

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Juliane Ritzel Farret

**PLATAFORMA DIDÁTICA HARDWARE-IN-THE-LOOP PARA
EMULAÇÃO DE SISTEMAS LINEARES**

Santa Maria, RS
2022

Juliane Ritzel Farret

PLATAFORMA DIDÁTICA HARDWARE-IN-THE-LOOP PARA EMULAÇÃO DE SISTEMAS LINEARES

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Engenharia Elétrica**.

ORIENTADOR: Prof. Lucas Vizzotto Bellinaso

COORIENTADOR: Prof. Rodrigo Varella Tambara

Santa Maria, RS
2022

©2022

Todos os direitos autorais reservados a Juliane Ritzel Farret. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

End. Eletr.: juliane.farret@acad.ufsm.br

Juliane Ritzel Farret

PLATAFORMA DIDÁTICA HARDWARE-IN-THE-LOOP PARA EMULAÇÃO DE SISTEMAS LINEARES

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Engenharia Elétrica**.

Aprovado em 16 de agosto de 2022:

Lucas Vizzotto Bellinaso, Dr. (UFSM)
(Presidente/Orientador)

Rodrigo Varella Tambara, Dr. (UFSM)
(Coorientador)

Andrei Da Cunha Lima, Eng. (UFSM)

Santa Maria, RS
2022

AGRADECIMENTOS

À minha família pelo amor incondicional, apoio e suporte durante toda minha vida e anos de faculdade.

Ao meu orientador, professor Dr. Lucas Bellinaso, pela paciência e conhecimento com o qual me auxiliou durante todas as etapas do trabalho e quem sempre me motivou a continuar.

Ao meu coorientador, professor Dr. Rodrigo Varella, por aceitar me orientar, pelo tempo, dedicação e paciência destinados à minha orientação.

Ao Émerson pela produção da aplicação em Python, o qual foi fundamental para o desenvolvimento desse trabalho.

Aos guris do INRI/LPEE e do GSEC pela companhia e ajuda durante os testes práticos do projeto.

*"You may have to fight a battle more than
once to win it."*

(Margaret Thatcher)

RESUMO

PLATAFORMA DIDÁTICA HARDWARE-IN-THE-LOOP PARA EMULAÇÃO DE SISTEMAS LINEARES

AUTORA: Juliane Ritzel Farret

ORIENTADOR: Lucas Vizzotto Bellinaso

COORIENTADOR: Rodrigo Varella Tambara

No campo das engenharias e ciências exatas, há uma demanda constante por verificação experimental de estratégias de controle, esquemas de modulação, entre outros aspectos. Projetos e testes de controle de processos são tarefas difíceis que podem desencorajar os alunos a trabalhar na área devido a curva de aprendizado muito acentuada. As abordagens modernas para a educação em sistemas de controle usam simulações e experimentos para facilitar o processo de aprendizado. Porém, ambientes, como laboratórios universitários, não possuem os altos recursos financeiros necessários para comprar essas soluções. A necessidade versus a dificuldade de aquisição motivou a criação de uma plataforma HIL de baixo custo para simulação de sistemas lineares genéricos capaz de desempenhar a mesma ou pelo menos as principais funções dos equivalentes comerciais. Neste trabalho é apresentado os estágios de projeto, simulação e testes experimentais do *hardware* e do *software* implementado em microcontrolador, para a implementação de sistemas de até segunda ordem. Assim como, são exploradas ainda as etapas de montagem final do produto eletrônico desenvolvido e a validação experimental junto a um computador dos modelos matemáticos simulados em tempo discreto e envio dos dados ao sistema a ser simulado. Por fim, os resultados experimentais são cuidadosamente discutidos.

Palavras-chave: Sistemas Lineares. Equações Diferenciais. Sistemas de Controle. *Hardware-in-the-Loop*. Simulação em tempo real. Microcontrolador.

ABSTRACT

DIDACTIC HARDWARE-IN-THE-LOOP PLATFORM FOR LINEAR SYSTEMS EMULATION

AUTHOR: Juliane Ritzel Farret

ADVISOR: Lucas Vizzotto Bellinaso

CO-ADVISOR: Rodrigo Varella Tambara

In the field of engineering and exact sciences, there is a constant demand for experimental verification of control strategies, modulation schemes, among other aspects. Process control projects and tests are difficult tasks that can discourage students from working in the field due to the steep learning curve. Modern approaches to control systems education use simulations and experiments to facilitate the learning process. However, environments such as university labs do not have the high financial resources needed to purchase these solutions. The need versus the difficulty of acquisition motivated the creation of a low-cost HIL platform for simulating generic linear systems capable of performing the same or at least the main functions of commercial equivalents. This work presents the stages of design, simulation, and experimental tests of hardware and software implemented in microcontrollers, for the generic implementation of up to second-order systems. Likewise, the final assembly steps of the electronic product developed and the experimental validation with a computer of the mathematical models simulated in discrete-time and sending the data to the system to be simulated are also explored. After demonstrating the validations, the results obtained are discussed.

Keywords: Linear systems. Differential Equations. Control systems. Hardware-in-the-Loop. Real Time Simulation. Microcontroller

LISTA DE FIGURAS

Figura 2.1 – Esquema básico de um HIL	11
Figura 3.1 – Diagrama de pinagem STM32	14
Figura 3.2 – Conversor DA com saídas digitais	15
Figura 3.3 – Circuito saídas digitais	16
Figura 3.4 – Circuito total simulado no software PSIM	17
Figura 3.5 – Circuito fonte simétrica $\pm 12V$	17
Figura 3.6 – Placa projetada	18
Figura 3.7 – Placa finalizada após soldagem dos componentes para teste	18
Figura 3.8 – Projeto interface de interação do usuário com o produto	19
Figura 3.9 – Dimensões da caixa utilizada	19
Figura 3.10 – Interface usuário aplicativo Python	21
Figura 3.11 – Entradas e saídas do sistema emulados	22
Figura 3.12 – Máquina de estados dos modos de operação do sistema	23
Figura 4.1 – Simulação sistema de primeira ordem com $f_{sw} = 100\text{ Hz}$, 1 kHz e 10 kHz ...	24
Figura 4.2 – Resultado experimental sistema de primeira ordem; $f_{sw} = 1\text{ kHz}$; $f_{sinal} = 10\text{ rad/s}$	25
Figura 4.3 – Resposta ao degrau (Matlab) para sistema G_1 ; $f_{sw} = 1\text{ kHz}$;	25
Figura 4.4 – Resultado experimental sistema de primeira ordem; $f_{sw} = 10\text{ kHz}$; $f_{sinal} = 4\text{ rad/s}$	26
Figura 4.5 – Resposta ao degrau (Matlab) para sistema G_1 ; $f_{sw} = 10\text{ kHz}$;	26
Figura 4.6 – Simulação sistema G_2 de segunda ordem com $f_{sw} = 100\text{ Hz}$, 1 kHz e 10 kHz	27
Figura 4.7 – Resultado experimental sistema G_2 ; $f_{sw} = 100\text{ Hz}$; $f_{sinal} = 40\text{ rad/s}$	27
Figura 4.8 – Resposta ao degrau (Matlab) para sistema G_2 ; $f_{sw} = 100\text{ Hz}$;	28
Figura 4.9 – Resultado experimental sistema G_2 ; $f_{sw} = 1\text{ kHz}$; $f_{sinal} = 40\text{ rad/s}$	28
Figura 4.10 – Resposta ao degrau (Matlab) para sistema G_2 ; $f_{sw} = 1\text{ kHz}$;	28
Figura 4.11 – Resultado experimental sistema G_2 ; $f_{sw} = 10\text{ kHz}$; $f_{sinal} = 20\text{ rad/s}$	29
Figura 4.12 – Resposta ao degrau (Matlab) para sistema G_2 ; $f_{sw} = 10\text{ kHz}$;	29
Figura 4.13 – Simulação sistema G_3 de segunda ordem com $f_{sw} = 100\text{ Hz}$, 1 kHz e 10 kHz	30
Figura 4.14 – Resultado experimental sistema G_3 ; $f_{sw} = 100\text{ Hz}$; $f_{sinal} = 40\text{ rad/s}$	30
Figura 4.15 – Resposta ao degrau (Matlab) para sistema G_3 ; $f_{sw} = 100\text{ Hz}$;	30
Figura 4.16 – Resultado experimental sistema G_2 ; $f_{sw} = 1\text{ kHz}$; $f_{sinal} = 40\text{ rad/s}$	31
Figura 4.17 – Resposta ao degrau (Matlab) para sistema G_3 ; $f_{sw} = 1\text{ kHz}$;	31
Figura 4.18 – Resultado experimental sistema G_3 ; $f_{sw} = 10\text{ kHz}$; $f_{sinal} = 40\text{ rad/s}$	31
Figura 4.19 – Resposta ao degrau (Matlab) para sistema G_3 ; $f_{sw} = 10\text{ kHz}$;	32

SUMÁRIO

1	INTRODUÇÃO	9
1.1	OBJETIVOS	10
1.2	ORGANIZAÇÃO DO TRABALHO	10
2	REVISÃO BIBLIOGRÁFICA	11
3	METODOLOGIA	13
3.1	FERRAMENTAS UTILIZADAS	13
3.1.1	STMicroelectronics SMT32F103C8T6 e ST-Link V2	13
3.1.2	Keil uVision e STM32CubeMX	14
3.1.3	Google Colaboratory	14
3.2	PROJETO E DESENVOLVIMENTO DO HARDWARE	15
3.2.1	Concepção do circuito total	15
3.2.2	Sistema de alimentação	16
3.2.3	Placa de circuito impresso	17
3.2.4	Produto final e interface do usuário	19
3.3	DESENVOLVIMENTO DO SOFTWARE	20
3.3.1	Propriedades do sistema	20
3.3.2	Modos de operação	21
4	RESULTADOS E DISCUSSÕES	24
4.1	SISTEMA DE PRIMEIRA ORDEM	24
4.2	SISTEMA DE SEGUNDA ORDEM	26
5	CONCLUSÃO	33
	REFERÊNCIAS BIBLIOGRÁFICAS	34
	APÊNDICE A – MATERIAIS UTILIZADOS NO PROJETO DE HARDWARE	36
	APÊNDICE B – CÓDIGO MATLAB PROJETO CIRCUITOS	37

1 INTRODUÇÃO

Sistemas de controle e processamento de sinais estão sendo cada vez mais utilizados em muitas aplicações, incluindo propulsão veicular, mecânica de voo, controle de aeronaves e distribuição de energia. O projeto desses sistemas requer um conhecimento significativo em várias áreas da engenharia elétrica e de computação. Isso inclui o entendimento de eletrônica, teoria de controle, teoria de sistemas lineares e não lineares, e eletromagnéticos. Além disso, o conhecimento do comportamento de máquinas elétricas, técnicas de modelagem e microprocessadores para implementação de algoritmos de controle deve ser estabelecido para criar projetos eficientes (WILLIAMS et al., 2004). Para muitos estudantes, um laboratório ou plataforma de ensino pode fornecer uma experiência inicial em síntese, exigindo que eles usem o conhecimento de todo o currículo com grande atenção aos detalhes (BALOG et al., 2005). Além disso, um conhecimento prático das ferramentas de simulação fornece suporte para síntese e avaliação criativa de circuitos. Um simulador é uma maneira poderosa de testar novas idéias sem as dificuldades associadas à fabricação de circuitos (TORREY, 1994). Da mesma forma, o estudo de sistemas lineares requer demonstração prática para que os alunos aprendam intuitivamente conceitos como dinâmica e estabilidade de sistemas. Isso sugere que as ferramentas didáticas usadas em laboratório têm grande valor como componente de um currículo de engenharia eletrônica e de computação.

A experiência mostra que sistemas de controle, da mesma maneira que, sinais e sistemas, como disciplina, não é facilmente assimilada pelos alunos devido às diferentes áreas envolvidas. Experimentos de laboratório podem ajudar os alunos a assimilar os conceitos teóricos de assuntos muito complexos e fazer um primeiro contato com a instrumentação e equipamentos com os quais os alunos trabalharão no futuro (JIMÉNEZ-MARTÍNEZ et al., 2005; TORREY, 1994). No entanto, experimentos de laboratório, embora sejam a melhor estratégia, apresentam problemas de tempo, dinheiro e segurança. O uso de microprocessadores e microcontroladores para implementar plataformas de simulação em tempo real estão presentes na literatura como opções alternativas quando existem impedimentos nos testes sobre os sistemas reais, devido aos custos, riscos ou necessidade de agilidade nos testes. Contudo, estas soluções são direcionadas a casos de estudos particulares onde é desejado simular apenas um componente de um sistema, como por exemplo máquinas indutivas (TAVANA; DINAHAHI, 2016). Quando se faz necessário um sistema de emulação para propósito geral, soluções comerciais, como as plataformas Typhoon HIL e dSpace, podem ser extremamente caras, o que pode ser impeditivo caso o estudo não justifique os custos com esses equipamentos.

Neste trabalho, é proposta uma arquitetura para operações vetoriais e matriciais com foco didático na emulação de sistemas lineares baseada na representação em espaço de estados e que seja simples o suficiente para ser implementada com o uso de microcontroladores e plataformas menos sofisticadas, para que o custo seja menor que as soluções comerciais.

1.1 OBJETIVOS

O objetivo geral do presente trabalho é a concepção de um hardware emulador genérico de baixo custo para implementação de sistemas de até ordem 4, para ensino e aprendizagem na área de engenharias e ciências exatas, bem como replicar esses sistemas para utilização nos laboratórios do Centro de Tecnologia.

Este objetivo geral se desmembra em 5 objetivos específicos:

- Projetar circuitos eletrônicos e confecção da placa de circuito impresso;
- Realizar a integração de componentes eletrônicos, *hardware* e *software*;
- Desenvolver algoritmo para realizar a simulação dos modelos matemáticos em tempo discreto;
- Desenvolver comunicação do computador com o microcontrolador para envio dos dados do sistema a ser simulado;
- Realizar simulações para validar o *software* e *hardware* desenvolvidos;

1.2 ORGANIZAÇÃO DO TRABALHO

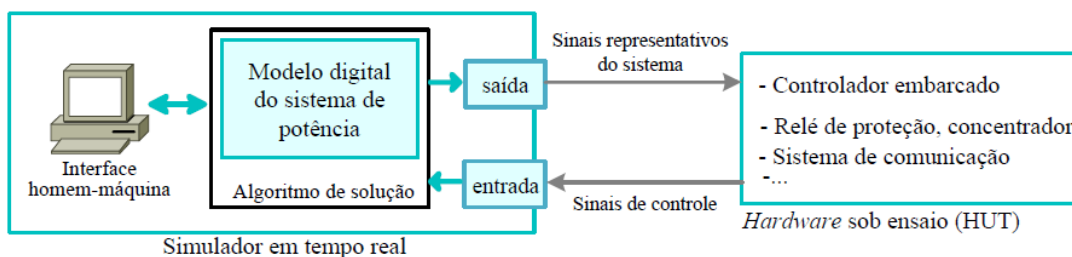
O trabalho é dividido em 5 capítulos. No presente capítulo são apresentados a contextualização, motivação, e os objetivos do trabalho. No capítulo 2 é apresentada uma revisão bibliográfica, a fim de situar o leitor, introduzindo conceitos como *hardware-in-the-loop*, e emulação em tempo real de sistemas lineares. No capítulo 3 são descritas as principais ferramentas utilizadas no desenvolvimento deste trabalho e apresenta-se uma visão geral do projeto, na qual o leitor será capaz de compreender o sistema que será discutido nos capítulos seguintes. De mais, será abordado o progresso do projeto, especificando os procedimentos e testes realizados. Já no capítulo 4 apresentam-se os resultados obtidos a partir da metodologia utilizada. Por fim, no capítulo 5 o trabalho é concluído, expondo os principais pontos observados durante todo o desenvolvimento do mesmo.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo é apresentada uma breve contextualização sobre dispositivos digitais para emulação de sistemas lineares programáveis. Bem como alguns trabalhos relevantes na área de HIL comentados com os principais conceitos envolvidos neste projeto.

A técnica *hardware-in-the-loop* (HIL) para simulações possui um longo histórico, com as mais diversificadas aplicações. Como método de simulação, o HIL se refere a uma condição onde partes da simulação digital em tempo real são substituídas por componentes físicos reais. A metodologia procede com um dispositivo sob teste ou *hardware-under-test* (HuT) conectado através de interfaces de entrada e saída, ou seja, conversores digitais analógico e/ou analógico-digital, à plataforma de simulação (AL, 2015). É uma ferramenta útil para situações onde o dispositivo a ser testado é difícil de ser modelado em programas de simulação ou é necessário testar o equipamento diante de condições específicas (SPARN DHEEPAK KRISHNA-MURTHY; RUTH; WU, 2018). As aplicações do HIL já ocorreram nas indústrias aeroespaciais e automotivas (BACIC, 2005) e atualmente encontram nas redes elétricas inteligentes um amplo campo de aplicação. A Figura 2.1 representa o funcionamento da técnica HIL. Através de uma simulação, sinais de entrada analógicos/digitais são enviados para o HuT. O equipamento ao fazer a leitura desses sinais, baseado em suas configurações, envia respostas para a simulação. (BEZERRA, 2019)

Figura 2.1 – Esquema básico de um HIL



Fonte: (RUEDA, 2019)

Em (SARHADI; YOUSEFPOUR, 2014) é discutido a importância e aplicações das técnicas HIL. Além disso, é mostrado em quais fases do projeto de um sistema ele pode ser aplicado, indo desde a construção e implementação, integração e teste em campo do sistema. A confiabilidade, eficiência e chance de sucesso do sistema é aumentada, pois erros são identificados e mitigados ainda na fase de implementação do *software* e *hardware*. É concluído que os testes HIL são especialmente importantes para testes destrutivos ou de alto risco, testar elementos com dinâmica altamente não-linear e testar leis de controle complexas observando as interações entre *hardware* e *software*. (SILVA, 2017)

A arquitetura HIL é apresentada para testes de sistemas de controle pela empresa Nati-

onal Instruments (NI) como uma técnica poderosa e eficiente que traz segurança, custos reduzidos e torna viável testes de sistemas críticos (National Instruments, 2022). É exemplificado um sistema HIL composto por basicamente 3 componentes: um microcontrolador de tempo real, interfaces I/O (*Input/Output*) e uma interface para o operador. O microcontrolador é tido como o núcleo do sistema HIL pois executa as tarefas em tempo determinístico, garantindo a fidelidade do sistema simulado. Empresas como a brasileira EMBRAER utilizam o sistema *Hardware-in-the-loop* da NI para teste e validação de leis de controle, sistemas aviônicos e sistema elétrico de alguns de seus aviões. O HIL é composto por 21 sistemas *PCIeExtensions for Instrumentation* (PXI), utilizando memória refletiva (*Reflective Memory*) e se comunicando entre si via Ethernet.

Em (BITTAR; FIGUEIREDO, 2013) é apresentada uma plataforma para simulação e testes de um VANT pequeno no simulador X-Plane, utilizando HIL. Testes de leis de controle de altitude são apresentados mostrando sua atuação na presença de rajadas de vento e turbulência. A interface HIL é composta pelo X-Plane enviando dados via serial para um microcontrolador ARM Cortex-M3 que possui leis de controle programadas. Outro microcontrolador de mesmo modelo foi utilizado para receber os dados de controle de solo (GCS). A comunicação é realizada com o X-Plane enviando dados para um programa de interface, o qual retransmite os dados via USB-Serial para o microcontrolador (não é comentado qual a taxa de transmissão de dados). Existe o problema de adicionar o protocolo USB no meio do caminho, por conta da interface, possivelmente limitando a taxa de transmissão e o funcionamento em tempo-real.

3 METODOLOGIA

Este capítulo contempla todo o procedimento de projeto e desenvolvimento do emulador proposto. O sistema como um todo é composto por

- Uma placa de circuito impresso, responsável por integrar com o microcontrolador os circuitos de alimentação, proteção, amplificação de sinal e conversor DA, que gera a forma de onda das saídas analógicas. Assim como, inclui os dispositivos de LEDs e botões no sistema.
- Um dispositivo microcontrolador para receber os parâmetros do computador, através de um protocolo USB, e configurar os modos de operação.
- Um computador PC equipado com ferramentas de software para configuração, processamento dos parâmetro e posterior teste do sistema.

A primeira seção a seguir apresenta cada uma das ferramentas utilizadas. A seguir, define-se funcionalmente a plataforma emuladora, com base no comportamento esperado, com a apresentação dos esquemáticos dos circuitos desenvolvidos, projeto da placa de circuito impresso e dos requisitos para o *hardware* final. Por fim, a subseção 3.3 descreve o desenvolvimento do *firmware* para troca dos modos de operação e protocolo para envio dos sistemas do PC para a placa.

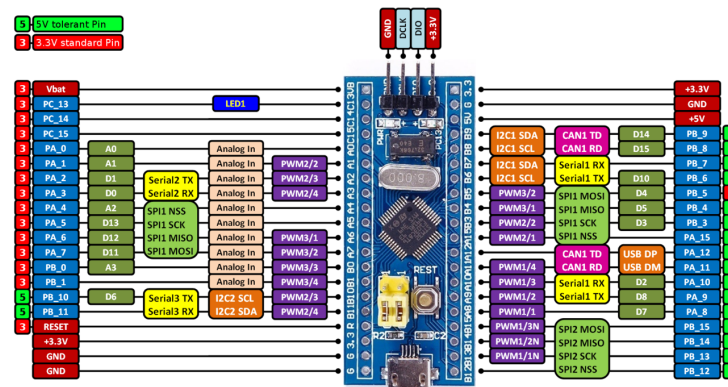
3.1 FERRAMENTAS UTILIZADAS

Esta seção apresenta a série de ferramentas utilizadas para o desenvolvimento deste projeto. As duas primeiras partes contêm as características mais relevantes da placa de desenvolvimento utilizada e ferramentas de *software* utilizadas.

3.1.1 STMicroelectronics SMT32F103C8T6 e ST-Link V2

A placa de desenvolvimento *STM32F103C8T6* é baseada na família de linhas de desempenho de média densidade *STM32F103xx* da *STMicroelectronics*, com core ARM Cortex-M3 (*STMicroelectronics*, 2022a). A placa é utilizada para receber os parâmetros e o sistema do computador através da emulação de uma porta serial comum sobre a interface USB. Do *hardware* da placa, destaca-se a utilização dos pinos de entrada e saída de uso geral, memória FLASH e conversores AD.

Figura 3.1 – Diagrama de pinagem STM32



Fonte: (RANDRIAKOTONJANAHARY, 2019)

O *ST-LINK V2* é um gravador compatível com a versão oficial disponibilizada pelo fabricante permitindo atualização através de link direto com a ST. Permite fazer upload, gravar e depurar programas em microcontroladores da família STM8 e STM32, com alto desempenho e custo benefício.

3.1.2 Keil uVision e STM32CubeMX

Keil uVision é uma das plataformas de desenvolvimento da ST. Para o microcontrolador da placa de desenvolvimento utilizada, a STM32F103, foi utilizada a versão 5.37 do software, que é a mais recente que a suporta até a data do trabalho. Juntamente com o *Keil uVision*, o fabricante disponibiliza o *CubeMX*, que é uma ferramenta gráfica que permite a configuração dos microcontroladores e microprocessadores STM32, bem como a geração do código C de inicialização correspondente para o núcleo Arm Cortex-M, por meio de um processo de configurações. (STMicroelectronics, 2022b)

3.1.3 Google Colaboratory

O Colab, ou "Colaboratory", permite escrever e executar Python no navegador sem a necessidade de configurações, acesso gratuito e fácil compartilhamento. Nesse trabalho foi utilizado para o processamento inicial da separação dos sistemas lineares para o envio ao microcontrolador. Assim, o uso da memória do microcontrolador é menor e consequentemente o processamento do sistema mais rápido.

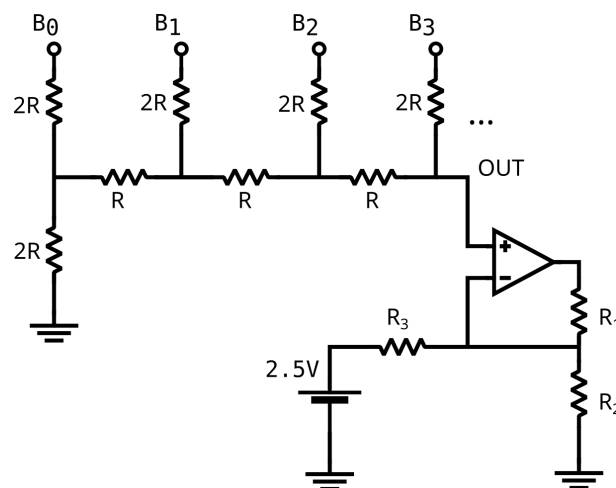
3.2 PROJETO E DESENVOLVIMENTO DO HARDWARE

O *hardware* do sistema é dividido principalmente entre os circuitos de entrada e saída analógica e sua relação com o microcontrolador. O objetivo desta seção é apresentar cada fase do projeto desses circuitos eletrônicos e descrever o seu funcionamento e interações com os demais periféricos.

3.2.1 Concepção do circuito total

Primeiramente, devido à falta de conversores DA na placa escolhida para o processamento dos sistemas emulados, foi necessário o projeto de uma rede R/2R para converter informação digital em saída analógica. Redes de resistores em escada provêm um simples e barato modo para fazer uma conversão digital para analógica (DAC). A escada R/2R foi escolhida nesse trabalho devido sua inerente exatidão e o fácil processo de construção.

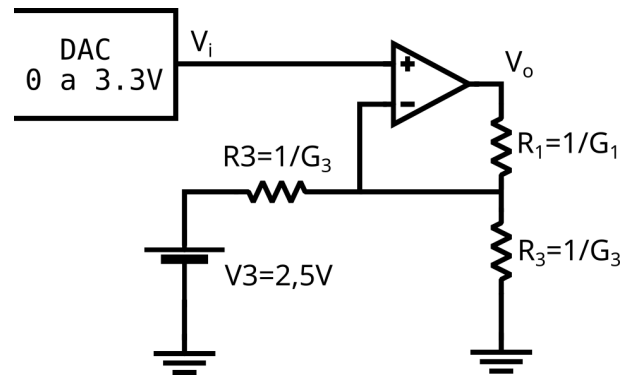
Figura 3.2 – Conversor DA com saídas digitais



Fonte: Autoria própria.

Informações digitais são apresentadas para a escada como bits individuais de uma palavra digital comutada entre uma tensão de referência (V_r) e a terra (ground). Dependendo do número e localização dos bits comutados para V_r ou ground, V_{out} irá variar entre 0 volts e V_r , dependendo onde são localizadas as conexões das entradas. Estas entradas variam do bit mais significativo (MSB - *Most Significant Bit*) para o bit menos significativo (LSB - *Least Significant Bit*). Como os nomes indicam, o MSB, quando ativado, causa a maior modificação na tensão de saída e o LSB, quando ativado, irá causar a menor modificação. No projeto apresentado a tensão varia de 0 a 3.3V para duas saídas digitais de 10 bits cada. A cascata de resistores, com a variação de tensão descrita, pode ser vista na figura 3.4 da simulação do circuito completo.

Figura 3.3 – Circuito saídas digitais



Fonte: Autoria própria.

Para o projeto dos circuitos de sinal de entrada e saída utilizou-se o amplificador operacional TL064 com os valores de resistores calculados a partir das leis das correntes de Kirchhoff. O desenvolvimento dos cálculos pode ser visto abaixo a partir da equação 3.1. Um algoritmo em Matlab foi criado para validar esses valores e pode ser encontrado no apêndice B do trabalho.

LCK:

$$\left(\frac{V_o - V_i}{R_1}\right) + \left(\frac{0 - V_i}{R_2}\right) + \left(\frac{V_3 - V_i}{R_3}\right) = 0 \quad (3.1)$$

$$\begin{bmatrix} V_o - V_i & -V_i \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \frac{1}{R_3} \begin{bmatrix} V_i - V_3 \end{bmatrix} \quad (3.2)$$

Caso 1: $V_i = 1.65 \text{ V}; V_o = 0 \text{ V}$

Caso 2: $V_i = 0 \text{ V}; V_o = -10 \text{ V}$

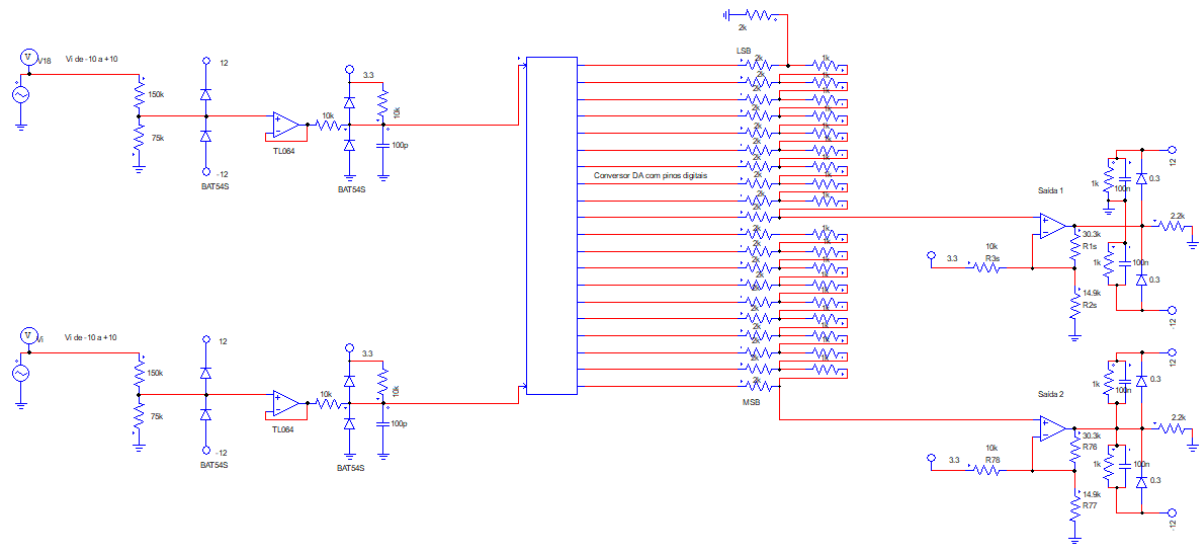
$$\begin{bmatrix} -1.65 & -1.65 \\ -12 & 0 \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \frac{1}{R_3} \begin{bmatrix} 1.65 - 2.5 \\ -2.5 \end{bmatrix} \quad (3.3)$$

$$\begin{bmatrix} \frac{1}{R_1} \\ \frac{1}{R_2} \end{bmatrix} = \frac{1}{R_3} \begin{bmatrix} -1.65 & -1.65 \\ -12 & 0 \end{bmatrix}^{-1} \begin{bmatrix} -0.85 \\ -2.5 \end{bmatrix} \quad (3.4)$$

3.2.2 Sistema de alimentação

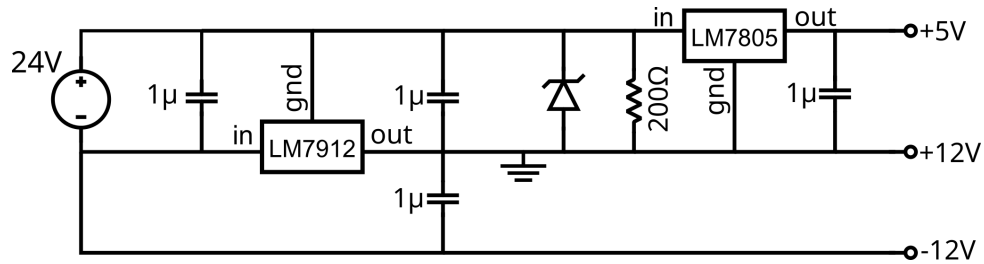
Foi projetada uma fonte simétrica de $\pm 12\text{V}$ a partir de uma fonte externa de 24V para alimentação dos amplificadores operacionais, do circuito integrado TL064, usados no projeto dos canais analógicos do emulador e na saída de $+5\text{V}$ para o microcontrolador. Para isso foram utilizados os reguladores de tensão LM7912 e LM7805, bem como capacitores e diodo zener para proteção. O circuito foi desenvolvido e simulado no software *LTspice*.

Figura 3.4 – Circuito total simulado no software PSIM



Fonte: Autoria própria.

Figura 3.5 – Circuito fonte simétrica $\pm 12V$

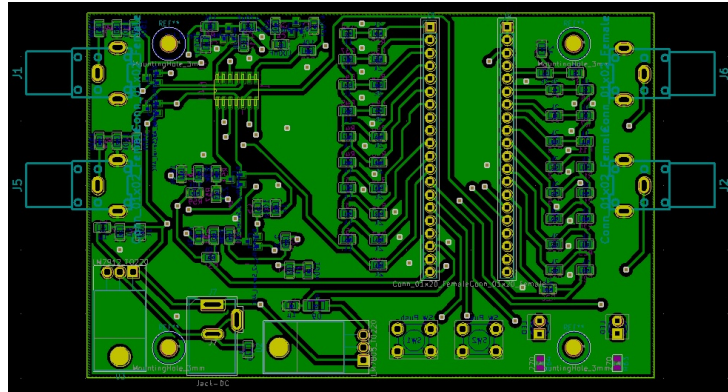


Fonte: Autoria própria.

3.2.3 Placa de circuito impresso

De forma a possibilitar a utilização em conjunto dos circuitos e integrar ao microcontrolador percebeu-se a necessidade de fabricação de uma placa de circuito impresso com um shield para acoplamento. Projetou-se a placa no software *Kicad*, com o uso de trilhas dos dois lados da placa devido ao volume de componentes e a disposição de ligação dos mesmos. O projeto pode ser visualizado na Figura 3.6.

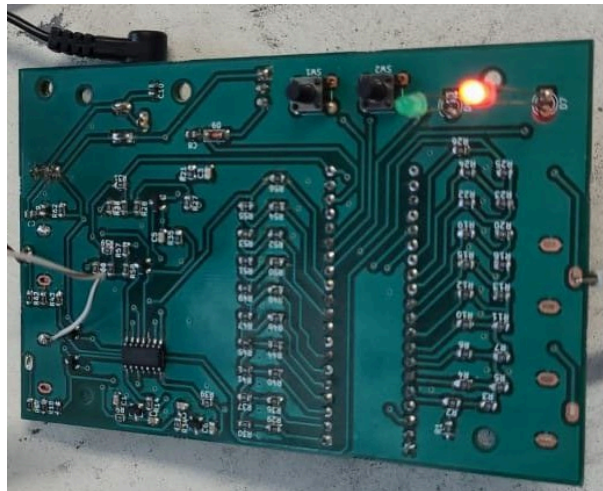
Figura 3.6 – Placa projetada



Fonte: Autoria própria.

Com o propósito de implementar o protótipo o mais próximo de um produto final, escolheu-se o uso de componentes SMD para montagem devido ao seu tamanho mais compacto. Todos componentes foram soldados com exceção dos conectores BNC devido ao seu acoplamento direto com a caixa, assim para os testes foram usados fios conectores comuns como é mostrado na figura 3.7.

Figura 3.7 – Placa finalizada após soldagem dos componentes para teste



Fonte: Autoria própria.

3.3 DESENVOLVIMENTO DO SOFTWARE

Os modos de operação são os principais módulos do sistema, tendo por objetivo enviar pra o microcontrolador o sistema linear (configuração) via USB e rodar as interrupções e equações do mesmo (emulação). Assim sendo, a finalidade desta seção é detalhar o funcionamento da máquina de estados geral do sistema, bem como a comunicação entre PC e emulador.

3.3.1 Propriedades do sistema

O estudo inicial consistiu em definir as principais características e periféricos necessários para o desenvolvimento do sistema, assim como as possíveis entradas e saídas do mesmo, os quais são descritos abaixo.

- Número de entradas
- Número de saídas
- Ordem do sistema
- Período de amostragem
- Parâmetros das matrizes
- Comunicação USB com computador
- Escrita e leitura da memória flash
- Interrupção periódica do timer

Possíveis **entradas** do usuário:

1. Função de transferência no tempo contínuo
2. Função de transferência no tempo discreto
3. Sistema no espaço de estados no tempo contínuo
4. Sistema no espaço de estados no tempo discreto

Possíveis **saídas**:

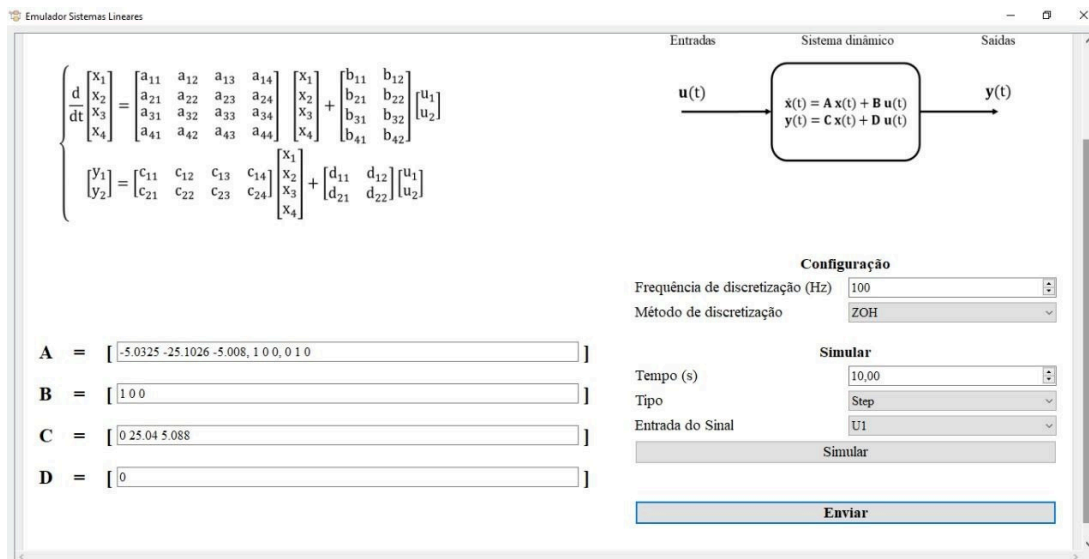
1. Sistema discreto no espaço de estados, com T_s dado, ou no melhor T_s possível para o microcontrolador, na forma de Jordan, mas com complexos conjugados agrupados (somente número real)

3.3.2 Modos de operação

Tendo como ideia processar no computador a separação do sistema para melhor envio dos parâmetros para o microcontrolador, foi criado um aplicativo em Python para simulação dos sistemas, processamento dos parâmetros e interface com o usuário. O aplicativo Python foi desenvolvido por outro aluno e possui também a função de envio dos sistemas do PC para o *hardware* através da integração com o protocolo USB em linguagem C no microcontrolador.

O funcionamento geral do sistema consiste nos modos de operação da caixa que são os modos emulação e configuração. O estado inicial da caixa é inativo, e quando essa é ligada na tomada pela fonte externa, é iniciado o modo de emulação onde se lê os valores da memória flash e começa a rodar a interrupção do timer. Uma das funções do aplicativo Python é calcular o preescaler e o contador do timer para em seguida enviar ao microcontrolador esses parâmetros via USB. São enviados junto com o preescaler e reset do timer o número de entradas, saídas, ordem do sistema e os parâmetros das matrizes. A taxa de amostragem e a função de transferência emulada são definidas pelo usuário através da interação com o aplicativo Python, o qual a interface pode ser vista na figura 3.10.

Figura 3.10 – Interface usuário aplicativo Python



Fonte: Autoria própria

O tratamento do sistema no computador serve para que os sistemas sejam arranjados a fim de simular no espaço de estados em tempo discreto. Isso pode ser feito a partir de dois algoritmos. Primeiro quando se recebe uma função de transferência em tempo contínuo e o segundo quando recebe uma função de transferência em tempo discreto.

FT contínua para SS:

1. Caso ordem relativa < 0 : adicionar polos na frequência de Nyquist para que a ordem fique igual a 0: $(s/(\pi f_s) + 1)$
2. Conferir a soma da ordem somada das duas FTs: deve ser no máximo igual a 4
3. Converter cada uma das FTs para espaço de estados contínuo

4. Obter o sistema completo através da associação dos sistemas
5. Discretizar o sistema completo no espaço de estados

FT discreta para SS:

1. Caso ordem relativa < 0: multiplicar as FT por $1/z$ (atraso) até a ordem de cada uma ser igual a 0
2. Conferir a soma da ordem somada das duas FTs: deve ser no máximo igual a 4
3. Converter cada uma das FTs para espaço de estados contínuo
4. Obter o sistema completo a partir da associação dos dois sistemas

Após processamento desses sistemas para o espaço de estados, os parâmetros são ajustados na *string* definida para envio ao microcontrolador, onde será gravada na memória flash nos endereços definidos a partir do protocolo USB. As equações abaixo demonstram a nomenclatura usada nos ponteiros utilizados do protocolo de comunicação para envio desses parâmetros. A equação 3.5 define as entradas que podem ser até duas e sistemas de quarta ordem e a equação 3.6 define as duas saídas possíveis.

$$\mathbf{x}[k+1] = \mathbf{G}\mathbf{x}[k] + \mathbf{H}\mathbf{u}[k] \quad (3.5)$$

$$\mathbf{y}[k] = \mathbf{C}\mathbf{x}[k] + \mathbf{D}\mathbf{u}[k] \quad (3.6)$$

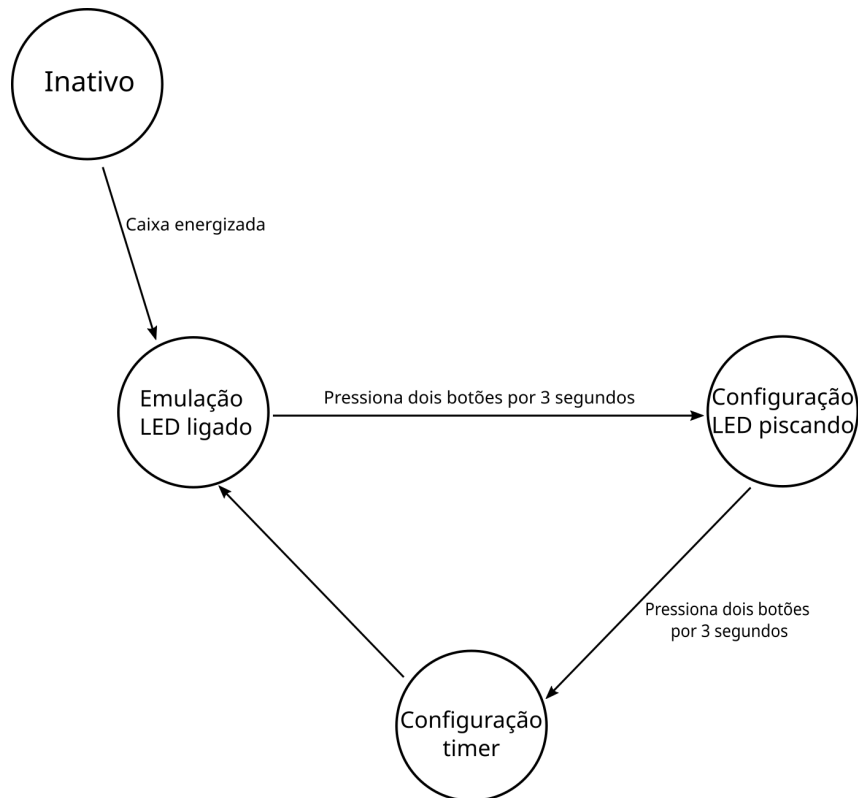
Figura 3.11 – Entradas e saídas do sistema emulados



Fonte: Autoria própria

O envio desses sistemas ocorre no modo configuração da caixa onde há apenas leitura da porta USB para o recebimento dos parâmetros e quando é registrada na memória o sistema. Além dos dois modos de operação principais da caixa, ainda existe a configuração do timer que ocorre durante a troca entre os estados principais, de configuração para emulação, apertando os dois botões simultaneamente por 3 segundos. Abaixo é possível ver a máquina de estados do funcionamento do sistema e um resumo de cada modo de operação.

Figura 3.12 – Máquina de estados dos modos de operação do sistema



Fonte: Autoria própria

Estado 1 (emulação): Começa a rodar o sistema no espaço de estados. Somente roda o sistema linear (apenas interrupção do timer e equações), mas lê os botões.

Estado 2 (configuração): Escuta a USB, recebe configurações definidas pelo usuário e registra na memória o sistema. LED piscando. (Usuário define qual FT vai simular e o valor de contagem do timer).

Estado 3 (configuração timer): Recupera *string* da memória (ler na memória flash e começa a rodar na interrupção do timer), converte para número (função `scanf`), depois atribui os valores lidos às variáveis do controlador (variáveis int).

4 RESULTADOS E DISCUSSÕES

Para a geração dos resultados, foi criado um ambiente de teste com a fonte externa ligada a placa, o gerador de sinais na entrada do sistema, o osciloscópio para a verificação e o computador para envio dos parâmetros. Simulou-se o modelo dos sistemas com o software em Python, a fim de comparar a resposta da simulação com os resultados em osciloscópio e os valores em Matlab.

Por se tratar de sistemas lentos e com baixa taxa de amostragem, foram testados nas frequências de 100 Hz, 1 kHz e 10 kHz. A seguir serão discutidos os resultados finais obtidos a partir da integração do sistema desenvolvido.

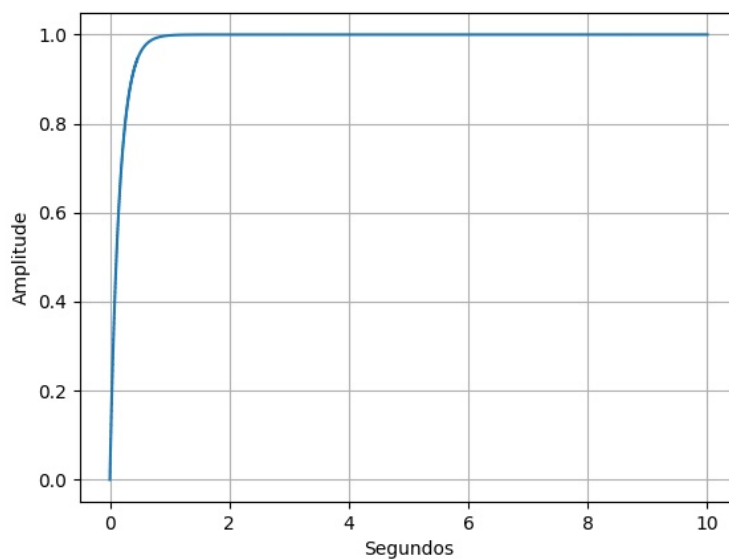
4.1 SISTEMA DE PRIMEIRA ORDEM

Primeiro foi testado um sistema com uma entrada, uma saída e foi aplicado um sinal de entrada do tipo degrau unitário. O sistema escolhido pode ser visto abaixo.

$$G_1 = \frac{1}{\frac{s}{2\pi} + 1} \quad (4.1)$$

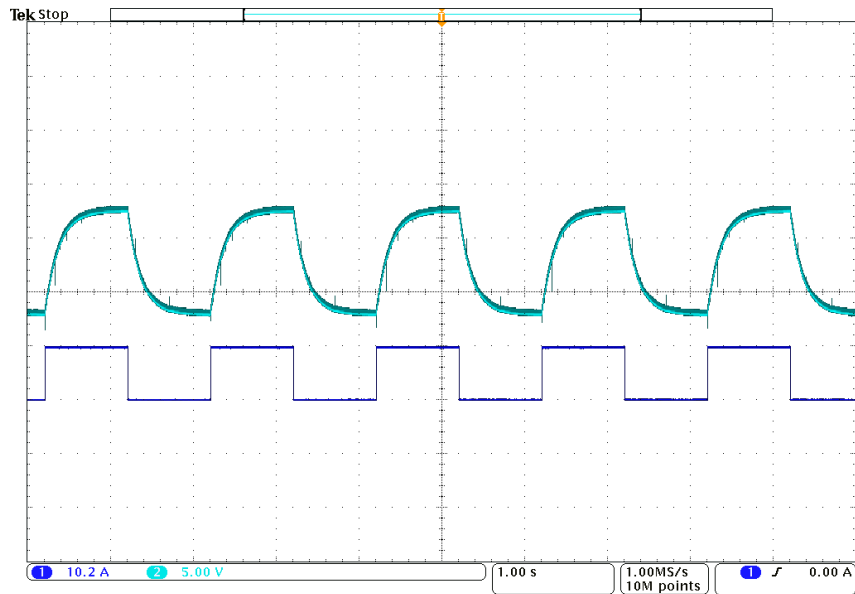
A figura 4.1 demonstra as simulações do sistema em todas as frequências escolhidas para os testes. Tendo em vista que os resultados são iguais e os gráficos estão sobrepostos. O mesmo pode ser observado nas simulações dos sistemas de segunda ordem nas figuras 4.6 e 4.13.

Figura 4.1 – Simulação sistema de primeira ordem com $f_{sw} = 100 \text{ Hz}, 1 \text{ kHz} \text{ e } 10 \text{ kHz}$



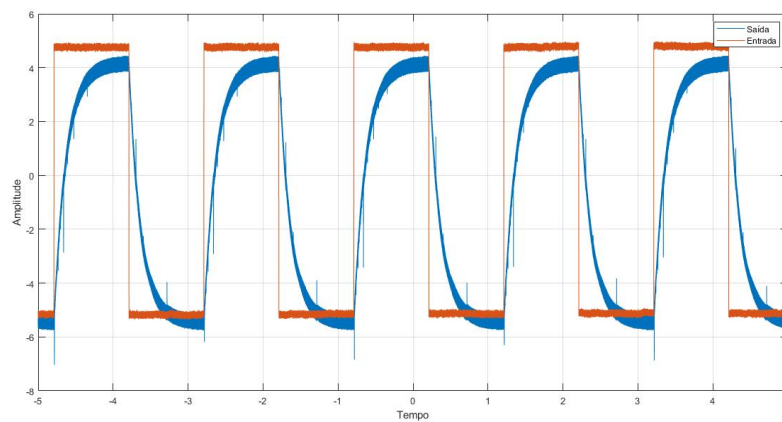
Como resultado inicial, pode-se ver o formato de onda esperado, mesmo para frequências mais elevadas comprovando assim o correto funcionamento do sistema proposto. Tendo em vista que o osciloscópio produz gráficos a partir de dados do modelo, para uma fiel representação fez-se necessário plotar os pontos obtidos a partir do arquivo csv em software Matlab. Destacando também que as escalas do osciloscópio ficaram diferentes nos dois sinais, tanto no degrau de entrada quanto na resposta do sistema. O que justifica os diferentes tamanhos dos gráficos em alguns casos.

Figura 4.2 – Resultado experimental sistema de primeira ordem; $f_{sw} = 1 \text{ kHz}$; $f_{sinal} = 10 \text{ rad/s}$



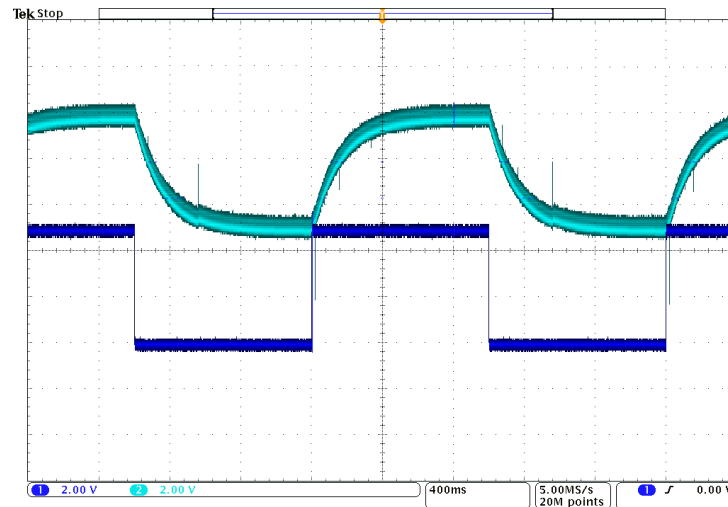
Fonte: Autoria própria

Figura 4.3 – Resposta ao degrau (Matlab) para sistema G_1 ; $f_{sw} = 1 \text{ kHz}$;



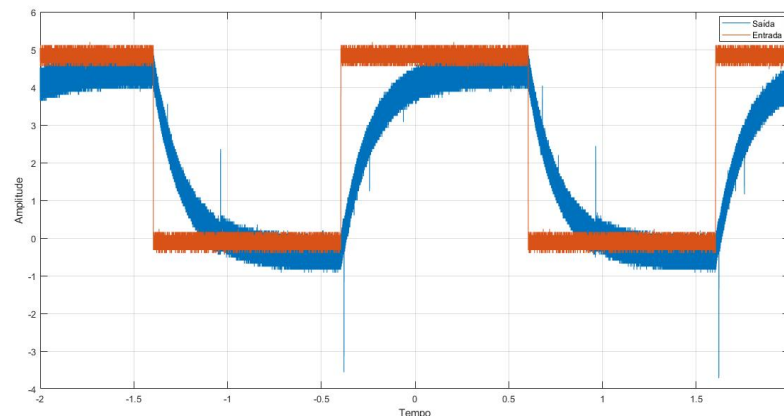
Fonte: Autoria própria

Figura 4.4 – Resultado experimental sistema de primeira ordem; $f_{sw} = 10 \text{ kHz}$; $f_{sinal} = 4 \text{ rad/s}$



Fonte: Autoria própria

Figura 4.5 – Resposta ao degrau (Matlab) para sistema G_1 ; $f_{sw} = 10 \text{ kHz}$;



Fonte: Autoria própria

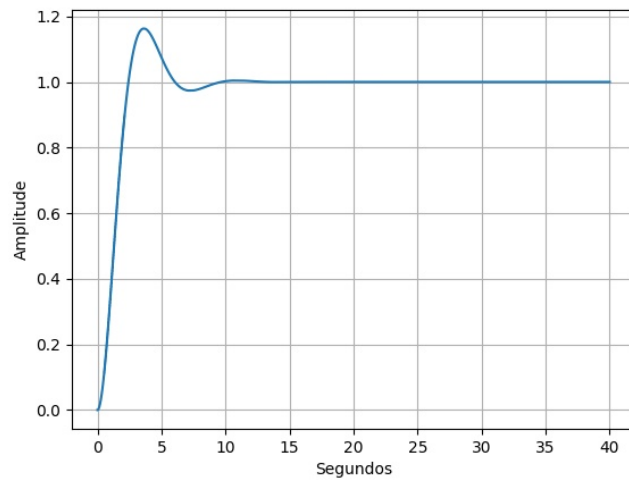
4.2 SISTEMA DE SEGUNDA ORDEM

Em seguida foram testados sistemas de segunda ordem, pois são capazes de exibir uma quantidade maior de respostas possíveis quando comparados aos sistemas de primeira ordem. Além de que, a variação de parâmetros dos sistemas de segunda ordem podem variar o tipo da resposta, em comparação com sistemas de primeira ordem em que alteração dos parâmetros altera apenas a velocidade de resposta. Os exemplos estudados nesse trabalho apresentam dois polos complexos conjugados e nenhum zero, e o teste foi realizado com entrada do tipo degrau unitário.

$$G_2 = \frac{1}{s^2 + s + 1} \quad (4.2)$$

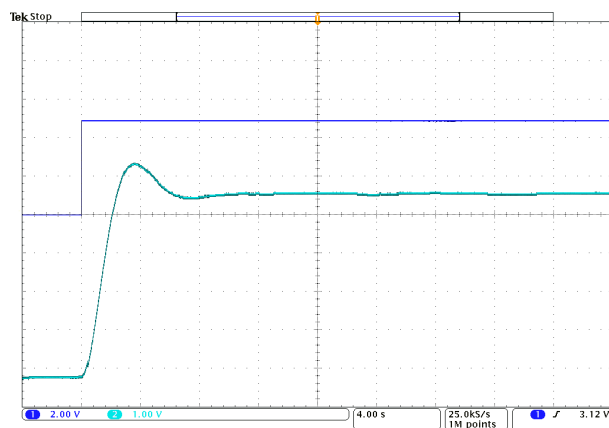
O primeiro sistema possui uma entrada, uma saída e resposta do tipo subamortecida, onde possui um polo na origem proveniente da entrada degrau unitário e dois polos complexos, provenientes do sistema.

Figura 4.6 – Simulação sistema G_2 de segunda ordem com $f_{sw} = 100 \text{ Hz}$, 1 kHz e 10 kHz



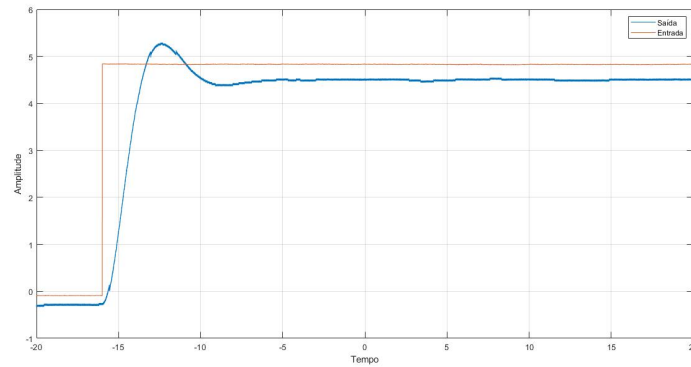
Fonte: Autoria própria

Figura 4.7 – Resultado experimental sistema G_2 ; $f_{sw} = 100 \text{ Hz}$; $f_{sinal} = 40 \text{ rad/s}$



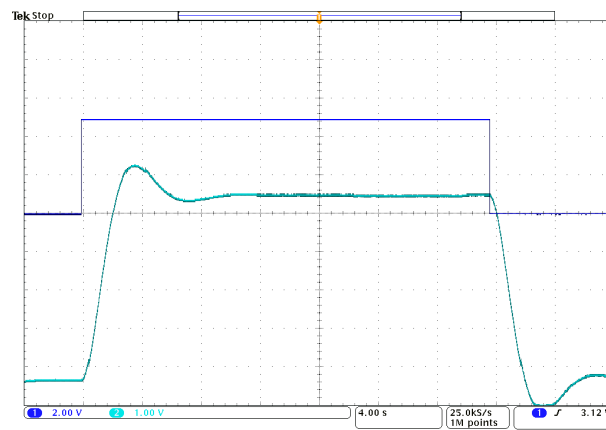
Fonte: Autoria própria

Figura 4.8 – Resposta ao degrau (Matlab) para sistema G_2 ; $f_{sw} = 100 \text{ Hz}$;



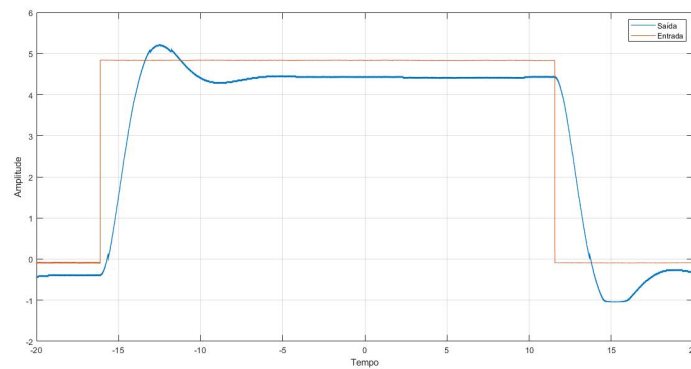
Fonte: Autoria própria

Figura 4.9 – Resultado experimental sistema G_2 ; $f_{sw} = 1 \text{ kHz}$; $f_{sinal} = 40 \text{ rad/s}$



Fonte: Autoria própria

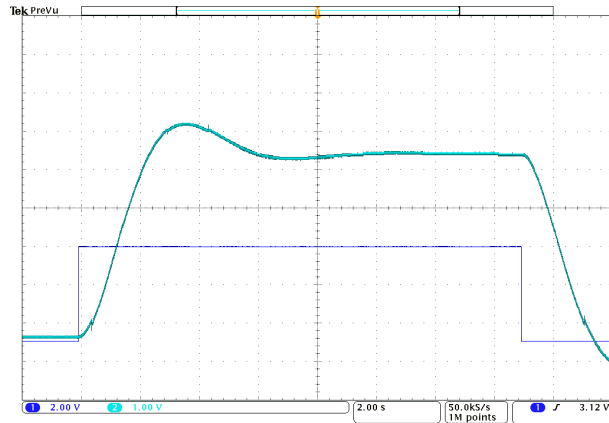
Figura 4.10 – Resposta ao degrau (Matlab) para sistema G_2 ; $f_{sw} = 1 \text{ kHz}$;



Fonte: Autoria própria

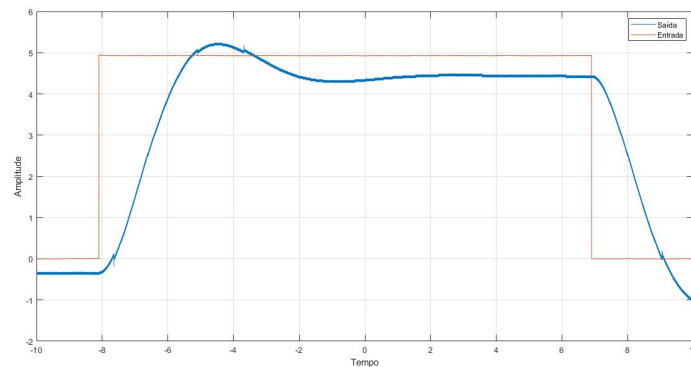
Nos resultados, demonstrados nas figuras 4.7, 4.9 e 4.11, pode-se perceber a correta geração da forma de onda do sistema subamortecido com frequência constante. Os resultados foram satisfatórios e corresponderam ao sistema proposto. Considerando que esse tipo de resposta com movimentação dos polos ao longo de uma linha radial constante produz o fator de amortecimento constante. As repostas são semelhantes exceto pela velocidade. Quanto mais afastados os polos, mais rápida será a resposta. A representação dos dados exportados em Matlab podem ser vistos nas figuras 4.8, 4.10 e 4.12 para comparação.

Figura 4.11 – Resultado experimental sistema G_2 ; $f_{sw} = 10 \text{ kHz}$; $f_{sinal} = 20 \text{ rad/s}$



Fonte: Autoria própria

Figura 4.12 – Resposta ao degrau (Matlab) para sistema G_2 ; $f_{sw} = 10 \text{ kHz}$;

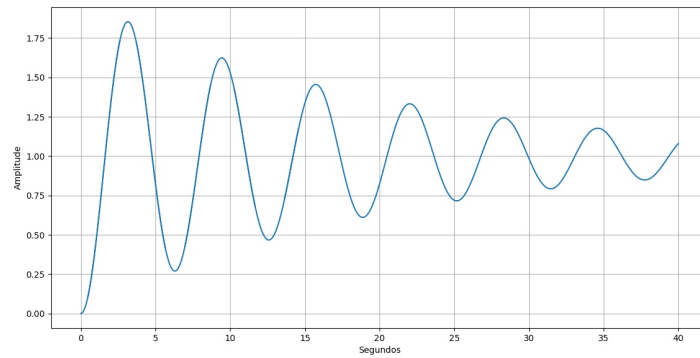


Fonte: Autoria própria

O último sistema possui uma entrada, uma saída e resposta do tipo não amortecida, onde possui um polo na origem proveniente da entrada degrau unitário e dois polos imaginários provenientes do sistema.

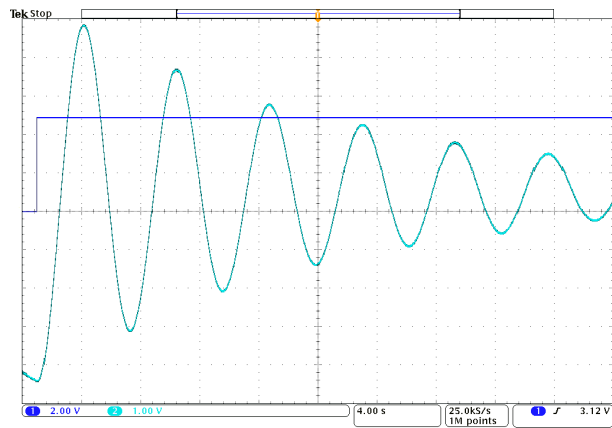
$$G_3 = \frac{1}{s^2 + 0.1s + 1} \quad (4.3)$$

Figura 4.13 – Simulação sistema G_3 de segunda ordem com $f_{sw} = 100 \text{ Hz}$, 1 kHz e 10 kHz



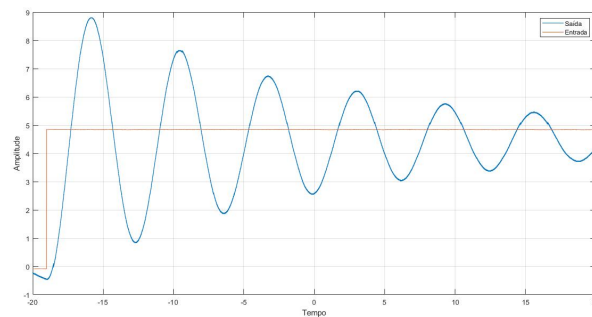
Fonte: Autoria própria

Figura 4.14 – Resultado experimental sistema G_3 ; $f_{sw} = 100 \text{ Hz}$; $f_{sinal} = 40 \text{ rad/s}$



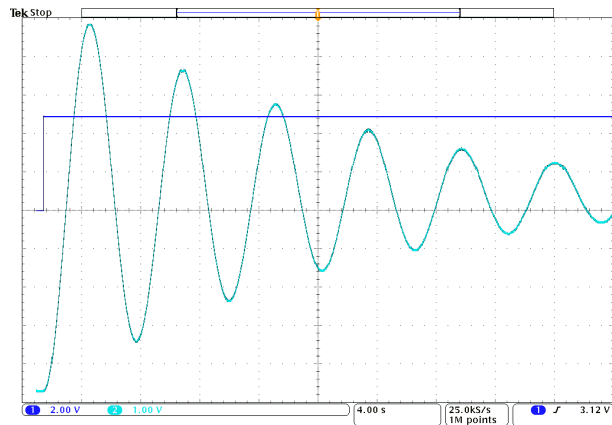
Fonte: Autoria própria

Figura 4.15 – Resposta ao degrau (Matlab) para sistema G_3 ; $f_{sw} = 100 \text{ Hz}$;



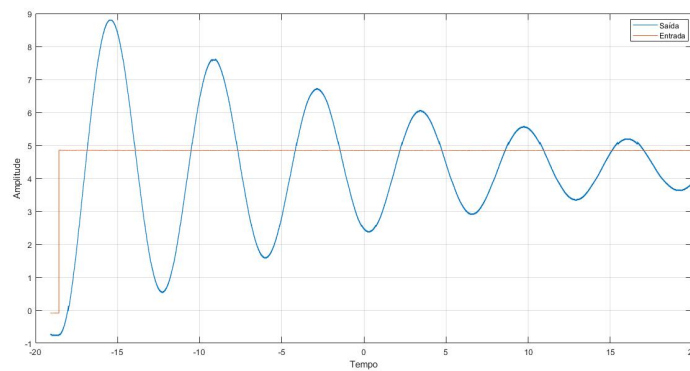
Fonte: Autoria própria

Figura 4.16 – Resultado experimental sistema G_2 ; $f_{sw} = 1\text{ kHz}$; $f_{sinal} = 40\text{ rad/s}$



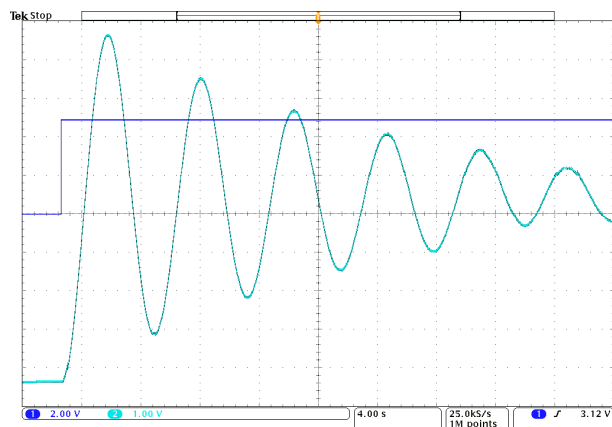
Fonte: Autoria própria

Figura 4.17 – Resposta ao degrau (Matlab) para sistema G_3 ; $f_{sw} = 1\text{ kHz}$;



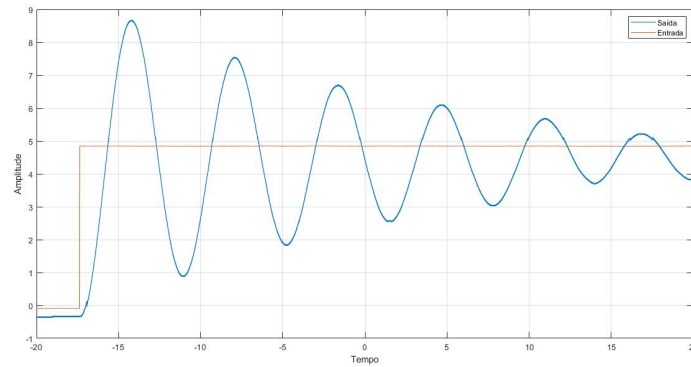
Fonte: Autoria própria

Figura 4.18 – Resultado experimental sistema G_3 ; $f_{sw} = 10\text{ kHz}$; $f_{sinal} = 40\text{ rad/s}$



Fonte: Autoria própria

Figura 4.19 – Resposta ao degrau (Matlab) para sistema G_3 ; $f_{sw} = 10 \text{ kHz}$;



Fonte: Autoria própria

O teste realizado com o último sistema também apresentou bom resultado, justificado pela senóide gerada pelos polos imaginários cuja frequência é igual a localização do eixo imaginário. A partir dos pontos plotados no Matlab foi possível ver a senóide da resposta não-amortecida e corroborar seu funcionamento.

5 CONCLUSÃO

O projeto apresentou o processo de projeto, simulação e montagem de uma plataforma didática *hardware-in-the-loop* de baixo custo para emulação de sistemas lineares como a integração dos componentes eletrônicos, confecção da placa de circuito impresso e do sistema de alimentação a partir da fonte simétrica. Assim como a implementação do sistema em microcontrolador com *software* auxiliar.

A plataforma apresentou bons resultados para sistemas até segunda ordem, visto que a velocidade de processamento desses foi satisfatória mesmo com o sistema de comunicação e a integração com o *hardware*. Após a implementação do sistema, notou-se a dificuldade para emulação de sistemas com mais de uma entrada e ordem superior a 2. Por esses motivos, faz-se necessário a implementação de outra estratégia de desenvolvimento de *firmware* para atingir a emulação de sistemas de ordem maior e atingir os parâmetros iniciais do projeto.

Apesar de não satisfazer todos requisitos de projeto, foi possível desenvolver o algoritmo inicial para simulação dos modelos matemáticos em tempo discreto, comunicação do computador com o microcontrolador e a validação dos circuitos eletrônicos. Visto que foi realizada a implementação da plataforma e fez-se a integração dos componentes de *hardware* e *software*, e ambos apresentaram um funcionamento satisfatório, pode-se afirmar que os principais objetivos do trabalho foram atingidos.

Para trabalhos futuros pode-se explorar o uso com microcontroladores em ponto flutuante e com conversor D/A integrado. Assim como pode-se simular sistemas com mais de duas entradas e duas saídas.

REFERÊNCIAS BIBLIOGRÁFICAS

- AL, M. O. F. et. Real-time simulation technologies for power systems design, testing, and analysis. **IEEE Power and Energy Technology Systems Journal**, v. 2, n. 2, p. 6373, 2015.
- ANDRADE, A. V. S.; STEPHAN, R. M. Didactic system for control of electrical machines in education and research laboratories. In: **BRAZILIAN POWER ELECTRONICS CONFERENCE**, 15., 2019, Santos. São Paulo: IEEE Power Electronics Society, 2019. Acesso em: 16/5/2020.
- BACIC, M. On hardware-in-the-loop simulation. In: . IEEE Conference on Decision and Control, 2005. Acesso em: 11/5/2022. Disponível em: <<https://ieeexplore.ieee.org/document/1582653>>.
- BALOG, R. S. et al. Modern laboratory-based education for power electronics and electric machines. **IEEE TRANSACTIONS ON POWER SYSTEMS**, v. 20, n. 2, p. 538–547, 2005.
- BEZERRA, M. A. **DESENVOLVIMENTO HARDWARE IN THE LOOP PARA AVALIAÇÃO DE REGULADORES DE TENSÃO NA DISTRIBUIÇÃO**. 2019. 110 f. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2019.
- BITTAR, N. M. F. d. O. A.; FIGUEIREDO, H. V. de. Hardware-in-the-loop simulation with x-plane of attitude control of a suav exploring atmospheric conditions. **Journal of Intelligent and Robotic Systems**, v. 73, n. 2, p. 271287 (2014), 2013.
- JIMÉNEZ-MARTÍNEZ, J. M. et al. A new approach for teaching power electronics converter experiments. **IEEE TRANSACTIONS ON EDUCATION**, v. 48, n. 3, p. 513–519, 2005.
- KOLEFF, L. et al. Development of a modular open source power electronics didactic platform. In: **BRAZILIAN POWER ELECTRONICS CONFERENCE**, 15., 2019, Santos. São Paulo: IEEE Power Electronics Society, 2019. Acesso em: 16/5/2020.
- National Instruments. **Hardware-in-the-Loop (HIL) Test System Architectures**: Home. NATIONAL INSTRUMENTS CORP., 2022. Acesso em 22 abr. 2022. Disponível em: <<https://www.ni.com/pt-br/innovations/white-papers/09/hardware-in-the-loop--hil--test-system-architectures.html>>.
- Patola. **Patola**: Home. Patola, 2019. Acesso em 06 abr. 2022. Disponível em: <<https://www.patola.com.br/index.php?route=common/home>>.
- RANDRIAKOTONJANAHARY, T. S. **A low-cost, small scale Unmanned Aerial Vehicle capable of a real-time onboard deep learning-based object detection system**. 2019. 113 p. Monografia (Final Thesis) — School of Engineering and the Built Environment, Glasgow Caledonian University, 2019.
- RUEDA, O. A. S. **SIMULADOR EM TEMPO-REAL BASEADO NA INTEGRAÇÃO DE MÓDULO FPGA E CPU_s PARA AVALIAÇÃO DE CONTROLADORES EMBARCADOS DE CONVERSORES ELETRÔNICOS DE POTÊNCIA**. 2019. 132 f. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2019.
- SARHADI, P.; YOUSEFPOUR, S. State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software. **International Journal of Dynamics and Control**, v. 3, n. 2, p. 470479 (2015), 2014.

SILVA, L. P. D. et al. Low-cost didactic module for single-phase inverter teaching. In: BRAZILIAN POWER ELECTRONICS CONFERENCE, 14., 2017, Juiz de Fora. Minas Gerais: IEEE Power Electronics Society, 2017. Acesso em: 27/5/2020.

SILVA, W. R. **Plataforma Hardware-in-the-loop para teste de leis de controle de voo e desenvolvimento de unidade de navegação inercial**. 2017. 199 f. Dissertação (Doutorado em Engenharia Elétrica) — Universidade Federal de Santa Maria, Santa Maria, 2017.

SOUSA, G. et al. Arquitetura para emulação de microrredes em fpga. In: . [S.l.: s.n.], 2020. Acesso em: 16/5/2020.

SPARN DHEEPAK KRISHNAMURTHY, A. P. B.; RUTH, M.; WU, H. Hardware-in-the-loop (hil) simulations for smart grid impact studies. In: IEEE POWER AND ENERGY SOCIETY GENERAL MEETING, 15., 2018, Portland. Oregon: IEEE Power Electronics Society, 2018. Acesso em: 11/5/2022.

STMicroelectronics. **Mainstream Performance line, Arm Cortex-M3 MCU with 64 Kbytes of Flash memory, 72 MHz CPU, motor control, USB and CAN**: Home. STMicroelectronics, 2022. Acesso em 06 abr. 2022. Disponível em: <<https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html>>.

_____. **STM32Cube initialization code generator**: Home. STMicroelectronics, 2022. Acesso em 06 abr. 2022. Disponível em: <<https://www.st.com/en/development-tools/stm32cubemx.html>>.

TAVANA, N. R.; DINAVAHI, V. Real-time nonlinear magnetic equivalent circuit model of induction machine on fpga for hardware-inthe-loop simulation. **IEEE Transactions on Energy Conversion**, v. 31, n. 2, p. 520–530, 2016.

TORREY, D. A. A project-oriented power electronics laboratory. **IEEE TRANSACTIONS ON POWER ELECTRONICS**, v. 9, n. 3, p. 250–255, 1994.

WILLIAMS, J. M. et al. Versatile hardware and software tools for educating students in power electronics. **IEEE TRANSACTIONS ON EDUCATION**, v. 47, n. 4, p. 436–445, 2004.

APÊNDICE A – MATERIAIS UTILIZADOS NO PROJETO DE HARDWARE

Material	Valor	Quantidade
Capacitor	100n	6
Capacitor	1 μ	4
Diodo	0.3	12
Diodo	15V	1
LED		2
Conec. JackP8		1
Resistor	150k	6
Resistor	10k	6
Resistor	2k	21
Resistor	15k	2
Resistor	3.3	2
Resistor	27k	2
Resistor	1k	22
Resistor	2.2k	2
Resistor	200	1
Resistor	270	2
PushButton		2
CI (TL064)		1
LM7912		1
LM7805		1
Conec. RCA		4
Caixa		1
Parafusos (p/ caixa)		4
Adaptador BNC/RCA		4
Cabo RCA		4
Headers (femêa)		2
TOTAL		116

Fonte: Aatoria própria

APÊNDICE B – CÓDIGO MATLAB PROJETO CIRCUITOS

```
%Projeto do circuito de condicionamento de entrada
```

```
Vimin=-12;
```

```
Voff=1.65;
```

```
Vt1431=2.5;
```

```
R2=2.2e3;
```

```
A=[Vimin Vt1431;-Voff Vt1431-Voff]
```

```
B=[0; Voff/R2]
```

```
G=inv(A)*B
```

```
R1=1/G(1)
```

```
R3=1/G(2)
```

```
%Projeto do circuito de saída
```

```
Vomin=Vimin;
```

```
R3s=10e3;
```

```
As=[-Voff -Voff; Vomin 0];
```

```
Bs=[Voff-Vt1431;-Vt1431];
```

```
Gs=(1/R3s)*inv(As)*Bs
```

```
R1s=1/Gs(1)
```

```
R2s=1/Gs(2)
```