

Análise de performance do controlador PI utilizando simulação HIL com Arduino UNO

Performance analysis of PI controller using HIL simulation with Arduino UNO

RESUMO

Luís Felipe de Paula Miranda
lfeliphemiranda@gmail.com
Universidade Tecnológica Federal do Paraná, Cornélio Procopio, Paraná, Brasil

Emerson Ravazzi Pires da Silva
emersonr@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procopio, Paraná, Brasil

A plataforma Hardware-in-the-loop (HIL) consiste em uma técnica na qual uma parte do processo é feita de maneira embarcada em um microcontrolador. O sistema embarcado simula o comportamento do sistema físico real, a ideia é simular apenas uma parte do processo, podendo embarcar o controlador ou as equações que regem o comportamento do sistema físico modelado/estimado. Neste trabalho, é apresentado o controle de um circuito RC (resistor-capacitor) através de um controlador Proporcional-Integral (PI) utilizando a implementação HIL. A resposta HIL será comparada com a resposta simulada, usando o modelo matemático do circuito RC, e com a resposta real do circuito. simulação HIL vem se tornando cada vez mais usual pelo fato de tal técnica permitir a aplicação das crescentes complexidades dos sistemas. O HIL elimina a necessidade da construção imediata de uma planta física, pois algumas plantas são caras em demasia e, em alguns casos, de difícil acesso para testes como, a título de exemplo, plataformas de petróleo, robôs industriais, aeronaves, entre outros diversos tipos de sistemas.

PALAVRAS-CHAVE: Hardware in The Loop. Controle PI. Arduino UNO.

ABSTRACT

The Hardware-in-the-loop (HIL) platform consists of a technique in which part of the process is carried out in a microcontroller. The embedded system simulates the behavior of the real physical system, the idea is to simulate only part of the process, being able to embark the controller or the equations that govern the behavior of the modeled / estimated physical system. In this work, the control of an RC circuit (resistor-capacitor) through a proportional-integral controller (PI) using the HIL implementation is presented. The HIL response will be compared with the simulated response, using the mathematical model of the RC circuit, and with the actual response of the circuit. HIL simulation is becoming more and more usual due to the fact that this technique allows the application of the increasing complexities of the systems. HIL eliminates the need for the immediate construction of a physical plant, as some plants are too expensive and, in some cases, difficult to access for tests such as, for example, oil platforms, industrial robots, aircraft, among others types of systems.

KEYWORDS: Hardware in The Loop. PI Control. Arduino UNO.

Recebido: 19 ago. 2020.

Aprovado: 01 out. 2020.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

Diferente da simulação computacional, que é feita em ambiente completamente virtual, e os sinais que geram o erro de regime são idealizados, a plataforma HIL aproxima o comportamento da planta, já que na prática há falhas da comunicação, taxa de amostragem do microprocessador, aquecimento, ruído, entre outras interferências, (Sun, 2011). Três pilares dão fomento para a utilização da plataforma HIL, sendo elas:

A iteração de teste e *design* ocorre como se o sistema do mundo real estivesse sendo usado. Pode-se executar facilmente milhares de cenários possíveis para exercitar adequadamente um controlador sem o custo e o tempo associados aos testes físicos reais, (Instruments, 2019). Acredita-se que o *design* HIL teve premissa na indústria aeroespacial onde se buscava testes de sistemas de voo complexos. Por exemplo, Sun (2011) utilizou a simulação HIL para implementar o controle de uma aeronave não tripulada *Unmanned Aerial Vehicle* (UAV) operada por piloto automático.

a) Tempo: Tendo posse da função de transferência do sistema, é rápido construir a plataforma HIL;

b) Custo: Vários microcontroladores/placas de desenvolvimento podem ser utilizados, ESP32, Arduino UNO, DSP entre outros. Isso permite que com baixo investimento a técnica possa ser implementada.

c) Segurança: Como vários sistemas físicos dispõem de perigo para testes, realizar uma verificação HIL aumenta a segurança uma vez que a resposta encontrada para uma estratégia de controle HIL será bem próxima à real.

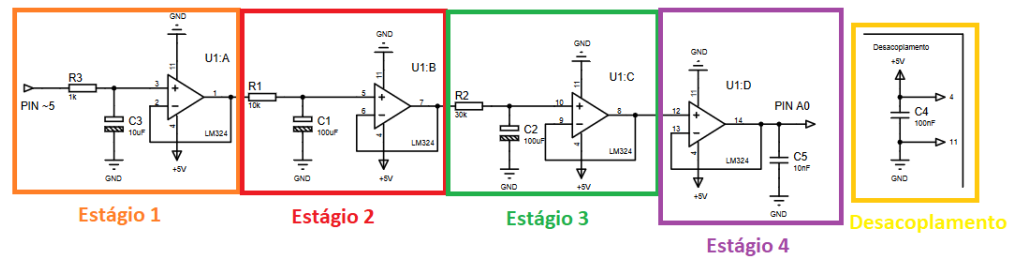
Neste estudo, a implementação HIL será realizada com aporte da placa de desenvolvimento Arduino UNO.

MATERIAIS E MÉTODOS

FUNÇÃO DE TRANSFERÊNCIA DO SISTEMA

O esquemático da Figura 1 representa um circuito Resistor-Capacitor (RC) em cascata, com casamento de impedâncias dada pelo amplificador operacional LM324N e configuração buffer de sinal, cuja função de transferência é vista na Eq. (1). No primeiro estágio do circuito tem-se como entrada a saída digital PWM do Arduino PIN ~ 5, onde um filtro RC passa-baixas foi incorporado para conversão em níveis DC. Nos estágios 2 e 3 temos a função de transferência da planta a ser controlada, e no estágio 4 a técnica de buffer do sinal da planta. Foram utilizados capacitores de 10 e 100 nF com a finalidade de desacoplar níveis AC (ruídos) do sinal de entrada e saída. A ilustração deve ser citada no texto e inserida o mais próximo possível do trecho a que se refere.

Figura 1 – Esquemático do circuito RC em cascata.



Fonte: Autoria própria (2020).

$$G(s) = \frac{1/3}{s^2 + 2/3s + 1/3} \quad (1)$$

ERROS EM REGIME PERMANENTE

Como visto na Eq.(1) o sistema é de 2ª ordem e do tipo 0, ou seja, para entrada degrau unitário o sistema tem erro finito e regime permanente. Como o intuito do projeto é eliminar este erro sem alterar significativamente o transitório do sistema em malha fechada, aumenta-se o tipo do sistema, projetando um controlador Proporcional-Integrativo (PI) no qual um polo na origem é inserido e assim o sistema passa a ser do tipo 1, a Tabela 1 mostra a relação dos tipos de entrada com os erros em regime permanente.

Tabela 1 – Erros de regime permanente.

Tipo do sistema	$e_{degrau}(\infty)$	$e_{rampa}(\infty)$	$e_{parabola}(\infty)$
1	$\frac{x_0}{1 + K_p}$	∞	∞
2	0	$\frac{v_0}{K_v}$	∞
3	0	0	$\frac{a_0}{K_a}$

Fonte: Adaptado de Ogata (2010).

Note que, x_0 , v_0 e a_0 são as entradas degrau unitário, rampa e parábola, respectivamente. Os ganhos de posição, velocidade e aceleração K_p , K_v e K_a são definidos pela Eq.(2).

$$K_{p,v,a} = \lim_{s \rightarrow 0} s^{n-1} A(s) \quad (2)$$

Sabendo que $n - 1$ diz respeito aos números de polos do tipo de entrada, por exemplo, o degrau unitário tem um polo na origem, logo $s^{1-1} = 0$, a entrada rampa tem depois polos na origem, implicando s^1 e assim sucessivamente.

PROJETO DE UM CONTROLADOR PI

Segundo Ogata (2010), os controladores PI podem ser usados para melhorar a resposta do sistema de controle em regime permanente, aumentando a precisão da resposta; sem alterar significativamente o desempenho em regime transitório.

Com o intuito de testar a plataforma HIL, e verificar se o comportamento da planta controlada será equivalente ao controle da planta real, projeta-se um controlador bastante usual, o controlador Proporcional-Integral (PI). Para projetar o controlador PI, os seguintes requisitos devem ser atendidos:

- Dado o circuito indicado na Figura 1, projetar o controlador PI de forma a anular o erro estacionário de posição, mas sem alterar o grau de amortecimento (ζ) para manter tanto quanto possível a mesma resposta.

Os polos do sistema em malha fechada são: $s_{1,2} = -\frac{2}{3} \pm \frac{\sqrt{2}}{3}j = 0.8165 \angle 144,7359^\circ$.

Com isso o valor da frequência natural do sistema é $\omega_n = 0,8165$, implicando-se assim que $\zeta = 0,8165$, porque o objetivo é manter similar a resposta transitória do sistema, isto implica manter a parte real do polo dominante, posto isso $-\zeta\omega_n = -\frac{2}{3}$.

O controlador PI tem a função de transferência vista na Eq.(3)

$$G_c(s) = k_p + \frac{k_i}{s}, x_i = \frac{k_i}{k_p} \Rightarrow \frac{k_p(s+x_i)}{s} \quad (3)$$

Segundo Ogata (2010), para fins de projeto, utilizar um zero na posição $-x_i$, que seja ao menos 10 vezes menor que a parte real do polo dominante, fará que o zero inserido pouco afetará a resposta transitória do sistema em malha fechada. Deste modo $x_i = \frac{1}{15}$.

Utilizando o *software* MATLAB®/Simulink® com o comando *rltool(G)*, obteve-se o ganho $k_p = 1.0382$, o controlador encontrado tem função de transferência da Eq.(4).

$$G_c(s) = \frac{1,0382(s+1/15)}{s} \quad (4)$$

Representação discreta do sistema e do controlador

(Nise, 2015) mostra que para a transformação de um sistema no domínio contínuo para o domínio discreto, Z , pode ser representado pela Eq.(5).

$$H(z) = (1 - z^{-1})Z \left\{ \mathcal{L}^{-1} \left(\frac{G(s)}{s} \right) \right\}_{t=nT} \quad (5)$$

Neste momento $H(z)$ é a representação discreta da função de transferência $G(s)$. O controlador discreto ($G_c(z)$) é obtido pelo mapeamento da função de transferência do controlador contínuo para o sistema discreto. Tal representação, $z = e^{-sT}$, permite mapear os polos e zeros do controlador da encontrado na Subseção 2.4 para o plano

discreto. Pelo teorema do valor final, encontra-se o ganho associado para a representação discreta. O mapeamento conta com um ganho DC de $K = 1.0382$.

A planta no domínio discreto tem função de transferência visto na Eq.(6):

$$H(z) = \frac{Y(z)}{R(z)} = \frac{0,0004075z + 0,0003986}{z^2 - 1,935z + 0,9355} \quad (6)$$

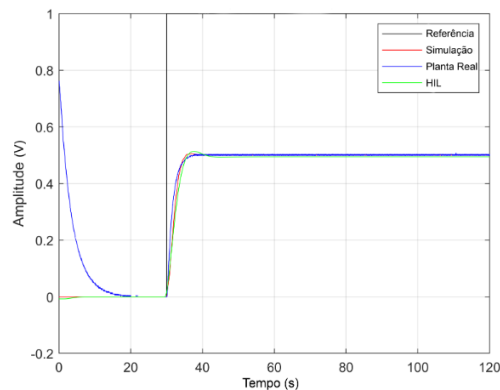
A equação de diferenças do sistema ao qual será embarcada no microcontrolador é vista na Eq.(7).

$$y[n] - 1,935y[n - 1] + 0,9355y[n - 2] = 4,075 \cdot 10^{-4}r[n - 1] + 3,986 \cdot 10^{-4}r[n - 2] \quad (7)$$

RESULTADOS

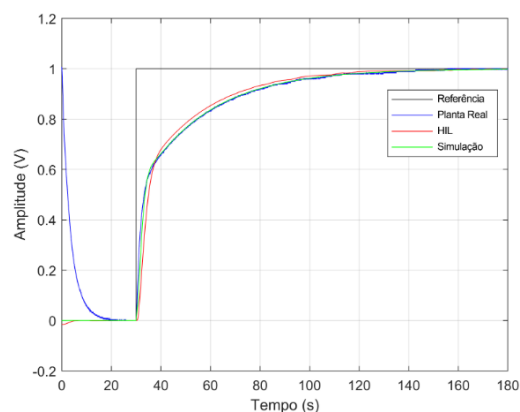
A comunicação do Arduino com o MATLAB®/Simulink® foi configurada segundo CONTROL TUTORIALS FOR MATLAB® & SIMULINK® (2019). Os resultados para a simulação, ensaio da planta real e HIL podem ser vistos na Figura 2 e Figura 3.

Figura 2 – Resposta dos ensaios HIL, planta real e simulação sem a ação do controlador.



Fonte: Autoria própria (2020).

Figura 3 – Resposta dos ensaios HIL, planta real e simulação com a ação do controlador.



Fonte: Autoria própria (2020).

CONCLUSÃO

Considerando os requisitos de projeto estipulados, como eliminar o erro de regime permanente para a entrada degrau, e manter o fator de amortecimento do sistema em malha fechada, o controlador PI agiu como o esperado no controle do circuito RC. A Figura 3 mostra que a metodologia HIL ficou bem próxima da simulação e do ensaio prático. Era de se esperar alguma divergência no resultado, o que ocorreu, porém ínfima. Pelo fato de ter utilizado a comunicação serial e o microcontrolador não ter a capacidade de alto processamento, eventuais *delays* e interrupções abruptas, fato este que se dá pelo computador no qual o Arduino foi conectado não ter dedicação somente para a comunicação serial, são esperadas. Não obstante, a plataforma HIL mostrou-se bem eficaz, abrindo margens para testes de outras plantas e até mesmo controle no espaço de estados discreto com características SISO e (MIMO) *Multiple-Input Multiple-Output*.

REFERÊNCIAS

CONTROL TUTORIALS FOR MATLAB® & SIMULINK®. **Simulink ArduinoIO Package**.

Disponível em:

http://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_IOPack#3. Acesso em: 2 ago. 2020.

Katsuhiko Ogata. **Modern Control Engineering**. Prentice Hal, 5 edition, 2010

NATIONAL INSTRUMENTS.. **What Is Hardware-in-the-Loop?**. Disponível em:

<https://www.ni.com/pt-br/innovations/white-papers/17/what-is-hardware-in-the-loop-.html>. Acesso em: 1 ago. 2020.

NISE, Norman S.. **Control Systems Engineering**. 7. ed. United States of America: Wiley, 2015. p. 1-943.

Yun-Ping Sun. **The Development of a Hardware-in-the-Loop Simulation System for Unmanned Aerial Vehicle Autopilot Design Using LabVIEW**. Disponível em:

<https://www.intechopen.com/books/practical-applications-and-solutions-using-labview-software/the-development-of-a-hardware-in-the-loop-simulation-system-for-unmanned-aerial-vehicle-autopilot-de>. Acesso em: 1 ago. 2020.