

CONTROLE PID DE UMA PLANTA TÉRMICA EMULADA COM PARAMETRIZAÇÃO VIA IOT

Guilherme Braga de Paula¹, Fabrizio Leonardi²

¹Elétrica, Centro Universitário FEI

²Mecânica, Centro Universitário FEI

guilhermebragadepaula7@gmail.com¹ – fabrizio@fei.edu.br²

Resumo: O projeto de controladores de malha fechada depende das características da planta e do desempenho esperado e, para implementá-lo, existem diversas opções de hardware e software. Esta pesquisa propõe analisar e comparar os controladores *PID* e *SMC* para o controle de temperatura de um fluido em repouso e com suas parametrizações sendo feitas de modo remoto por meio de um sistema baseado em Internet das Coisas (*IoT*). Testes realizados mostram que a solução é viável e que poderá trazer benefícios não planejados inicialmente, mas igualmente relevantes.

1. Introdução

O controle de temperatura de um fluido em repouso depende por exemplo, do tipo do fluido, do nível e das condições de perda de energia para o meio, requerendo reprojetado para cada condição. O reprojetado pode se tornar inviável dependendo do acesso ao controlador embarcado para a troca de seus parâmetros.

Este projeto propõe investigar comparativamente os controladores *PID* [1] e *SMC* [2], analisando suas técnicas de sintonia e seus desempenhos no controle de temperatura de um fluido em repouso com parametrização remota via *IoT*. Devido ao período de quarentena, a planta real foi substituída por uma emulação digital da planta implementada no microcontrolador *Arduino UNO* em um esquema *hardware-in-the-loop (HIL)*, técnica onde o controlador é conectado a um dispositivo (*hardware*) de teste que simula uma “realidade” desejada [3], como o *CarSim* da empresa *Mechanical Simulation Corporation*, um *software* que simula o comportamento de veículos a partir dos comandos do motorista. No projeto, a “realidade” desejada é a planta, representada agora por meio de uma função de transferência de segunda ordem, conforme (2), com os ganhos K_g , K_1 e K_2 variáveis. Esse tipo de planta foi escolhido pois aumenta a complexidade do projeto do controlador, uma vez que no controle de temperatura real, a planta se daria por uma função de primeira ordem, conforme (1).

$$G(s) = \frac{K_g}{\tau s + 1} \quad (1) \quad G(s) = \frac{K_g}{s^2 + K_1 s + K_2} \quad (2)$$

O controlador *PID* (Proporcional, Integral, Derivativo) é um compensador com ação proporcional, integral e derivativa. Esse tipo de controlador costuma conferir bom desempenho com a junção da estabilidade da ação derivativa e com a eliminação do erro estacionário (*offset*) que a ação integral proporciona [1] [4].

Internet das Coisas (*IoT*) é a definição dada para a tecnologia que conecta pessoas ou máquinas a uma rede sem fio e proporciona comunicação direta entre ambas, ou seja, é uma rede de “coisas” físicas interligadas via rede e que conseguem coletar e trocar dados [5].

A parametrização dos controladores será feita remotamente utilizando o conceito *IoT* e, para tal, será implementada no microcontrolador *ESP32* uma comunicação sem fio utilizando o protocolo *MQTT* (*Message Queuing Telemetry Transport*), um protocolo de comunicação que atua a partir de um cliente e um servidor, onde o primeiro envia ou recebe dados e o segundo organiza e administra os dados recebidos ou a serem enviados[6].

2. Metodologia

O projeto se inicia a partir da revisão bibliográfica e do planejamento das etapas do procedimento experimental. Em seguida, foram feitos testes das funções do *ESP32* para que houvesse maior familiarização com o dispositivo e observar suas peculiaridades.

Testes iniciais em malha aberta foram emulados para validação de rotinas de atuação na variável manipulada utilizando uma lógica *PWM*. Também foram feitos testes de conectividade *MQTT* do *ESP32* com o *broker*. Os resultados observados sugerem que será possível atuar na planta e fazer a supervisão pelo computador.

Em seguida, diversos testes foram feitos utilizando o *software* *MATLAB*, assim, apenas após obter resultados satisfatórios na simulação pelo *software*, começaram os testes utilizando os microcontroladores.

Para avaliar a viabilidade do esquema de *hardware-in-the-loop* com os microcontroladores disponíveis, foram feitos testes simplificados com uma lei de controle elementar em execução no *ESP32* e com a planta emulada em execução no *Arduino UNO*. Os bons resultados sugerem que a metodologia utilizada deverá ser adequada também para suportar a lei de controle com valores quaisquer da lei de controle *PID*.

2.1 Procedimento Experimental

A IDE do *Arduino* foi configurada para haver compatibilidade com o *ESP32* e, em seguida, foram realizados testes de implementação das diversas funções do microcontrolador, sendo elas: conectividade *WiFi* e *MQTT*, temporizadores de *hardware* e leitura e escrita de sinais *PWM*. O envio de dados via *MQTT* foi testado a partir do cliente *MQTT HiveMQ*.

Em seguida, a implementação do *PID* foi feita a partir de testes de codificação das ações integral e derivada. Após obter resultados satisfatórios, o *PID* foi

implementado utilizando o temporizador de *hardware* na geração de uma interrupção para obter uma integração numérica precisa.

No esquema utilizado de *hardware-in-the-loop*, o microcontrolador *ESP32* é responsável pela leitura do sinal *PWM* da saída da planta (*Arduino*), leitura da referência fornecido por um potenciômetro e pela escrita do sinal de saída do *PID* em saída analógica.

A implementação da planta no *Arduino* foi feita através de duas funções de integração, definindo os ganhos da função da planta para diferentes testes.

As entradas e saídas dos microcontroladores tiveram de ser tratadas, pois cada controlador trabalha com níveis de tensão diferentes, números distintos de bits de resolução de leitura e escrita, e sinais *PWM* diferentes. Outro tratamento feito foi o do sinal de saída do *PID*, uma vez que o controlador pode apresentar valores negativos ou positivos, enquanto a tensão de saída varia apenas de 0 a 3,3V.

Na sequência foram feitos testes de funcionamento enquanto todas as funções do *ESP* eram utilizadas ao mesmo tempo, permitindo verificar se havia problemas de interrupção ou implementação do código.

3. Resultados

O controle da planta foi realizado para uma referência de 1 V, constantes K_p , K_i e K_d de 0,3, 0,5 e 0,1, respectivamente e constantes da planta K_g , K_1 e K_2 iguais a 1, 3 e 5, respectivamente. As constantes do controlador foram escolhidas através da técnica algébrica de cancelamento de polos. Na Figura 1 mostra-se a resposta simulada no MATLAB do sistema e a resposta real do sistema.

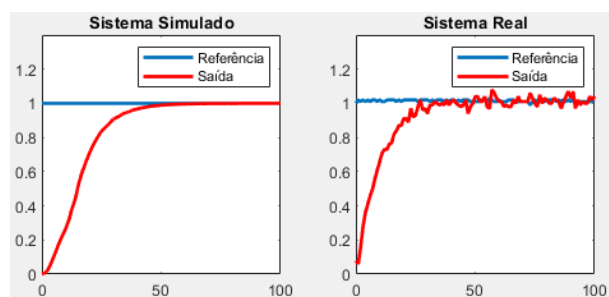


Figura 1 – Resposta simulada (esquerda) e real (direita)

4. Conclusões

O objetivo inicial da pesquisa, comparação de controlador e parametrização via IoT, foi adaptada para uma implementação de um esquema *hardware-in-the-loop* por meio de um microcontrolador *ESP32* onde deve estar a lei de controle e outro microcontrolador *Arduino Uno* onde deve estar a planta emulada.

A resposta do sistema a partir da implementação nos microcontroladores correspondeu bastante com a resposta obtida via *software*, portanto, atendeu as expectativas.

Algumas transformações lineares tiveram de ser implementados, como um fator para reduzir o tamanho do esforço de controle para que fosse possível sua conversão para tensão e outro fator para corrigir a saturação da ação integral do controlador.

As próximas etapas incluem a parametrização via IoT dos parâmetros do controlador, documentação geral do projeto e hospedagem do projeto no *Git*hub.

5. Referências

- [1] BARRETO, C. G. Tudo o que você precisa saber sobre controle PID - Automacao & Cartoons. Disponível em: <<https://automacaoecartoons.com/2019/01/23/control-pid/>>. Acesso em: 14 maio. 2020.
- [2] EDUARDES C. and SPURGEON, S.K. SMC Theory and Application. CRC press, 1998.
- [3] The MathWorks, I. (2020, May 11). What Is Hardware-in-the-Loop? - NI. Artigo Técnico. <https://www.ni.com/pt-br/innovations/white-papers/17/what-is-hardware-in-the-loop-.html>. Acesso em: 14 setembro. 2020.
- [4] ASTROM, K.J. PID controllers: Theory, design and tuning – 2nd ed. ISA, 1995.
- [5] GURU99. Internet of Things (IoT) Tutorial for Beginners: Introduction, Basics, Applications. Disponível em: <<https://www.guru99.com/iot-tutorial.html>>. Acesso em: 23 maio. 2020.
- [6] Conhecendo o MQTT – IBM Developer. Disponível em: <<https://developer.ibm.com/br/articles/iot-mqtt-why-good-for-iot/>>. Acesso em: 14 setembro. 2020.

Agradecimentos

À instituição Centro Universitário da FEI pelo suprimento dos equipamentos necessários para a realização do projeto.

¹ Aluno de IC do Centro Universitário FEI. Número do processo PBIC155/19. Projeto com vigência de 12/19 a 11/20.