo com Silveira (2018), existem três tipos de controle de temperatura, que são:

- ON/OFF: é o dispositivo mais simples, no qual a saída é alternada somente quando a temperatura ultrapassar o ponto de ajuste;
- Proporcional: diminui a potência média fornecida ao aquecedor quando a temperatura se aproxima do ponto de ajuste, mantendo uma temperatura estável;
- PID: é o controle proporcional com integral e derivativo, que permite que o sistema opere dentro de uma banda proporcional.

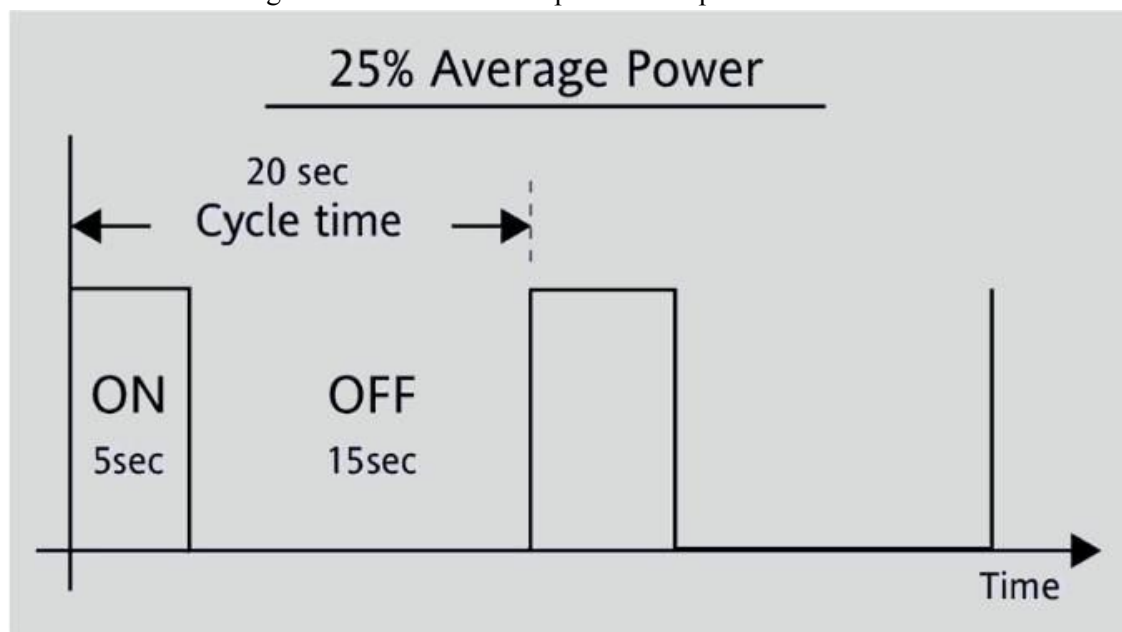
Os gráficos referentes ao controle de temperatura ON/OFF, Proporcional e PID, podem ser observados nas Figuras 1, 2 e 3, respectivamente.

Figura 1: Controle de Temperatura ON/OFF



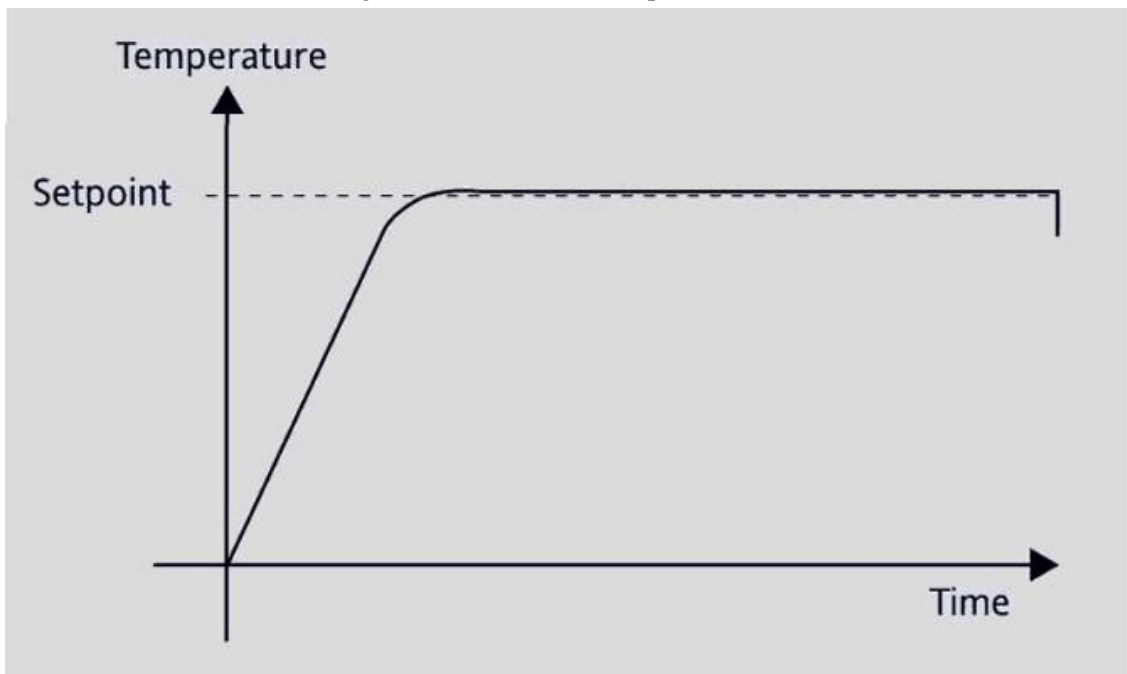
Fonte: Silveira (2018).

Figura 2: Controle de Temperatura Proporcional a 25%



Fonte: Silveira (2018).

Figura 3: Controle de Temperatura PID



Fonte: Silveira (2018).

O objetivo deste trabalho é desenvolver um controlador de temperatura PID, por isso, se faz necessário obter um maior conhecimento sobre o seu funcionamento.

O controle PID é definido como proporcional-integral-derivativo. O proporcional permite que o controle seja ajustado de acordo com a situação atual. Enquanto a ação integral faz a correção proporcional do erro vezes o tempo. O derivativo determina a correção proporcional à taxa de variação do erro. Combinados, os três utilizam dados atuais, dados passados e a taxa na qual os dados estão variando para definir um algoritmo específico de controle de temperatura, atualizando o erro de temperatura entre a variável de processo e o ponto de ajuste, fazendo com que a temperatura esteja sempre constante (CHRISTIAN, 2017).

2 DESENVOLVIMENTO

2.1 Materiais

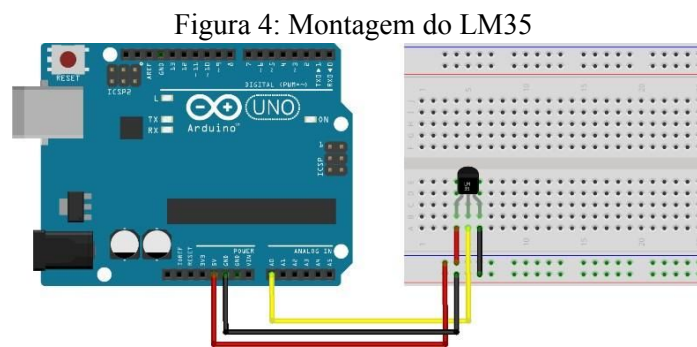
Para o desenvolvimento deste trabalho foram utilizados os seguintes materiais:

- Protoboard;
- Ferro de solda ;
- Sensor de temperatura LM35;
- Arduino Uno;
- Cooler;

- Transistor MOSFET irfb3306;
- Jumpers de conexão;
- *Software* Scilab versão 6.0.1;
- Ambiente de desenvolvimento do Arduino.
- Cola quente;
- Fita isolante;
- Tábua;

2.2 Metodologia

Inicialmente foi realizada a montagem do sensor de temperatura separadamente para testar o funcionamento do mesmo, para isso, o sensor de temperatura foi ligado nas entradas Ground e Vcc (5 volts) do Arduino. A Figura 4 exibe a montagem realizada. A programação para a leitura correta de temperatura foi realizada na IDE do Arduino.



Fonte: Os Autores (2019).

Para a realização do controle do *cooler* é necessário a utilização de um MOSFET, o componente recebe um sinal analógico da plataforma Arduino. Por meio do sinal recebido, a potência do *cooler* é controlada.

A variação de temperatura é de suma importância para que o PID possa ser devidamente utilizado. Para se alcançar temperaturas mais elevadas, um ferro de solda foi posicionado próximo do sensor de temperatura, dessa maneira, quando o ferro de solda for ligado, uma variação de temperatura vai ocorrer.

Após a montagem dos componentes realizada pode-se iniciar o desenvolvimento do PID. O primeiro passo foi a codificação do algoritmo para realizar o controle PID. Em seguida ocorreu a energização do ferro de solda, sendo que a temperatura foi monitorada até que o sistema se estabilizasse em um valor máximo. Com a temperatura estabilizada, o *cooler* foi ligado em potência máxima até que a temperatura se estabilizasse novamente.

Os dados das temperaturas capturadas foram salvas em um arquivo de texto para

posterior utilização no *software* Scilab. No Scilab, os dados foram lidos e um gráfico contendo a variação de temperatura por tempo pôde ser plotado.

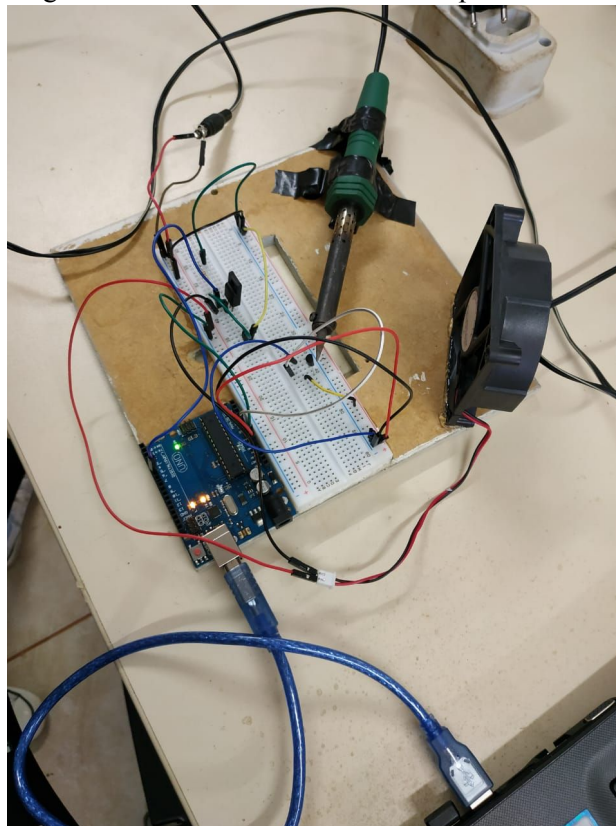
A função XCos do Scilab permite a simulação de um PID, para isso é necessário encontrar os valores da função $G(s)$ do sistema em questão. Após encontrar a $G(s)$, os valores proporcional (k_p), integrativo (k_i) e derivativo (k_d) foram calculados.

Voltando para a plataforma Arduino, o controle PID deve ser realizado com as variáveis encontradas, alterando os valores das mesmas via código. Após o acionamento do sistema, foram capturados os dados de temperatura com o PID funcionando. O resultado de todos os processos descritos pode ser visualizado na seção de resultados.

2.3 Resultados

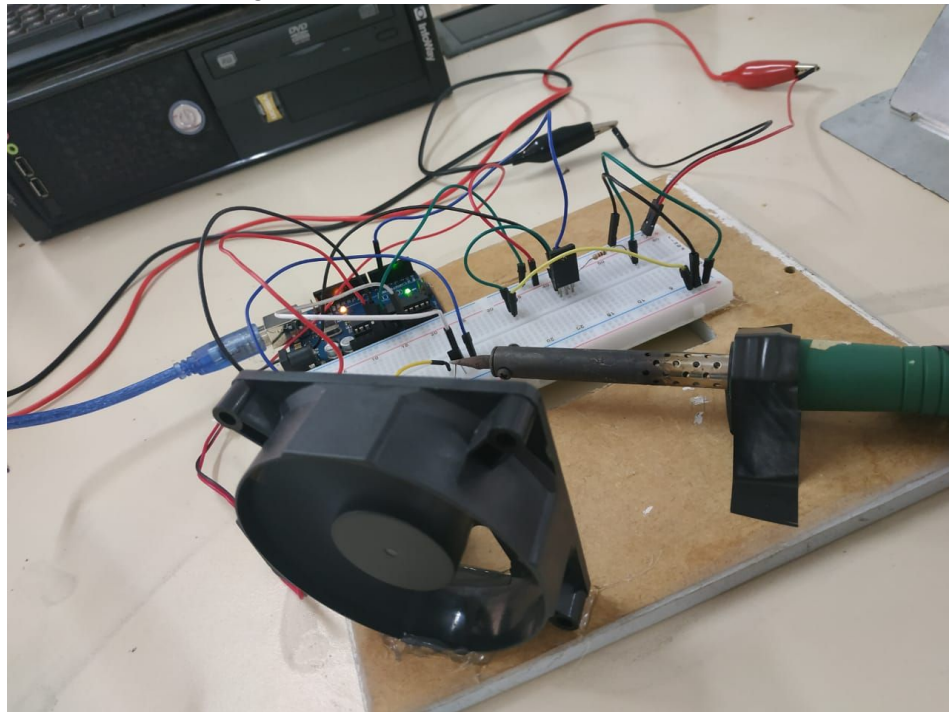
Inicialmente, o sensor LM35 foi testado e o seu correto funcionamento foi comprovado. O passo seguinte foi a montagem completa do circuito, sendo acoplado o MOSFET, o *cooler* e o ferro de solda ao sistema. O projeto montado pode ser observado nas Figuras 5 e 6.

Figura 5: Circuito montado - visão superior



Fonte: Os Autores (2019).

Figura 6: Circuito montado - visão lateral



Fonte: Os Autores (2019).

Após a montagem do circuito, o ferro de solda foi ligado e coletou-se a leitura feita pelo sensor de temperatura até que a mesma estivesse estabilizada, com objetivo de obter o valor máximo que o sistema poderia chegar em malha aberta. Quando a temperatura chegou a aproximadamente 71,8°C não houve nenhuma outra mudança significativa na leitura.

Determinado o maior valor, ligou-se o *cooler* para que ocorresse o resfriamento do sistema até, novamente, encontrar um ponto de estabilidade. O menor valor obtido pela leitura do sensor foi de 26,4°C.

Com os valores em malha aberta obtidos, foi calculado o valor de k , que é dado pela diferença entre o valor máximo e mínimo da temperatura. Então, fez-se a substituição do mesmo na fórmula para encontrar τ (tau), sendo que $\tau = k \cdot 0,63$, obtendo o valor de 24,6.

O *software* SciLab foi utilizado para esboçar o gráfico que relaciona os valores de temperatura obtidos com o intervalo de tempo observado. O gráfico é exibido na Figura 7.

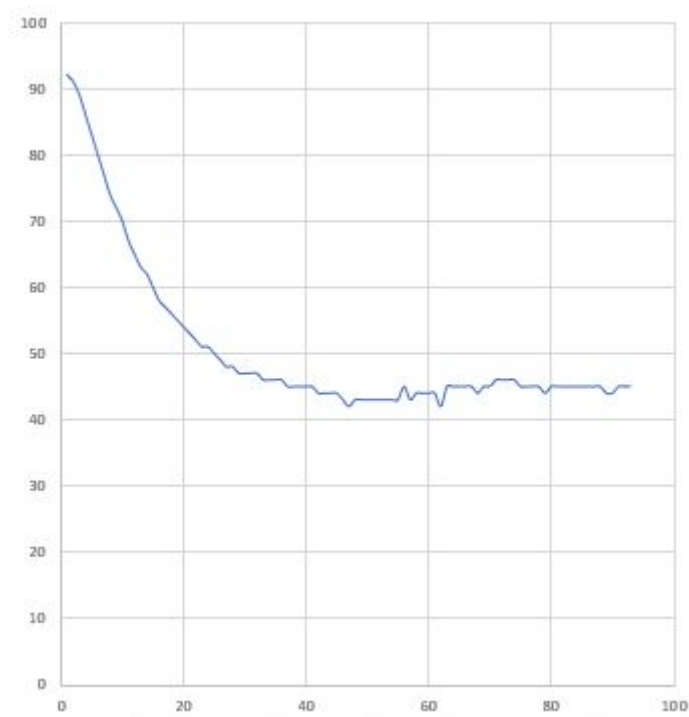


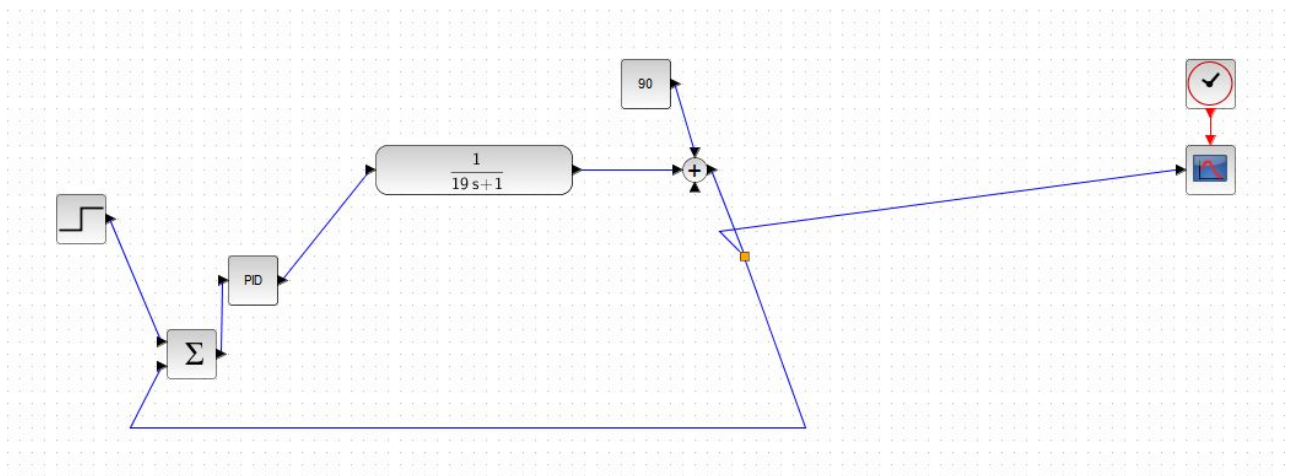
Figura 7: Gráfico de temperatura por tempo
Fonte: Os Autores (2019).

Analisando a curva do gráfico e usando o valor encontrado de τ , montou-se então a função $G(s)$:

$$G(s) = \frac{-1}{19s + 1}$$

Determinado a $G(s)$, foi feita uma simulação no SciLab, usando o XCos para implementação de um PID por meio de um diagrama de blocos. A Figura 8 mostra o diagrama de blocos em questão.

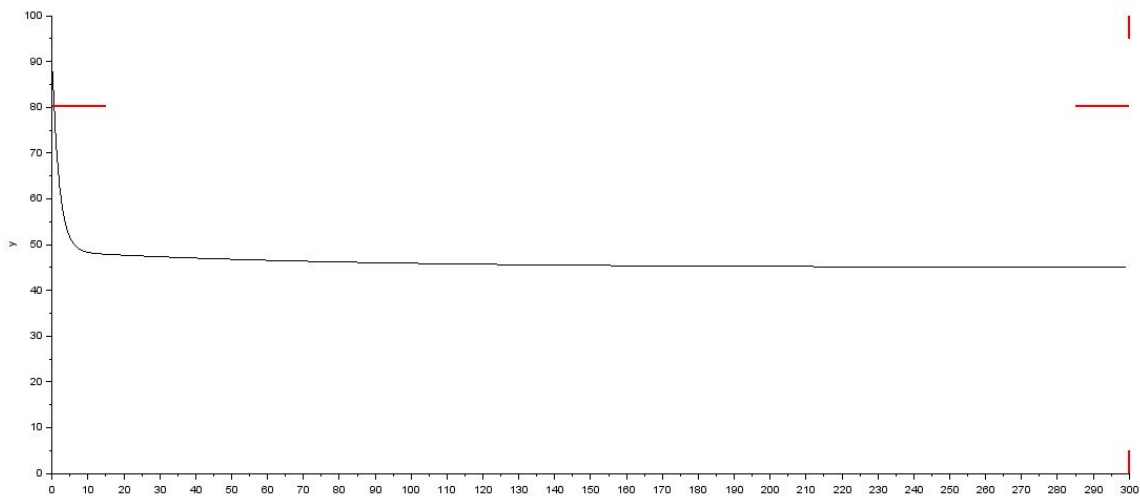
Figura 8: Diagrama de bolcos



Fonte: Os Autores (2019).

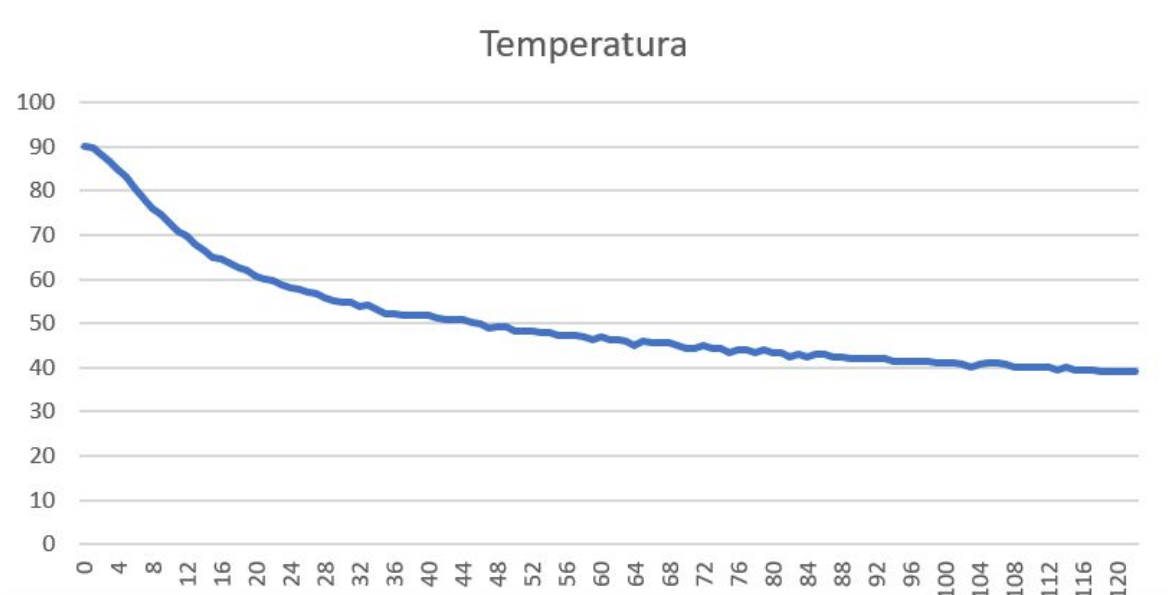
É possível visualizar na Figura 8 a função $G(s)$ no sistema, no qual são montados dois diagramas, o primeiro com um sistema fechado e o segundo um sistema aberto. As Figuras 9 e 10 apresentam os gráficos gerados pela simulação realizada.

Figura 9: Gráfico gerado pela execução do PID simulado



Fonte: Os Autores (2019).

Figura 10: Gráfico gerado pelo excel



Fonte: Os Autores (2019).

Então, foi feito o cálculo para encontrar os valores de k_p , k_i e k_d , considerando o tempo de acomodação do sistema de 10s e foram obtidos os valores de $k_p = 9,4$, $k_i = 4$ e $k_d = -23,6$. Estes valores foram substituídos na plataforma Arduino, por meio de código, com o intuito de obter o ajuste automático da temperatura com um valor especificado. A Figura 11 apresenta o gráfico obtido com o controle PID realizado com o Arduino.

3 CONCLUSÃO

Ao término deste trabalho pode-se concluir que o controle PID é um algoritmo utilizado em controle de processos, sendo aplicado em sistemas de malha fechada, isto é, controle com realimentação, que objetiva minimizar os erros e acertar a saída do sistema de acordo com um valor esperado.

Também é importante destacar que o projeto mecânico deve estar bem especificado para que o controle PID funcione perfeitamente. Caso isso não aconteça, o sistema não responderá adequadamente aos comandos dados.

REFERÊNCIAS

CHRISTIAN, Y. Desmistificando: **Controle PID (Proporcional Integral Derivativo)**. 2017. Disponível em: <<http://engenheirocaicara.com/desmistificando-controle-pid-proporcional-integral-derivativo/>>. Acesso em: 25 jun. 2019.

SILVEIRA, C. **Como Funciona o Controle de Temperatura?**. 2018. Disponível em: <https://www.citisystems.com.br/controle-de-temperatura/#disqus_thread>. Acesso em: 25 jun. 2019.