

12º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2021

SISTEMA HARDWARE IN THE LOOP PARA SIMULAÇÃO DE CONTROLE DE UM MOTOR CC

HERBERT G. MIRANDA¹, AGUINALDO C. DA COSTA FILHO²

¹Graduando em Engenharia de Controle e Automação, bolsista PIBIFSP, IFSP, Câmpus São José dos Campos, herbert.gomes@aluno.ifsp.edu.br.

²Docente EBTT Engenharia de Controle e Automação, IFSP, Câmpus São José dos Campos, aguinaldo@ifsp.edu.br.
Area de conhecimento (Tabela CNPq): 3.04.02.05-0 Sistemas Eletrônicos de Medida e de Controle.

RESUMO: Propõe-se a construção de um simulador para o sistema de controle de um motor CC, que permitirá a alteração da sua lei de controle com objetivo de analisar as respostas a diversas excitações. As excitações recebidas pelo sistema de controle no simulador serão provenientes do sinal analógico enviado do Arduino, o qual será gerado a partir de um potenciômetro instalado na placa de desenvolvimento. Por isso pretende-se também desenvolver um ambiente de comunicação entre o Arduino e o computador, utilizando o programa MATLAB[®]. Uma janela no simulador ilustrará a resposta do sinal e a velocidade de rotação do motor (em rpm), que permitirá observar como o sistema de controle está se comportando com essas excitações.

PALAVRAS-CHAVE: Controle de um motor CC; *Hardware in the Loop*; Simulação de um motor CC.

HARDWARE IN THE LOOP SYSTEM FOR A DC MOTOR SIMULATION

ABSTRACT: It is proposed to build a simulator for the control system of a DC motor, which will allow the modification of its control law in order to analyze the responses to different excitations. The excitation received by the control system in the simulator will come from the analog signal sent from the Arduino, which will be generated from a potentiometer installed on the development board. Therefore, it is also intended to develop a communication environment between the Arduino and the computer, using the MATLAB[®] program. A window in the simulator will illustrate the signal response and the engine rotation speed (in rpm), which will allow to observe how the control system is behaving with these excitations.

KEYWORDS: DC motor control, Hardware in the Loop, DC motor simulation.

INTRODUÇÃO

Dentro do universo de máquinas elétricas os motores são aquelas capazes de converter energia elétrica em energia mecânica, eles são divididos em CA (corrente alternada) e CC (corrente contínua) (CHAPMAN, 2013, p.1)

Por mais que atualmente os motores de indução, CA, sejam os preferidos nas aplicações que exigem controle de velocidade, por conta do avanço tecnológico dos VFD (*variable-frequency driver*) ou Conversor de frequência variável, o uso de motores CC continua popular. Quando se dispõe de um sistema de potência com corrente contínua, faz sentido utilizar um motor CC, como por exemplo em aviões, carros, tratores, drones e entre outros veículos. (CHAPMAN, 2013, p.465).

O motor simulado nesse trabalho é um *brushless* DC (BLDC), não escovado CC, o qual é virtualmente muito parecido com o motor de indução CA, os dois possuem três conjuntos de enrolamentos distribuídos no estator, a diferença fica no rotor e na técnica de controle. (RIPPEL, 2007)

Como pode-se ver na figura 1, o rotor do BLDC é composto por pares de ímãs permanentes, diferentemente do motor de indução, onde seu rotor é composto por finas chapas de metal que são curto-

circuitadas. Na figura 1 apresentada, o rotor é a parte externa, mas também podemos encontrar configurações onde a parte girante é a interna.

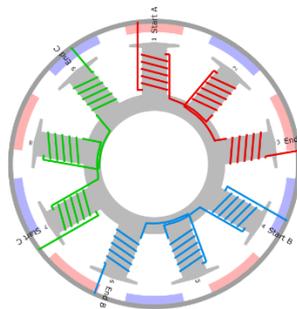


FIGURA 1. Estrutura de um motor BLDC. Fonte: NIESSEN, 2021. Disponível em: <<https://www.bavaria-direct.co.za/scheme/calculator/>>

Motores desse tipo demandam da utilização de um sistema eletrônico complexo. Estudar e aprimorar o desempenho desses sistemas é necessário para garantir o funcionamento preciso desses motores, já que em alguns casos o mal funcionamento pode causar graves acidentes. Pelos motivos citados e outros que o controle automático se torna essencial, não apenas para motores, mas também para qualquer campo da engenharia e da ciência. (OGATA, 2005, p.1)

A utilização de um simulador é primordial para que, antes mesmo de ligar o motor CC real, seja possível observar como ele se comportará, e como o sistema de controle responderá ao receber as excitações, evitando assim possíveis falhas na execução do projeto. Inicialmente esse trabalho propõe a construção de um simulador e de um sistema de controle para um motor CC genérico, os quais receberão os dados de um dispositivo físico.

O objetivo principal do projeto é construir um programa utilizando o MATLAB que seja capaz de simular um motor CC e seu respectivo sistema de controle, mostrando em tempo real a resposta, velocidade de rotação do motor, em RPM. O sinal de referência desse programa será proveniente do hardware externo utilizado. Antes de alcançar o objetivo principal é necessário entender como se dá a comunicação entre o MATLAB e o Arduino que será constatado pela realização de atividades adicionais, as quais subsidiarão as informações de como se dá a leitura e escrita entre o programa no computador e a placa de desenvolvimento. A figura 2 ilustra o diagrama de blocos do projeto.

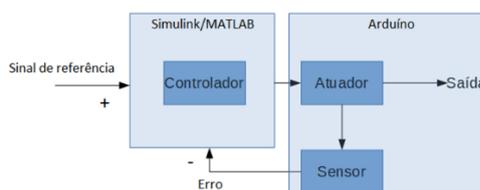


FIGURA 2. Diagrama de blocos referente ao projeto. Fonte: Próprio autor.

MATERIAL E MÉTODOS

O trabalho foi dividido em atividades conforme descrito a seguir. Todas elas realizadas utilizando o auxílio de um Arduino, que é uma plataforma de desenvolvimento de projetos eletrônicos, constituída tanto de hardware quanto de software (LOUSADA, 2020). O Arduino mega 2560 utilizado tem como componente principal o microcontrolador ATmega2560.

ATIVIDADE 1 – Conexão entre o programa MATLAB® e o Arduino: A base para o funcionamento desse projeto se fundamenta na comunicação entre o simulador e o hardware (neste caso o Arduino Mega 2560). Faz-se necessário a instalação de pacotes complementares, que além de permitir essa comunicação serial, também disponibilize ferramentas como o suporte a motores, sensores, leitura e escrita tanto analógica quanto digital, as quais são necessárias para o desenvolvimento do projeto.

Após a instalação dos pacotes foi necessário configurar o MATLAB® para reconhecer a placa, selecionando manualmente o modelo e a porta COM no qual ele estava conectado. Também foi preciso atualizar os drivers do Arduino de forma manual, no Gerenciador de dispositivos do Windows,

utilizando os arquivos disponíveis nas pastas dos pacotes complementares instalados. Isso foi necessário por conta da versão utilizada do *software*.

ATIVIDADE 2 – Teste com saídas digitais: Após a instalação e configuração dos pacotes foi necessário testar o funcionamento. O primeiro teste realizado foi enviar dados digitais do MATLAB® para o Arduino, utilizando-se de um pequeno programa criado para sinalizar leds que devem acompanhar o nível de um sinal senoidal. Quanto mais próximo dos picos positivos e negativos do sinal, mais leds acendiam, criando também uma espécie de “sinal” com eles. Para essa atividade, um sistema físico foi montado, utilizando-se dez (10) resistores de 300Ω, dez (10) leds brancos, onze (11) cabos *jumper* e um Arduino Mega 2560. Vide figura 3.

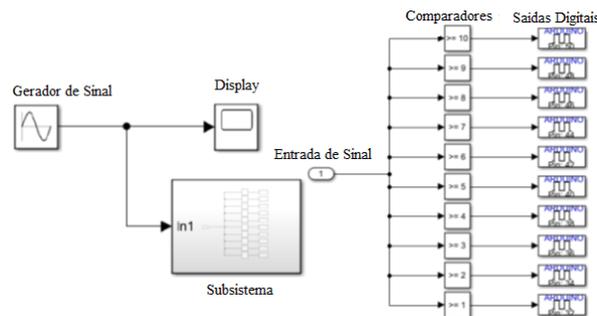


FIGURA 3. Programa de saída digital e subsistema. Fonte: Próprio autor.

ATIVIDADE 3 – Teste com entrada analógica: Após verificar se os dados eram enviados com sucesso para o Arduino, era necessário verificar se o contrário poderia ser feito, se o hardware conseguiria enviar dados para o MATLAB®. Para isso utilizou-se um arranjo eletrônico com um potenciômetro utilizado para variar a entrada de tensão, de 0 a 5V, em uma porta analógica do Arduino e verificar se era possível observar isso em outro programa criado, por meio de uma janela no MATLAB®.

ATIVIDADE 4 – Teste com entrada e saída: Verificadas as possibilidades de entrada e saída de dados, um programa para testar as duas ações simultaneamente foi criado. O novo programa contou com um bloco para a entrada do sinal analógico, um subsistema responsável pelo processamento do sinal e acendimento dos leds e uma janela no simulador para acompanhar o sincronismo. Um outro circuito foi montado como a junção dos dois anteriores. A figura 4 ilustra esse programa.

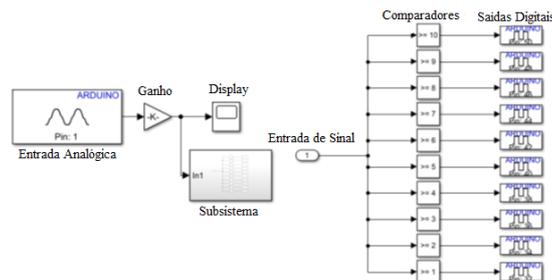


FIGURA 4. Programa (MATLAB) de entrada e saída e subsistema. Fonte: Próprio autor.

ATIVIDADE 5– Simulação do motor: Compreendido o funcionamento de entrada e saída de sinais, foi construído um modelo de simulação para um motor BLDC, que utilizou controle para ajustar a rotação medida no eixo para que esta fique próxima ao número de rotações (inserido com o auxílio de um potenciômetro). Para isso, um novo programa foi criado utilizando o Simulink mostrado na figura 5.

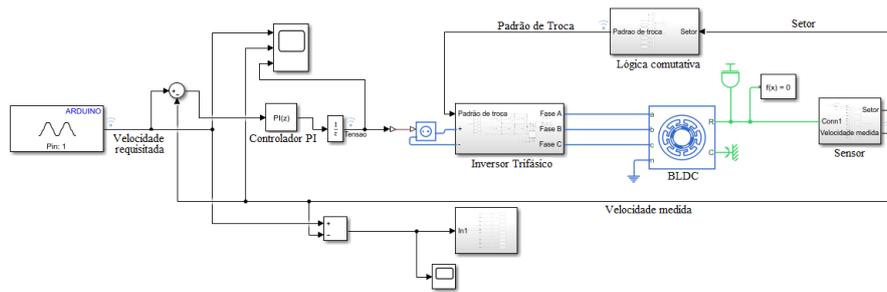


FIGURA 5. Simulador do motor BLDC. Fonte: Próprio autor.

Como é possível observar na figura 5, o sistema é composto por diversos subsistemas, e de um bloco em cor azul, no qual são inseridas as características do motor. A figura 6 permite observar um desses subsistemas, nesse caso o subsistema que simula os sensores de efeito *hall*.

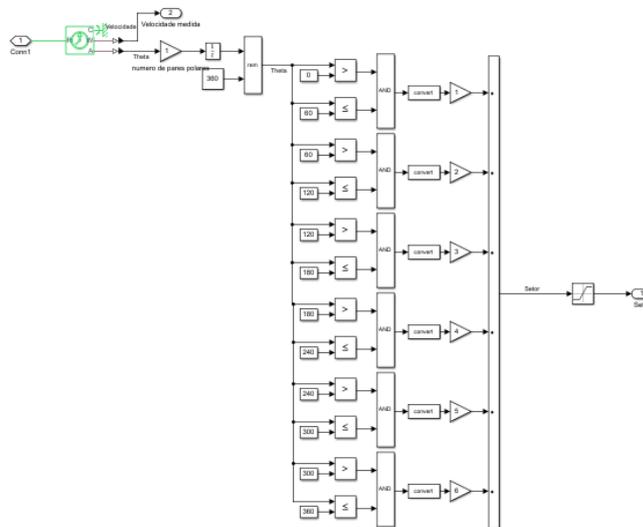


FIGURA 6. Subsistema de sensores. Fonte: Próprio autor.

É possível obter duas informações do motor, a velocidade do seu eixo, e a posição do seu polo magnético, através da utilização de três sensores de efeito *hall*. Pode-se dividir os 360° em seis setores diferentes: Entre 0° e 60°, 60° e 120°, 120° e 180°, 180° e 240°, 240° e 300°, 300° e 360°. Após identificar qual setor que o polo está, é necessário criar um comando para que ele siga para o setor seguinte. Isso é realizado pela lógica comutativa.

Em seguida, o sinal que varia entre os setores 1 a 6, apresenta a lógica comutativa como responsável por enviar os vetores de comando que irão ativar o circuito inversor trifásico de forma que o polo magnético passe para o próximo setor. Após selecionar a fase receberá tensão positiva ou negativa, é preciso conectá-la ao motor. Como se trata de uma simulação, fez-se necessário a utilização de um bloco que fizesse referência a inércia do eixo.

Como descrito anteriormente, tem-se um bloco que contém a entrada de sinal analógico. Um bloco de soma é responsável por calcular a diferença entre a velocidade desejada e a velocidade medida no eixo, para então enviar esse sinal ao controlador PI que ajustará a tensão de entrada no inversor trifásico, assim controlando a velocidade do motor. Existe também um *scope*, responsável por mostrar em tempo real a correção de velocidade sendo realizada pelo controlador, assim como a alteração da tensão da fonte.

RESULTADOS E DISCUSSÃO

Em todas as atividades realizadas, o resultado que se obteve foi o esperado. Após a instalação dos pacotes adicionais, todas as outras atividades necessitavam de um mesmo processo para funcionar, que são:

- i- Selecionar o modelo da placa (porta na qual ela estava conectada), a construção do modelo no Simulink (MATLAB®),
- ii- Selecionar o modo de reprodução, que nesse caso é o externo, já que o intuito é fazer com que o programa tenha comunicação com o *hardware* externo; e depois,
- iii- Selecionar “*build, deploy and start*” responsável por adaptar, carregar e reproduzir o programa criado no Arduino.

Os resultados representaram sucesso na idealização e implementação do projeto, uma vez que foi possível observar em tempo real as correções realizadas pelo controlador, independentemente de quantas vezes a velocidade desejada fosse alterada, a velocidade medida no eixo do motor sempre tendia a se igualar a esta. Como pode ser observado na figura 7.

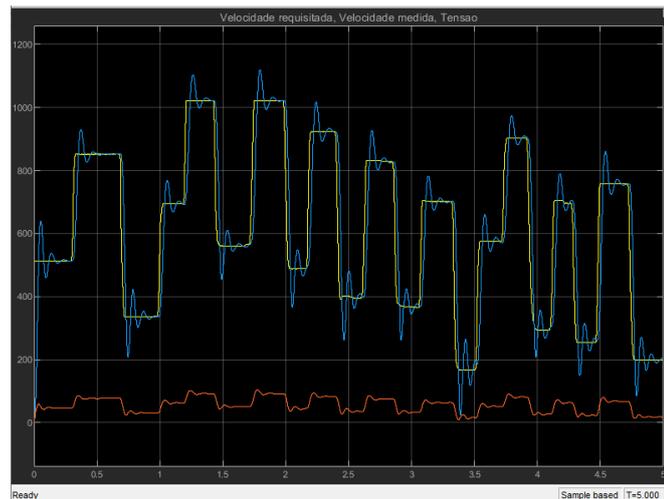


FIGURA 7. Velocidades e tensão. Fonte: Próprio autor.

O sinal em cor vermelha representa a tensão que a fonte fornece, o sinal amarelo representa o sinal inserido por meio do Arduino, referente a velocidade desejada e por fim, o sinal azul representa a velocidade medida no eixo do motor. Pode-se observar que ao longo do tempo, esse sinal sofre correções até que se aproxima do sinal amarelo.

CONCLUSÕES

Em função da pandemia da COVID-19 e das práticas de distanciamento social adotadas, a utilização das instalações e laboratórios da unidade de ensino não estão disponíveis, não sendo possível utilizar um motor BLDC real, por isso a implementação de um simulador. Todavia, os resultados buscados foram alcançados conforme ilustrado nos resultados.

Como parte final, resta testar o controle de motores e a leitura de dados de sensores, para então adaptar o programa de simulação de um motor para se comunicar com esse hardware, finalizando assim o sistema *hardware in the loop*.

AGRADECIMENTOS

Agradeço ao IFSP pelo suporte financeiro a partir da Bolsa de Iniciação Científica PIBIFSP.

REFERÊNCIAS

CHAPMAN, Stephen J. Fundamentos de máquinas elétricas/ tradução: Anatólio Laschuk. – 5. ed. Porto Alegre: AMGH, 2013.

LOUSADA, Ricardo. O que é Arduino: Para que serve, vantagens e como utilizar. Eletrogate, 2020. Disponível em: <<https://blog.eletrogate.com/o-que-e-arduino-para-que-serve-vantagens-e-como-utilizar/>> Acesso em: 30 de set. de 2021.

MATLAB. (2019). versão 9.7.0 (R2019b). Natick, Massachusetts: The MathWorks Inc.

OGATA, K. Engenharia de controle moderno. 5.ed. São Paulo: Pearson Prentice Hall, 2009.

RIPPEL, Wally. Induction Versus Brushless Motors. Tesla, 2007. Disponível em: <<https://www.tesla.com/blog/induction-versus-dc-brushless-motors>> Acesso em: 30 de set. de 2021.