

UNIVERSIDADE FEDERAL DE SANTA MARIA
DEL C 1048 – PROJETO DE SISTEMAS EMBARCADOS

Lorenzo Moreira Donatti

Helena Fernanda Kray

RELATÓRIO PROJETO FINAL

SANTA MARIA, 26 DE AGOSTO DE 2021

INTRODUÇÃO

O projeto desenvolvido refere-se a uma aplicação genérica de um sistema embarcado que controla a temperatura de um ambiente através de um controlador PID em programação concorrente. Para fins práticos, foi utilizado o simulador PROTEUS para simular um circuito com Arduino. Para o controle de concorrência foi utilizado o sistema operacional FreeRTOS e suas funções referentes a mutexes e semáforos.

É importante frisar que devido a simulação ocorrer em um ambiente virtual, limitações ocorreram, como a falta de controle da temperatura na saída do motor utilizado. Porém, mesmo com as limitações, o sistema embarcado conseguiu cumprir com sua função.

CONTROLADOR PID

Para o melhor entendimento do sistema embarcado, é necessária uma análise de como funciona o controle PID. Essa sigla vem de Proporcional-integral-derivativo, nome que se refere aos processos utilizados pelo mesmo. O controlador passa por essas três etapas de controle, onde:

A ação proporcional produz uma saída proporcional ao erro da aplicação, sendo capaz de reduzir oscilações durante a execução do controlador.

A ação integral produz uma saída proporcional a magnitude e tempo de erro, ou seja, por menor que seja o erro, a integral assume valores acumulados com o tempo, servindo para acelerar a chegada no valor de referência, além de corrigir o erro.

A ação derivativa produz uma saída proporcional à velocidade em que o erro varia, suavizando o valor de saída caso o erro aumente rapidamente, melhora a estabilidade do sistema.

O controle PID se dá pela soma de todas as ações, tendo a seguinte fórmula:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

APLICAÇÃO

No sistema embarcado, o controle PID foi implementado para o controle da temperatura através de um motor, onde no mundo real significaria uma hélice de um fan resfriando o ambiente.

É necessário definir um valor de referência para a temperatura (*chamado de Setpoint no algoritmo*), onde nessa temperatura o motor irá rotacionar em uma velocidade base, através de um sinal PWM constante.

Porém, se a temperatura detectada pelo sensor for diferente da referência, o algoritmo de controle entra em ação, calculando o PID frequentemente, aumentando ou diminuindo inteligentemente o valor do PWM, a depender do valor de erro, que é calculado através da subtração do valor marcado pelo valor de referência. Sinal PWM que é responsável pela velocidade de rotação do motor.

Infelizmente pelas limitações do simulador não é possível calcular a diferença de temperatura após o aumento ou diminuição da velocidade do motor, mas no mundo real, ao ter sua velocidade aumentada, atuaria diminuindo a temperatura, fazendo o valor do erro diminuir e consequentemente diminuindo sua velocidade automaticamente, devido ao controle PID. O inverso também se aplica quando sua velocidade é diminuída, devido a temperatura medida ser menor que a referência, onde no mundo real a temperatura aumentaria, fazendo o motor voltar a rotacionar em sua velocidade padrão.

É possível notar que o comportamento do sistema se assemelha a um ar condicionado “inteligente”, onde o mesmo seria adaptado ao valor de referência escolhido sempre que houvesse alguma variação.

SIMULAÇÃO

Para a simulação, como dito anteriormente, foi utilizado o software PROTEUS, inicialmente foi proposto o uso do TINKERCAD, porém não foi possível aportar bibliotecas externas para o Arduino, impossibilitando o uso do FreeRTOS. No PROTEUS foi necessário o uso da biblioteca Arduino.lib, (*obtida no site <https://www.theengineeringprojects.com>*).

Para o circuito foi-se utilizado um sensor de temperatura TMP36, ligado a entrada analógica A0 do Arduino UNO, além de um motor genérico DC do simulador para exemplificar o aumento/diminuição do PWM, ligado na saída 10 (PWM) do Arduino.

Além do circuito, o principal é o algoritmo, porém este não será mostrado neste relatório, mas sim em um PDF separado gerado pelo software *Doxygen*.

O comportamento simulado agiu dentro do esperado, sendo possível observar a velocidade do motor variando com o cálculo do PWM necessário de acordo com o controle PID. Sendo esta velocidade aumentada proporcionalmente ao erro, no caso de erro positivo (temperatura medida maior que temperatura de referência), ou tendo a velocidade diminuída proporcionalmente ao erro caso a temperatura medida seja menor que a temperatura de referência (erro negativo).

Figura 1 – Circuito simulado no PROTEUS

