XXVII Congresso Nacional de Estudantes de Engenharia Mecânica
08 a 12 de fevereiro de 2021, Curitiba, PR, Brasil

CONTROLE DE UMA PLANTA EMULADA COM PARAMETRIZAÇÃO VIA IOT

Bruno Lolli Rodrigues, lolli_bruno@hotmail.com[1]
Guilherme Braga de Paula, guilhermebragadepaula7@gmail.com[1]
Fabrizio Leonardi, fabrizio@fei.edu.br[1]

[1]Centro Universitário FEI, Av. Humberto Alencar Castelo Branco, 3972- B. Assunção, São Bernardo do Campo - SP.

Resumo. *Existem vários recursos comerciais de alto desempenho para a implementação de simulações com plantas emuladas, os chamados sistema de controle hardware-in-the-loop, porém muitas vezes inacessíveis para a maioria dos usuários pelos altos custos envolvidos. Objetivo deste trabalho é propor uma plataforma de baixo custo e uma metodologia para implementação de sistemas hardware-in-the-loop por meio das plataformas de prototipagem eletrônica Arduino UNO e ESP32, com parametrização via IoT. A motivação para a parametrização remota é viabilizar reprojeto do controlador e download dos seus parâmetros mesmo em locais de difícil acesso. Para ilustrar a metodologia, um controlador PID foi implementado num ESP32 e interligado a uma planta dinâmica de segunda ordem emulada no Arduino Uno e interligados por meios dos conversores A/D e D/A das plataformas. A metodologia proposta é baseada na implementação em tempo real das equações diferenciais por meio de simulação dos integradores da sua representação no espaço de estados e fazendo uso de interrupções por hardware. Os resultados obtidos foram comparados com simulações numéricas e indicam que o método pode ser usado em aplicações reais diversas.*

Palavras-chave*: IoT. Controle PID. Hardware-in-the-loop*

Abstract. *There are several high-performance commercial resources for implementing simulations with emulated plants, the so-called hardware-in-the-loop control system, but often inaccessible to most users due to the high costs involved. The objective of this work is to propose a low-cost platform and a methodology for implementing hardware-in-the-loop systems through the electronic prototyping platforms Arduino UNO and ESP32, with parameterization via IoT. The motivation for remote parameterization is to make it possible to redesign the controller and download its parameters even in places with difficult access. To illustrate the methodology, a PID controller was implemented in an ESP32 and connect to a second order dynamic plant emulated in Arduino Uno by means of the A/D and D/A converters of the platforms. The proposed methodology is based on the real-time implementation of differential equations by simulating the integrators of their representation in the state space and making use of hardware interrupts. The results obtained were compared with numerical simulations and indicate that the method can be used in different real applications.*

Keywords: *IoT. PID Control. Hardware-in-the-loop*

1. INTRODUÇÃO

Os sistemas Hardware-in-the-loop (HIL) podem ser entendidos como uma técnica para desenvolver controladores embarcados através da utilização de um firmware que emula o sistema a ser controlado de maneira fiel dentro de uma estrutura em malha fechada, ou seja, a emulação fica constantemente atualizando suas entradas e saídas em tempo real. Temos então uma estrutura onde o controlador é implementado num hardware e a planta emulada também é implementada num hardware porem distinto daquele do controlador, conforme ilustração da Figura 1. A ligação entre os dois subsistemas é física e, caso o modelo da planta seja fidedigno, o controlador reagirá da mesma forma com a planta emulada e com a planta real. Nesse cenário os resultados observados nesse esquema podem ajudar a prever os comportamentos que o sistema real apresentaria. Note-se que esse framework é bastante adequado para testes de novos controladores. As maiores vantagens do uso desse sistema são a redução de custos, segurança, redução do tempo necessário para testes e a possibilidade de repetição da simulação sem ocorrer o desgaste do sistema.

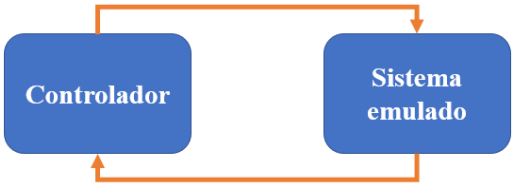


Figura 1. Representação do Sistema HIL, Autor

Os sistemas HIL surgiram dos desafios que aparecem no processo de prototipagem de projetos. Confeccionar um protótipo requer conhecimento e experiência e, muitas vezes, a aquisição de experiência pode levar ao aumento dos gastos com o projeto. Portanto, foi necessário encontrar um modo de implementar esse sistema real de maneira virtual, ou seja, por meio de um programa que reproduza de maneira mais precisa possível o comportamento em tempo real do sistema em que se deseja trabalhar.

Os sistemas HIL podem ser aplicados em diversos tipos de testes, como testes de sistemas de voo (Ronaldo Barros dos Santos et al., 2011), de performance de veículos como aquele disponível no software CarSim®. Existem também outros softwares que realizam essa simulação de maneira totalmente customizável, como o SIMULINK™ e o LabView™. Em resumo, o termo hardware-in-the-loop é o nome dado a uma técnica onde o controlador é conectado a um dispositivo (hardware) de teste que simula uma "realidade" desejada (National Instruments CORP., 2020).

Internet das Coisas (*Internet of Things - IoT*) é o nome dado para a tecnologia que conecta e permite transmissão de informações entre máquina e usuários através da rede, assim permitindo por exemplo o controle de um sistema de refrigeração de casa, ou até mesmo ter uma resposta do sistema nos informando do custo de energia.

Para um sistema IOT completo são considerados os seguintes tópicos (Yuan, 2017):

1) Sensores: com sensores é aberto a capacidade de coletar dados em tempo real, coletar dados como vídeo, áudio, temperatura, entre outros.

2) Conectividade: Todos os dados coletados são enviados para a nuvem através de redes moveis, Wi-Fi etc.

3) Processamento de dados: Temos algum software para processar as informações recebidas dos sensores. No caso deste projeto, teremos um sistema PID no ESP32 para o processamento dos dados recebidos para a planta.

4) Interface: As informações processadas devem estar disponíveis ao usuário, através de alguma forma, como uma mensagem SMS, e-mail, aplicativo, entre outros.

Para este projeto utilizou-se o MQTT (Message Queue Telemetry Transport), surgindo no final dos anos 90, e se tornando código aberto no final de 2014, por meio deste protocolo, é possível realizar a troca de dados em hardwares limitados, por ser um protocolo leve, e é assíncrono assim tornando o envio de dados mais rápido. (Barros, 2015)

Nesse sistema de comunicação, os dados são recebidos, processados e enviados através da administração do servidor MQTT chamado de broker, elemento que realiza a comunicação entre dois clientes. Assim, ao implementar esse método de comunicação no ESP32, o usuário poderá realizar a alteração de um dado em um cliente conectado ao servidor (broker) e após o dado ser processado, o broker reencaminha diretamente para o microcontrolador onde foi configurado outro cliente. Esse tipo de comunicação se baseia em "tópicos", ou seja, para que o cliente se comunique com outro, ambos devem estar inscritos no mesmo tópico.

Existem vários recursos comerciais de alto desempenho para a implementação de sistemas HIL, porém muitas vezes inacessíveis a maioria dos usuários pelos altos custos envolvidos. Objetivo deste trabalho é propor uma plataforma de baixo custo e uma metodologia para implementação de sistemas hardware-in-the-loop por meio das plataformas de prototipagem eletrônica Arduino UNO e ESP32, com parametrização via IoT. A motivação para a parametrização remota é viabilizar reprojeto do controlador e download dos seus parâmetros mesmo em locais de difícil acesso

2. METODOLOGIA

O procedimento proposto neste artigo consta de três etapas. Inicialmente as equações diferenciais ordinárias (EDO) do controlador e da planta são representadas no espaço de estados por meio de diagramas de blocos na forma canônica observável. Isso garante que cada EDO necessita apenas de tantos integradores quanto for sua ordem, independentemente do controlador ou da planta serem lineares ou não lineares.

2.1 REPRESENTAÇÃO POR DIAGRAMA DE BLOCOS

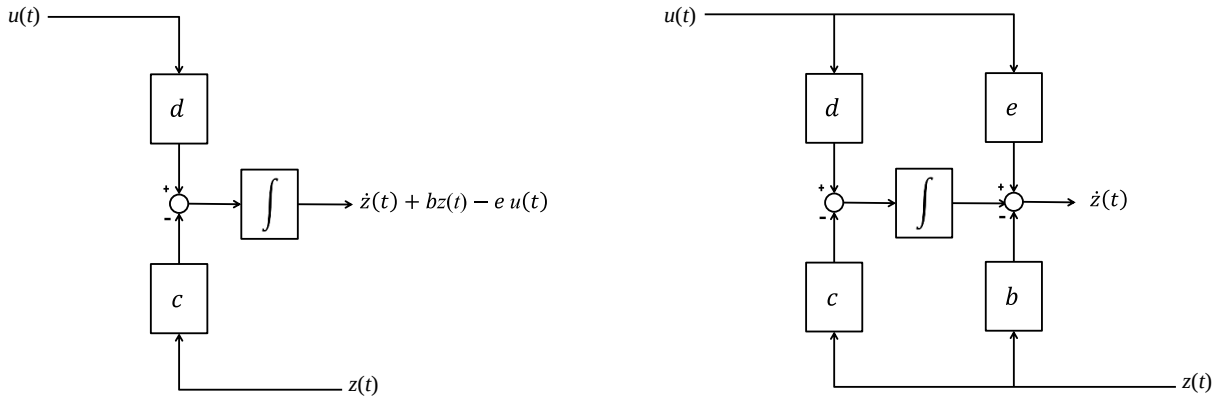
Para ilustrar o procedimento de representar uma EDO por meio de um diagrama de blocos com integradores, considere o exemplo a seguir onde a entrada é $u(t)$ e a saída é $z(t)$

$$\ddot{z}(t) + b\,\dot{z}(t) + c\,z(t) = d\,u(t) + e\,\dot{u}(t) \tag{1}$$

Para a construção do diagrama de blocos da forma canônica observável, representa-se as variáveis de entrada e de saída de menor zero (sem derivada)

$$d\,u(t) - c\,z(t) = \ddot{z}(t) + b\,\dot{z}(t) - e\,\dot{u}(t)$$

A partir da representação por blocos a operação do lado esquerdo, integra-se o resultado (vide Figura 2a) e removem-se os ternos de menor ordem por adição de parcelas de sinais contrários (vide Figura 2b).



a) Primeira etapa b) Etapas seguintes

Figura 2. Construção do diagrama de blocos canônico, Autor.

Por fim o sinal $\dot{z}(t)$ é integrado resultando no diagrama na forma canônica observável da Figura 3.

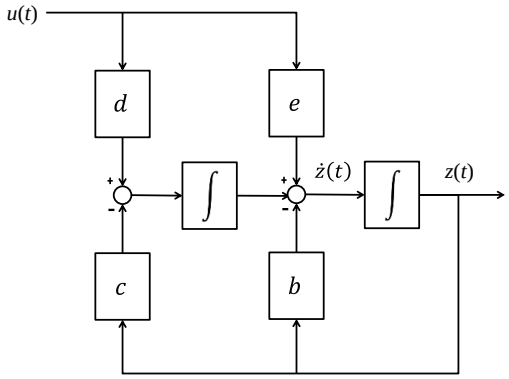


Figura 3. Diagrama de blocos da Eq. (1), Autor.

Note que mesmo havendo uma derivada do sinal de entrada, foi possível obter um diagrama apenas com dois integradores, ordem da derivada máxima de $z(t)$, e nenhum diferenciador, além de somadores e blocos de ganhos.

Existem outras formas de conversão de uma EDO num diagrama de blocos com integradores. Ilustra-se a seguir o procedimento aplicado para representar a EDO da planta utilizada nos experimentos deste artigo, descrita pela função de transferência de segunda ordem a seguir para $K_g = 1$, $K_1 = 3$ e $K_2 = 5$.

$$\frac{Y(s)}{R(s)} = \frac{K{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2} = \frac{K_g}{s^2 + K_1 s + K_2}$$

O procedimento inicia com a transformação da função de transferência numa EDO

$$\ddot{y}(t) = K_g u(t) - K_1 \dot{y}(t) - K_2 y(t) \tag{2}$$

Na sequência o lado direito é representado por um diagrama de blocos apenas com ganhos. O sinal resultante é então integrado duas vezes obtendo-se $y(t)$ como saída, conforme ilustrado na Figura 4.

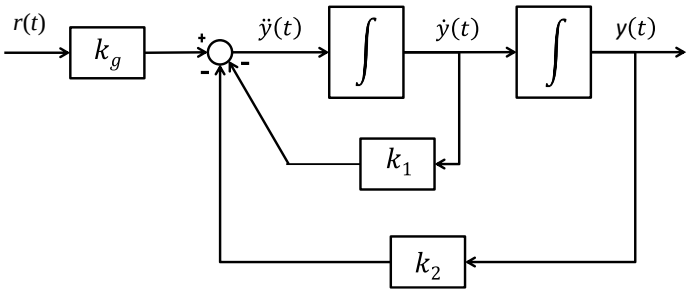


Figura 4. Diagrama de blocos da Eq. (2), Autor.

A seção seguinte da metodologia mostra como proceder a implementação do diagrama de blocos com integradores numa plataforma de prototipagem eletrônica tipo Arduino UNO ou ESP32.

2.2 IMPLEMENTAÇÃO DAS EQUAÇÕES DIFERENCIAIS

Para a implementação da EDO na plataforma de prototipagem Arduino UNO ou ESP32, cada um dos integradores deve ser convertido do tempo contínuo para o tempo discreto. Existem várias formas de se fazer isso, tais como aproximação de ordem zero (ZOH), aproximação bilinear (Tustim), etc. A discussão de qual técnica é mais interessante, depende da aplicação e, portanto, não é escopo deste trabalho. Por simplicidade, utilizou-se aqui a aproximação baseada na definição de derivada de uma função

$$\frac{dy(t)}{dt} = \lim_{T\to 0}\left(\frac{y(t+T)-y(t)}{T}\right) \cong \frac{y(t+T)-y(t)}{T}$$

Reescrevendo a aproximação para $t = kT$ para $k = \{0,1,2,\cdots\}$, podemos mapear cada integrador $\frac{1}{s}$ do diagrama de blocos da EDO correspondente da seguinte forma

$$s \rightarrow \frac{y(k+1)-y(k)}{T} \qquad \text{ou} \qquad \frac{1}{s} \rightarrow \frac{T}{y(k+1)-y(k)}$$

Dessa forma cada integrador i será descrito por uma equação de diferenças do tipo em que u_i é a i-ésima entrada e y_i é a i-ésima saída.

$$y_i(k+1) = y_i(k) + T\, u_i(k)$$

O restante do diagrama dos blocos da EDO correspondente, compreende apenas operações algébricas, imediatas de codificação.

Note-se que T corresponde a duração do intervalo de tempo entre kT e $(k+1)T$. Isso implica que cada rotina de integração precisa ler o valor u_i produzindo o valor y_i em intervalos fixos de tempo T. Para tanto, recomenda-se aqui a utilização de interrupções produzidas pelos temporizadores das próprias plataformas de prototipagem Arduino UNO ou ESP32. No caso da planta da Figura 4, utilizou-se interrupções de 1 [ms], tempo suficientemente pequeno para que a aproximação do integrador na forma discreta no Arduino UNO.

O fluxograma da Figura 5 ilustra a codificação da planta. Inicialmente as condições iniciais dos integradores devem ser impostas. O programa entra então em num ciclo de infinitas repetições que ocorrem a cada período de tempo T. O valor da referência r é obtida do conversor A/D e a expressão algébrica do primeiro somador e codificada. Na sequência, os dois integradores têm seus valores atualizados e a saída é enviada para o conversor D/A. A partir desse ponto a rotina fica aguardando a interrupção para que o ciclo se repita.

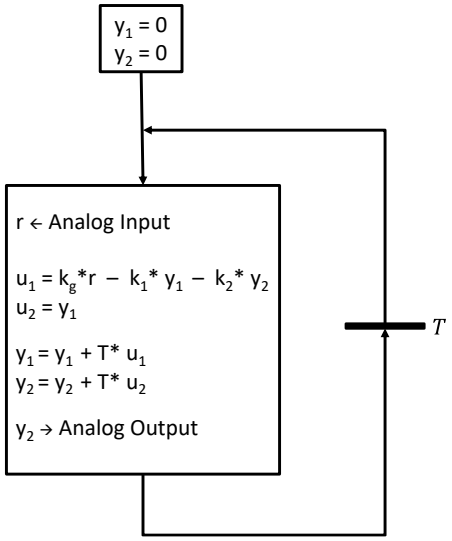


Figura 5. Fluxograma da síntese da EDO da planta no Arduino UNO, Autor.

Como ilustração um controlador PID foi implementado por meio da mesma metodologia da planta, mas na plataforma de prototipagem ESP32. O diagrama da Figura 6 ilustra como um PID pode ser realizado por meio de um diagrama de blocos com integradores e sem nenhum diferenciador

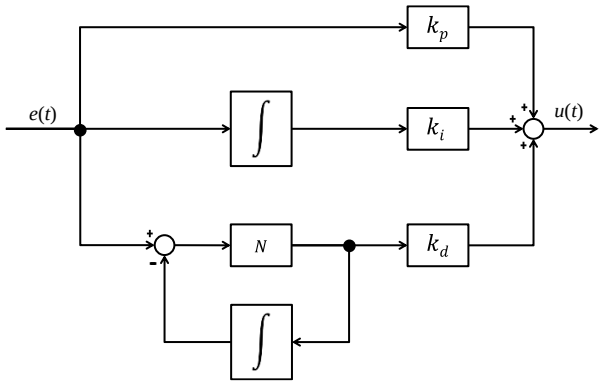


Figura 6. Diagrama de blocos da EDO do controlador PID, Autor.

O diagrama de blocos da Figura 6 implementa a função de transferência do PID onde N representa a filtragem do termo derivativo, um número normalmente "grande" escolhido por simulação

$$\frac{U(s)}{Y(s)} = k_p + \frac{k_i}{s} + \frac{s}{\frac{1}{N}s + 1}k_d \cong k_p + \frac{k_i}{s} + sk_d$$

Para ilustrar o sistema HIL operando, uma versão modificada da equação anterior, incluindo a ação anti-windup do integrador, foi implementada no ESP32 para valores das constantes do controlador obtidas num projeto por cancelamento de polos e zeros.

Para a implementação do HIL há outras questões a serem resolvidas, pois os microcontroladores possuem características próprias que devem serem levadas em conta. O Arduino UNO, por exemplo, trabalha com uma tensão de referência de 0 a 5V, ou seja, ele consegue obter e gerar sinais analógicos de amplitudes de no máximo 5V, com uma resolução de 10 bits, ou seja, o sinal analógico é mapeado para um valor entre 0 e 1023 e resolução de escrita de sinal PWM de 8 bits de resolução, ou seja, podem ser definidos valores de escrita entre 0 e 255. O ESP32 por sua vez trabalha com tensões de até 3,3V, com resoluções de escrita e leitura de sinais que podem variar entre 9 a 12 bits, sendo a resolução de 12 bits escolhida para este projeto. Por isso, a saída do microcontrolador Arduino UNO foi limitada a 3,3V para compatibilidade com o nível máximo do ESP32.

Os sinais da lei de controle podem ser valores negativos, apesar dos sinais de entrada e saída dos conversores serem valores somente positivos. Portanto, para superar este problema, uma mudança de referencial foi implementada de forma

que sinais de tensão entre 0 a 1,65V representam sinais negativos, enquanto valores positivos ficam entre 1,65 e 3,3V. Também foi feito outro tratamento para minimizar a perda de precisão, os sinais de entrada do ESP32 foram multiplicados por uma potência de dois para que assim fosse possível trabalhar com valores numéricos com uma range maior e não haver perdas por arredondamento. No Arduino UNO, foi utilizada a mesma estratégia, chamando de Compressão a ação de transformar o sinal multiplicado pela potência de dois em um valor dentro da faixa da resolução de saída dos microcontroladores e de Descompressão a ação de multiplicar os valores de leitura pela potência de dois.

A ação integral do Controlador PID foi tratada de modo a eliminar o efeito do fenômeno de windup. Para isso, foram estipulados limites máximos e mínimos em que a ação integral poderia alcançar, com base em um fator definido pelo usuário que pode escolher esses limites. O valor numérico do sinal de controle foi tratado para que não saturasse a saída do ESP32.

2.3 PARAMETRIZAÇÃO IOT

A parametrização IoT foi implementada utilizando o broker público HiveMQ e o aplicativo gratuito MQTT Dash do sistema operacional Android para o cliente que realizará a parametrização. As bibliotecas PubSubClient e WiFi realizaram a conexão do ESP32 com a Internet e com o broker, sendo a primeira utilizada para criar o cliente MQTT do ESP32, realizar o recebimento e envio de dados para o broker e a inscrição nos tópicos e a segunda biblioteca conecta o ESP32 com a rede WiFi desejada.

Assim, foram criados tópicos para cada um dos parâmetros do controlador e um tópico para o envio da resposta da planta, para que seja possível monitorá-la também a distância. O aplicativo MQTT Dash é customizável, ou seja, é possível adicionar elementos como campos de texto, imagens e barras de progresso, onde cada elemento deve ser configurado para conectar ao tópico desejado. Neste projeto, foi utilizado o campo de texto para receber os dados enviados pelo ESP32 e para realizar a parametrização. A Figura 7 mostra o layout do aplicativo.

Figura 7. Layout MQTT Dash

3. RESULTADOS

O controle da planta emulada foi realizado para uma referência de 1 V, constantes Kp, Ki e Kd do controlador PID de 0,3, 0,5 e 0,1, respectivamente e constantes Kg, K_1 e K_2 da planta iguais a 1, 3 e 5, respectivamente. As constantes do controlador foram escolhidas através da técnica algébrica de cancelamento de polos (Maya & Leonardi, 2011). A Figura 8 mostra a resposta simulada e a resposta real do sistema.

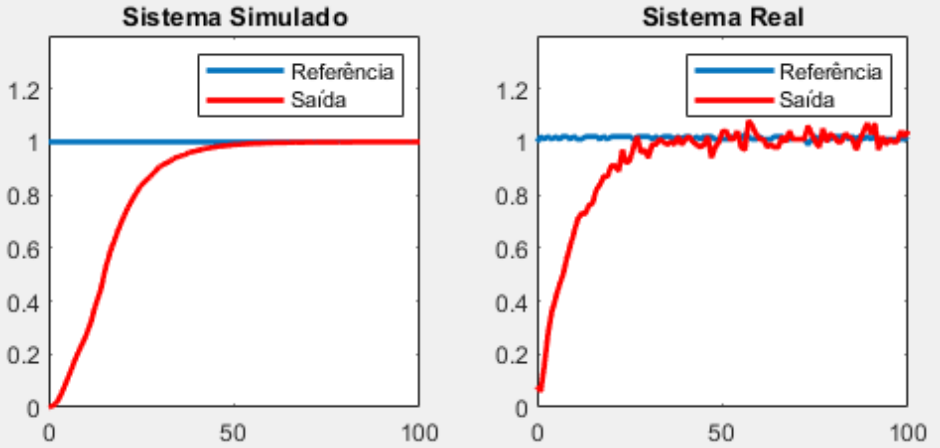


Figura 8. Resultado 1 - Resposta simulada (esquerda) e real (direita), Autor.

A Figura 9 mostra outra simulação, desta vez com constantes Kp, Ki e Kd do controlador PID de 1, 0,3 e 0,1, respectivamente e constantes Kg, K_1 e K_2 da planta iguais a 1, 10 e 3, respectivamente.

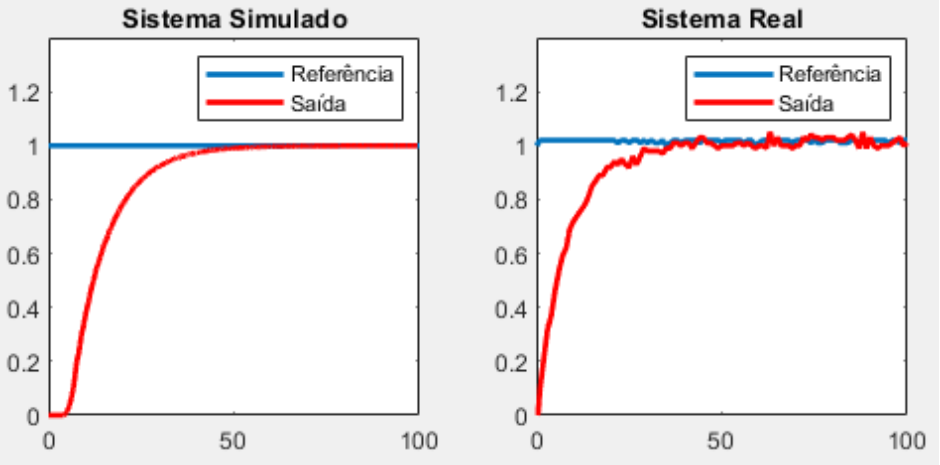


Figura 9. Resultado 2 - Resposta simulada (esquerda) e real (direita), Autor.

4. CONCLUSÕES

Este trabalho propôs o uso de uma plataforma de baixo custo e uma metodologia para implementação de sistemas hardware-in-the-loop por meio das plataformas de prototipagem eletrônica Arduino UNO e ESP32, com parametrização via IoT. A metodologia proposta é baseada na implementação em tempo real das equações diferenciais por meio de simulação dos integradores da sua representação no espaço de estados e fazendo uso de interrupções por hardware.

O código obtido é bastante simples e modular, ou seja, quando a ordem da equação diferencial cresce aumenta-se proporcionalmente o número de integradores. Essa facilidade corrobora para que seja viável a implementação das funções da planta ou controlador em microcontroladores de baixo custo.

A partir da implementação das EDO's da planta e do controlador, cada um numa plataforma de prototipagem eletrônica, o desempenho em malha fechada foi comparado com simulações numéricas no Simulink. Os bons resultados obtidos para essa planta e controlador particulares sugerem que a metodologia pode ser estendida para outras plantas e controladores que possam ser representados no espaço de estados.

A parametrização IoT respondeu de maneira razoavelmente rápida, sendo possível a troca dos parâmetros do controlador em tempo real e a distância, sem a necessidade de ambo o cliente e o microcontrolador estarem em uma rede local. Logo, o atraso de resposta entre os clientes é definido pelo broker utilizado, sendo o atraso maior em brokers públicos. Em resumo, o microcontrolador trabalhou bem realizando as funções de controle e comunicação MQTT, sendo, portanto, uma alternativa viável para projetos que requerem essa interação remota.

Como proposta de continuidade deste trabalho propõe-se a utilização de modelos não lineares e dinâmicas mais rápidas para se tentar avaliar desempenhos limites da arquitetura e do framework.

6. REFERÊNCIAS

Barros, M. (2015, June 15). *MQTT Publish/Subscriber - Protocolos para IoT - Embarcados.* https://www.embarcados.com.br/mqtt-protocolos-para-iot/

Maya, P. A., & Leonardi, F. (2011). *Controle Essencial* (1ª). Pearson Education do Brasil.

National Instruments CORP. (2020, May 11). *What Is Hardware-in-the-Loop? - NI*. https://www.ni.com/pt-br/innovations/white-papers/17/what-is-hardware-in-the-loop-.html

Ronaldo Barros dos Santos, S., Nascimento Givigi Junior, S., Lúcio Nascimento Júnior, C., Bittar, A., & Maria Franco de Oliveira, N. (2011). *MODELING OF A HARDWARE-IN-THE-LOOP SIMULATOR FOR UAV AUTOPILOT CONTROLLERS.* http://www.ele.ita.br/~neusa/MODELING OF A HARDWARE-IN-THE-LOOP SIMULATOR FOR UAV AUTOPILOT CONTROLLERS.pdf

Yuan, M. (2017, October 4). *Conhecendo o MQTT*. https://developer.ibm.com/br/articles/iot-mqtt-why-good-for-iot/

7. RESPONSABILIDADE PELAS INFORMAÇÕES

Os autores são os únicos responsáveis pelas informações incluídas neste trabalho.