



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

Alexandre Pedro Yure Cariri Gomes

Relatório de Estágio Integrado

Campina Grande, Paraíba, Brasil
16 de agosto de 2024

Alexandre Pedro Yure Cariri Gomes

Relatório de Estágio Integrado

Relatório de Estágio Integrado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Controle e Automação

Orientador: Prof. Gutemberg Gonçalves dos Santos Júnior, Dr.

Campina Grande, Paraíba, Brasil

16 de agosto de 2024

Alexandre Pedro Yure Cariri Gomes

Relatório de Estágio Integrado

Relatório de Estágio Integrado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Aprovado em: ___/___/___

Prof. Gutemberg Gonçalves dos Santos Júnior, Dr.
Orientador

Prof. Danilo Freire de Souza Santos, Dr.
Avaliador

Campina Grande, Paraíba, Brasil
16 de agosto de 2024

Agradecimentos

Gostaria de expressar minha profunda gratidão a todas as pessoas e instituições que me apoiaram durante a realização deste estágio.

Primeiramente, agradeço a Deus pelo dom da vida e por me dar forças para continuar, apesar das dificuldades e desafios enfrentados.

Agradeço ao meu pai, João Batista, e à minha mãe, Maria de Lourdes, por me acolherem, por seu suporte e motivação ao longo dessa jornada.

Agradeço à Universidade Federal de Campina Grande (UFCG) e ao Departamento de Engenharia Elétrica pela oportunidade de realizar este estágio e por todo o suporte educacional fornecido durante minha formação. Em especial, agradeço ao professor Gutemberg por sua orientação, paciência e por compartilhar seus conhecimentos e experiências, o que foi fundamental para o meu desenvolvimento profissional.

Resumo

No presente relatório, são descritas as atividades do Estágio Integrado desempenhadas pelo estagiário Alexandre Pedro Yure Cariri Gomes, estudante de Engenharia Elétrica na Universidade Federal de Campina Grande, durante o estágio no Laboratório EMBEDDED. Dentre as principais atividades realizadas, destacam-se o estudo e desenvolvimento de softwares para sistemas embarcados, utilizando microcontroladores Arduino e C2000, com foco na aplicação em malhas de controle. Foram empregados métodos como hardware-in-the-loop (HIL) e a geração automática de código pelo Model-Based Design utilizando o MATLAB Simulink.

Palavras-chaves: Sistemas embarcados, C++, HIL, Model-Based Design, Estágio integrado.

Abstract

This report details the activities undertaken by the intern, Alexandre Pedro Yure Cariri Gomes, a student of Electrical Engineering at the Federal University of Campina Grande, during his internship at the EMBEDDED Laboratory. The primary activities involved studying and developing software for embedded systems, utilizing Arduino and C2000 microcontrollers with a focus on control loops. Key methods employed included hardware-in-the-loop (HIL) and automatic code generation through Model-Based Design using MATLAB Simulink.

Key-words: Embedded, C++, HIL, Model-Based Design, Integrated internship.

Lista de abreviaturas e siglas

UAEE	Unidade Acadêmica de Engenharia Elétrica
CEEI	Centro de Engenharia Elétrica e Informática
UFMG	Universidade Federal de Campina Grande
HIL	Hardware-in-the-Loop
MBD	Model-Based Design
CCS	Code Composer Studio
IDE	Integrated Development Environment
CC	Corrente Contínua
CA	Corrente Alternada

Lista de ilustrações

Figura 1 – Fotografia do Arduino UNO	5
Figura 2 – Fotografia do c2000	5
Figura 3 – maquina síncrona	8
Figura 4 – Circuito equivalente	9
Figura 5 – Diagrama PI	9
Figura 6 – Comunicação serial entre Arduino e Matlab	13
Figura 7 – Modelagem Para Arduino HIL Básico	14
Figura 8 – Implementação Matlab HIL Básico	15
Figura 9 – DC Machine	15
Figura 10 – Controlled Voltage Source	16
Figura 11 – Diagrama de Simulação Controle	17
Figura 12 – Resultados da malha de controle	17
Figura 13 – Sistema de Controle implementado ao microcontrolador Arduino	18
Figura 14 – Planta de motor CC	18
Figura 15 – Resultados de controle usando microcontrolador	19

Sumário

1	INTRODUÇÃO	1
1.1	Objetivos	1
1.1.1	Objetivo Geral	1
1.1.2	Objetivos Específicos	1
2	O LABORATÓRIO EMBEDDED	2
3	FUNDAMENTAÇÃO TEÓRICA	3
3.1	Sistemas Embarcados	3
3.2	Microcontroladores	4
3.2.1	Arduino Uno	4
3.2.2	C2000	5
3.3	Model-Based Design	5
3.4	Hardware-In-The-Loop	7
3.5	Motor CC	7
3.5.1	Controle PI Aplicado a Motor CC	8
4	ATIVIDADES REALIZADAS	10
4.1	Preparação do Ambiente de Desenvolvimento	10
4.1.1	MATLAB	10
4.1.2	Arduino IDE	11
4.1.3	Code Composer Studio	11
4.1.4	Drivers dos Microcontroladores	11
4.2	Teste de Conexão serial	12
4.3	HIL Básico	13
4.4	Controlador PI para Motor CC (simulado)	15
4.5	Controlador PI para Motor CC (HIL)	18
5	CONSIDERAÇÕES FINAIS	20
	REFERÊNCIAS	21

1 Introdução

Este relatório apresenta as atividades desenvolvidas durante o Estágio Integrado realizado como parte do curso de Engenharia Elétrica na Universidade Federal de Campina Grande (UFCG). O estágio integrado é uma etapa crucial para a formação profissional, permitindo a aplicação prática dos conhecimentos adquiridos ao longo da graduação e proporcionando uma imersão no ambiente de trabalho. Além disso, o estágio oferece a oportunidade de trabalhar em equipe, desenvolver habilidades específicas da área e compreender melhor os desafios enfrentados em projetos reais.

Neste relatório, são descritas as atividades realizadas durante a execução da disciplina de Estágio Supervisionado pelo aluno Alexandre Pedro Yure Cariri Gomes, no curso de graduação em Engenharia Elétrica na UFCG. O estágio foi realizado no período de 9 de março a 7 de julho, totalizando 274 horas de atividades desenvolvidas.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver sistemas de controle utilizando técnicas de *Model-Based Design* e validar o desempenho desses sistemas por meio de testes com *Hardware-In-the-Loop*.

1.1.2 Objetivos Específicos

Como objetivos específicos do estágio, podem ser destacadas as seguintes atividades realizadas:

- Estudar e desenvolver software para sistemas embarcados utilizando microcontroladores Arduino e C2000, com foco na aplicação em malhas de controle .
- Empregar métodos como *Hardware-In-the-Loop* (HIL) e a geração automática de código pelo *Model-Based Design* utilizando MATLAB Simulink.
- Aprimorar habilidades técnicas e profissionais, compreendendo melhor os desafios enfrentados em projetos reais e aplicando soluções práticas.
- Documentar todas as atividades desenvolvidas durante o estágio, detalhando os projetos realizados, os métodos utilizados e os resultados obtidos, contribuindo para a avaliação do aproveitamento e aplicação prática dos conhecimentos adquiridos ao longo da graduação.

2 O Laboratório EMBEDDED

O laboratório Embedded, situado na Universidade Federal de Campina Grande (UFCG), faz parte do grupo de laboratórios do CEEI. Fundado em 2005, é um centro de pesquisa e desenvolvimento dedicado ao estudo e à aplicação de sistemas embarcados. O laboratório se destaca por sua infraestrutura moderna e pelo ambiente colaborativo, onde estudantes e pesquisadores trabalham em conjunto para desenvolver soluções inovadoras em diversas áreas da engenharia elétrica e computação.

No laboratório Embedded, são realizadas pesquisas avançadas e projetos práticos que envolvem a utilização de microcontroladores, sistemas de controle, e métodos de desenvolvimento baseados em modelos (*Model-Based Design*), utilizando ferramentas como MATLAB Simulink. O laboratório promove uma integração entre teoria e prática, permitindo que os alunos apliquem os conhecimentos adquiridos em sala de aula em projetos reais, preparando-os para os desafios do mercado de trabalho.

3 Fundamentação Teórica

A fundamentação teórica é uma parte essencial deste relatório, pois fornece a base científica e técnica para as atividades desenvolvidas durante o estágio. Nesta seção, serão abordados os conceitos e teorias fundamentais que sustentam o desenvolvimento de sistemas embarcados e as metodologias utilizadas no estágio. A compreensão desses conceitos é crucial para a aplicação prática e para a resolução dos desafios enfrentados em projetos reais. A seguir, são apresentados os principais tópicos teóricos que embasaram as atividades realizadas no estágio.

3.1 Sistemas Embarcados

Sistemas embarcados são sistemas computacionais dedicados que realizam funções específicas dentro de um sistema maior. Eles são projetados para executar uma ou várias tarefas específicas, geralmente com requisitos de tempo real e restrições de recursos, como potência e memória. Esses sistemas são encontrados em uma ampla gama de dispositivos, desde eletrodomésticos e automóveis até equipamentos médicos e sistemas de controle industrial (Heath 2002).

Um sistema embarcado típico é composto por vários componentes principais: o microcontrolador ou microprocessador, memória, interfaces de entrada e saída (I/O) e, frequentemente, sensores e atuadores. O microcontrolador é o núcleo do sistema, responsável por executar o software embarcado que controla as operações do dispositivo. A memória, que pode incluir RAM, ROM e EEPROM, armazena o código do programa e os dados necessários para a operação. As interfaces de I/O permitem que o sistema interaja com o mundo externo, recebendo informações de sensores e enviando comandos para atuadores.

O desenvolvimento de sistemas embarcados envolve várias etapas, desde a especificação dos requisitos até o projeto, implementação, teste e manutenção. O processo começa com a definição dos requisitos do sistema, incluindo funcionalidade, desempenho, consumo de energia e custo. Em seguida, é feito o projeto do hardware e do software. Ferramentas de desenvolvimento como MATLAB/Simulink, *Integrated Development Environment* (IDEs) e kits de desenvolvimento de hardware são frequentemente utilizados.

A programação de sistemas embarcados é feita geralmente como C/C++ e Assembly, devido à necessidade de controle preciso do hardware e eficiência de recursos. No entanto, linguagens de alto nível, como Python e Java, também são usadas em alguns casos. A programação envolve a escrita de código para inicializar o hardware, controlar

dispositivos de I/O, processar dados de sensores, tomar decisões baseadas em algoritmos de controle e acionar atuadores.

Os sistemas embarcados são utilizados em diversas aplicações, refletindo sua versatilidade e importância em várias indústrias. No setor automotivo, eles controlam desde o motor e a transmissão até sistemas de entretenimento e assistência ao motorista. Em dispositivos médicos, monitoram sinais vitais e controlam equipamentos de diagnóstico e tratamento. Na indústria, são utilizados em sistemas de controle de processos, automação e robótica. Em produtos de consumo, como smartphones, eletrodomésticos e dispositivos de IoT (*Internet of Things*), os sistemas embarcados proporcionam funcionalidades avançadas e conectividade

3.2 Microcontroladores

Microcontroladores são dispositivos semicondutores que integram em um único chip um processador, memória e periféricos de entrada/saída. Eles são projetados para controlar sistemas embarcados, executando tarefas específicas e respondendo a eventos em tempo real. Os microcontroladores são amplamente utilizados em uma variedade de aplicações, desde eletrodomésticos até sistemas automotivos e dispositivos médicos. Sua interação com o mundo físico pode ser realizada através de sensores e atuadores.

3.2.1 Arduino Uno

O Arduino Uno (Figura 1) , é uma das plataformas de desenvolvimento de microcontroladores. Equipado com um microcontrolador Atmega328P de 8 bits que opera a uma frequência de 16 MHz. Ele oferece 14 pinos digitais configuráveis como entrada ou saída, sendo que 6 desses suportam saídas PWM (*Pulse Width Modulation*). Além disso, possui 6 pinos de entrada analógica, permitindo a conexão de sensores que utilizam sinais analógicos (Arduino.cc).

A programação do Arduino Uno é realizada através da linguagem C/C++. Esta plataforma inclui um IDE intuitivo e de fácil utilização. A linguagem Arduino simplifica o processo de programação ao oferecer uma ampla gama de bibliotecas e funções pré-definidas, facilitando o acesso aos recursos do microcontrolador sem necessidade de conhecimento detalhado de baixo nível.

O Arduino Uno é amplamente adotado devido à sua simplicidade e à robusta comunidade de suporte disponível, tornando-o uma escolha popular para iniciantes e desenvolvedores que buscam uma solução prática e acessível para projetos eletrônicos.

Figura 1 – Fotografia do Arduino UNO

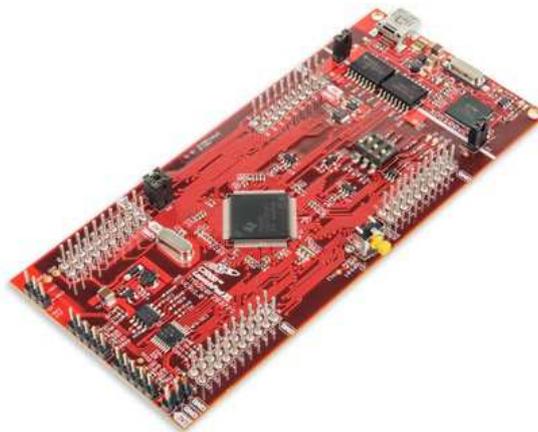


Fonte: ([Arduino.cc](https://www.arduino.cc) 2024)

3.2.2 C2000

A série C2000 da Texas Instruments (Figura 2), é uma família de microcontroladores projetados para aplicações de controle digital de alta performance, especialmente em controle de motores e fontes de alimentação digitais. Os microcontroladores C2000 são conhecidos por sua capacidade de processamento rápido e eficiente, integração de periféricos específicos para controle de potência e suporte a algoritmos de controle complexos.

Figura 2 – Fotografia do c2000



Fonte: ([Texas Instruments](https://www.ti.com) 2024)

3.3 Model-Based Design

O *Model-Based Design* (MBD) é de fato uma metodologia avançada que transforma o desenvolvimento de sistemas complexos ao integrar modelagem, simulação e geração de código automatizada. Originado nos setores aeroespacial e automotivo, o MBD agora está sendo adotado em diversas indústrias onde máquinas complexas controladas por compu-

tador são desenvolvidas. Ao contrário das abordagens tradicionais, que frequentemente dependem da codificação manual em linguagens de programação, o MBD enfatiza a criação de modelos matemáticos e de simulação que representam o comportamento do sistema de forma abstrata e detalhada. Esses modelos não apenas permitem uma compreensão profunda do sistema, mas também facilitam análises rigorosas, simulações detalhadas e testes exaustivos antes da implementação final em hardware.

No MBD, o processo se inicia com a criação de modelos que representam o comportamento do sistema real. Esses modelos podem incluir diagramas de blocos, funções de transferência, equações e outros modelos matemáticos que ajudam a descrever o funcionamento do sistema de maneira detalhada, porém abstraída de detalhes que poderiam tornar o problema excessivamente complexo. Essa abordagem permite uma representação precisa e estruturada do sistema, facilitando análises, simulações e testes antes da implementação final em hardware.

Após a criação dos modelos, inicia-se a fase de simulação. Esta etapa envolve o teste do sistema, levando em consideração diversos cenários operacionais, incluindo condições extremas e situações de falhas simuladas. A simulação é essencial para validar o modelo, identificar possíveis erros, e implementar melhorias e otimizações no design antes da implementação final em hardware.

Um dos grandes benefícios do MBD é que ele permite criar automaticamente código, a partir dos modelos que foram criados e testados. Isso é feito usando ferramentas que estão integradas ao software de modelagem. Essas ferramentas transformam os modelos em código de programação que é otimizado para o tipo específico de dispositivo que será usado. Esse processo automatizado reduz muito o tempo e o esforço necessários para colocar o sistema em funcionamento. Além disso, ele ajuda a evitar erros que poderiam surgir se o código fosse feito manualmente.

Essa metodologia proporciona diversas vantagens significativas, incluindo a redução de erros e custos através da detecção precoce de problemas durante as fases iniciais do desenvolvimento. Além disso, os modelos visuais facilitam a comunicação entre equipes multidisciplinares e *stakeholders*, promovendo uma compreensão clara do sistema em desenvolvimento. A flexibilidade e reutilização dos modelos permitem adaptações rápidas às mudanças de requisitos, enquanto a automação na geração de código e a validação antecipada aceleram o processo de desenvolvimento e permitem uma introdução mais rápida de novos produtos no mercado. Essas características fazem do MBD uma metodologia eficaz para aumentar a eficiência, qualidade e competitividade no desenvolvimento de sistemas complexos em diversas indústrias.

3.4 Hardware-In-The-Loop

Hardware-in-the-Loop (HIL) é uma técnica de teste que integra componentes de hardware reais em um ambiente de simulação, permitindo a avaliação de sistemas complexos de maneira segura e controlada. Esta metodologia é amplamente utilizada em diversas áreas da indústria, para testar e validar controladores e sistemas embarcado.

O desenvolvimento de modelos precisos é crucial para o sucesso dos testes HIL. Ferramentas como MATLAB/Simulink são usadas para criar modelos de simulação detalhados que devem replicar fielmente o comportamento do ambiente operacional ([MathWorks](#)).

O uso de HIL oferece várias vantagens significativas. Primeiramente, a segurança é um dos principais benefícios, pois permite testar controladores e sistemas em condições extremas ou de falha sem risco de danos ao sistema completo ou à segurança dos operadores. Em termos de custo, o HIL reduz a necessidade de protótipos físicos completos, permitindo a detecção precoce de falhas e economizando tempo e recursos. A flexibilidade é outro ponto forte, pois facilita a modificação de modelos de simulação, permitindo a avaliação de diferentes cenários e condições operacionais. Por fim, a eficiência do HIL acelera o processo de desenvolvimento e validação, permitindo testes contínuos e repetíveis, o que contribui significativamente para a melhoria e a inovação em diversos setores industriais ([Li e Ang 2006](#)).

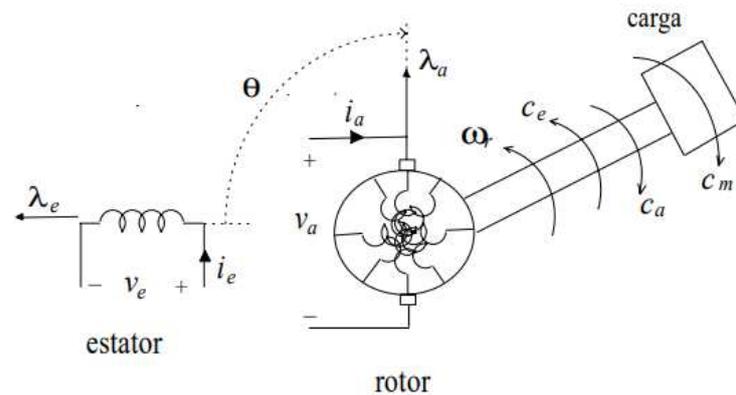
3.5 Motor CC

As máquinas de corrente contínua (CC) ([Figura 3](#)) são conhecidas por sua versatilidade, pois podem ser projetadas com diferentes combinações de enrolamentos de campo - em derivação, série ou independentes - permitindo uma ampla variedade de características de tensão versus corrente ou velocidade versus torque, tanto para operações dinâmicas quanto para regime permanente. Devido à facilidade de controle, esses motores são frequentemente usados em aplicações que exigem uma ampla faixa de velocidades ou controle preciso. Embora a tecnologia de acionamento de corrente alternada (CA) tenha avançado e começado a substituir as máquinas CC em várias aplicações, a simplicidade e versatilidade das máquinas CC garantem seu uso contínuo em muitas áreas ([Fitzgerald e Jr. 2003](#)).

O estator é a parte fixa da máquina que fornece o campo magnético necessário para a operação, podendo ser composto por ímãs permanentes ou por bobinas de campo alimentadas por corrente elétrica. O rotor, também conhecido como armadura, é a parte móvel da máquina onde a corrente é induzida, criando um campo magnético que interage com o campo do estator. O rotor é montado no eixo do motor, que gira para realizar trabalho mecânico. O comutador é um dispositivo rotativo que alterna a direção da cor-

rente elétrica nas bobinas do rotor, garantindo que a força gerada sempre atue na direção necessária para manter a rotação. As escovas são componentes que fazem contato com o comutador, permitindo a transferência de corrente elétrica do circuito externo para o rotor (Han 2015).

Figura 3 – máquina síncrona



Fonte: (Jacobina 2005)

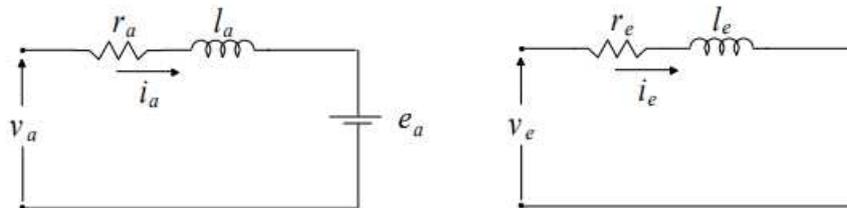
3.5.1 Controle PI Aplicado a Motor CC

As máquinas CC são amplamente utilizadas em diversos setores industriais e comerciais por sua habilidade de controlar com precisão a velocidade e o torque. Elas são também conhecidas por sua simplicidade em termos de projeto e operação. Essas máquinas possuem dois componentes principais: o enrolamento de campo e o enrolamento de armadura. O enrolamento de campo gera um campo magnético fixo, enquanto o enrolamento de armadura é responsável por criar movimento mecânico ou corrente elétrica, dependendo do uso.

No caso de um motor CC, quando uma corrente elétrica passa pelo enrolamento de armadura, ela interage com o campo magnético do enrolamento de campo, produzindo torque que faz o rotor girar. Por outro lado, em um gerador CC, a rotação mecânica do rotor dentro do campo magnético gera uma corrente elétrica no enrolamento de armadura, devido à variação do fluxo magnético.

O circuito equivalente visto na Figura 4, pode ser utilizado para simular o comportamento da máquina de corrente contínua sob diversas condições, como partida, aceleração, desaceleração e operação em regime permanente. Esse modelo é também valioso no desenvolvimento de sistemas de controle para a máquina, como os controles de velocidade e torque.

Figura 4 – Circuito equivalente



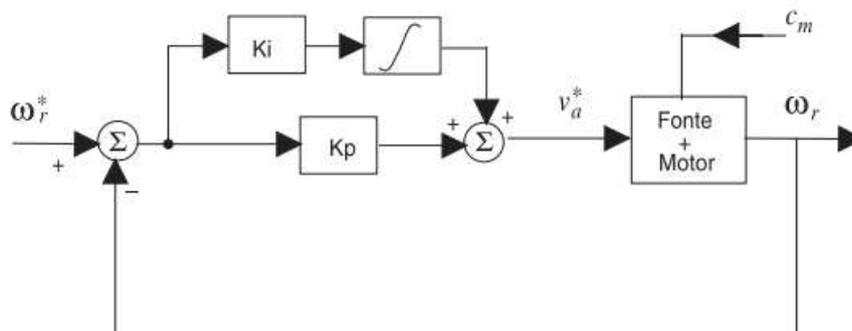
Fonte: (Jacobina 2005)

O controlador proporcional compara a variável de referência ω_r^* com a variável de saída medida ω_r definindo o erro $e_r = \omega_r^* - \omega_r$ que, amplificado por um ganho K_p , define a variável de entrada do sistema sob controle. Esse controlador não proporciona erro zero. Tendo em vista que para erro zero o valor de K_p deverá apresentar um valor infinito o que não é possível em sistema computacional.

Para assegurar que o erro estacionário do sistema em malha fechada seja zero, é necessário que ao menos uma das partes da função de transferência do controlador possua um pólo em $s = 0$ (integrador). O controlador do tipo PI com função de transferência $D(s) = k_p + k_i/s$ tem um polo em $s = 0$, que assegura um erro estacionário nulo, e um zero em $s = -k_i/k_p$.

Para o dimensionamento das constantes k_p e k_i do controlador PI (Figura 5) pode-se utilizar uma técnica de projeto baseada no cancelamento do polo dominante (mais lento) do sistema e alocação dos polos do sistema em malha fechada segundo o comportamento dinâmico especificado. Este procedimento simplifica a dedução dos valores dos ganhos do controlador .

Figura 5 – Diagrama PI



Fonte: (Jacobina 2005)

4 Atividades Realizadas

Nessa seção, serão apresentadas as atividades realizadas pelo estagiário, detalhando os processos e as experiências adquiridas ao longo desse período. As atividades desenvolvidas focaram na aplicação prática dos conhecimentos teóricos adquiridos ao longo do curso de Engenharia Elétrica. As atividades foram estruturadas com uma complexidade crescente e serão detalhadas nos próximos tópicos, proporcionando uma visão clara e abrangente do aprendizado e das habilidades desenvolvidas durante o estágio.

4.1 Preparação do Ambiente de Desenvolvimento

Antes de iniciar as atividades de forma efetiva, foi necessário realizar a instalação e configuração de diversas ferramentas essenciais. Esse processo envolveu a preparação do ambiente de desenvolvimento, garantindo que todos os softwares e hardwares necessários estivessem adequadamente configurados e funcionando corretamente. As principais ferramentas instaladas e configuradas incluem:

4.1.1 MATLAB

A escolha do software MATLAB se deu pela sua ampla capacidade de modelagem, simulação e análise de sistemas complexos. MATLAB é uma ferramenta robusta e versátil, especialmente adequada para o desenvolvimento e validação de sistemas embarcados. Ele permite a criação de modelos matemáticos precisos que representam o comportamento de sistemas reais, facilitando a análise e a compreensão de suas dinâmicas. O ambiente Simulink, integrado ao MATLAB, proporciona uma plataforma gráfica intuitiva para a simulação de sistemas. Além disso, MATLAB/Simulink oferece ferramentas para a geração automática de código C/C++ (*EMBEDDED CODER*), a partir de modelos simulados, otimizando o processo de desenvolvimento e reduzindo erros de codificação manual. Suas funcionalidades para análise e visualização de dados permitem uma avaliação detalhada dos resultados das simulações. A vasta biblioteca de funções e a capacidade de criar *scripts* personalizados tornam o MATLAB altamente flexível e adaptável às necessidades específicas de cada projeto. Por fim, a ampla comunidade de usuários e a disponibilidade de documentação e tutoriais proporcionam um suporte significativo para a resolução de problemas e o aprimoramento contínuo das habilidades no uso do software.

- Realizada a instalação do software MATLAB na versão R2023b, com uso da licença estudante;

- Durante a instalação foi adicionada a ferramenta Simulink como pacote adicional em sua versão de teste;
- Foi baixado e instalado o pacote de ferramentas "Support Package for Arduino Hardware", que permite a utilização e conexão com o Arduino .

4.1.2 Arduino IDE

O Arduino IDE é uma plataforma de software utilizada para escrever, compilar e fazer upload de códigos para placas de microcontroladores Arduino. A IDE é uma ferramenta fundamental para o desenvolvimento com Arduino, oferecendo uma interface amigável e um conjunto de recursos que facilitam a programação e a depuração de projetos eletrônicos.

4.1.3 Code Composer Studio

O Code Composer Studio (CCS) é uma ferramenta integrada de desenvolvimento (IDE) desenvolvida pela Texas Instruments para programar, depurar e analisar microcontroladores e processadores digitais de sinal (DSP) fabricados pela TI. O CCS oferece um conjunto abrangente de ferramentas e recursos projetados para facilitar o desenvolvimento de sistemas embarcados de alto desempenho.

4.1.4 Drivers dos Microcontroladores

Os drivers podem ser instalados a partir da IDE de cada fabricante ou através do site oficial. A instalação destes componentes é crucial para garantir que o sistema operacional possa reconhecer o hardware e permitir uma comunicação eficiente para desenvolvimento e depuração.

A IDE do Arduino inclui várias bibliotecas que facilitam a interação com sensores, atuadores e outros componentes. Essas bibliotecas são fundamentais para simplificar a programação e utilização dos recursos do microcontrolador. Elas podem ser instaladas diretamente pela IDE do Arduino através do gerenciador de bibliotecas.

CCS fornece bibliotecas específicas para os microcontroladores C2000, que incluem APIs para controle de motores, processamento de sinal e outras funcionalidades avançadas. Essas bibliotecas são geralmente incluídas nos pacotes de suporte ao dispositivo (Device Support Packages) e podem ser configuradas e adicionadas aos projetos conforme necessário.

4.2 Teste de Conexão serial

Inicialmente, para familiarizar-se com as ferramentas e o microcontrolador em questão, foi realizado um estudo preliminar abrangente dos conceitos que seriam abordados durante o estágio. Isso incluiu a compreensão do HIL, o estudo das ferramentas de MBD do MATLAB e a investigação das bibliotecas de comunicação entre MATLAB e Arduino. Posteriormente, foi conduzido um pequeno projeto piloto com o objetivo de validar esses conceitos e testar a comunicação serial entre as plataformas.

Foi feito o upload do código abaixo para o micro controlador Arduino:

```
1 void setup() {
2     Serial.begin(9600); // Inicia a comunicacao serial
3 }
4
5 void loop() {
6     if (Serial.available() > 0) { // Verifica se ha dados disponiveis
7         para ler
8         String mensagem = Serial.readString(); // Le a mensagem serial
9         Serial.println("Mensagem recebida pelo Arduino: " + mensagem);
10        // Envia mensagem de confirmacao
11    }
12 }
```

Listing 4.1 – Código de Teste para Comunicação Serial do Arduino

Em seguida foi executado o seguinte código no MATLAB:

```
1 % Configuracao da comunicacao serial com o Arduino
2 s = serial('COM12', 'BaudRate', 9600);
3
4 try
5     % Abre a comunicacao serial
6     fopen(s);
7
8     % Loop de comunicacao
9     while true
10        % Envia uma mensagem para o Arduino
11        fprintf(s, 'Hello Arduino!\n');
12
13        % Espera um curto periodo de tempo para o Arduino processar
14        pause(0.3);
15
16        % Verifica se ha dados recebidos
17        if s.BytesAvailable > 0
18            % Le e exhibe a mensagem recebida do Arduino
```

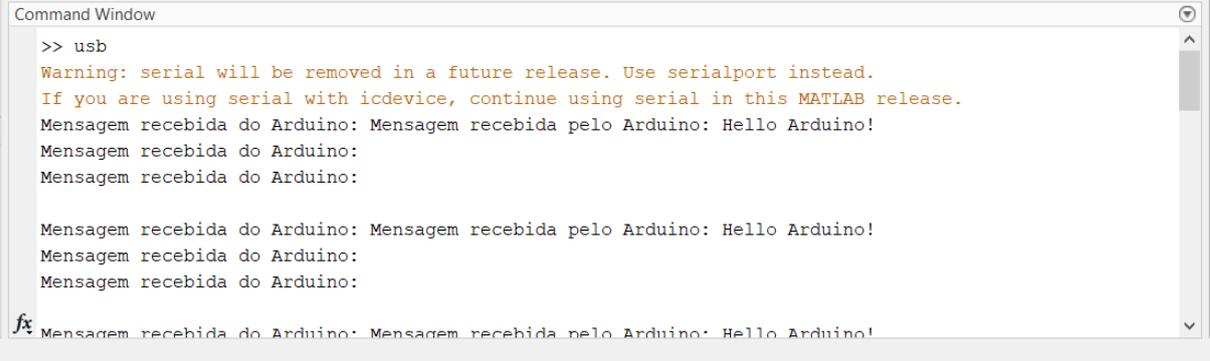
```

19     mensagem_recebida = fgetl(s);
20     disp(['Mensagem recebida do Arduino: ' mensagem_recebida]);
21     end
22
23     % Espera um pouco antes de enviar a proxima mensagem
24     pause(1);
25     end
26 catch
27     % Em caso de erro, fecha a comunicacao serial
28     fclose(s);
29     delete(s);
30     error('Erro na comunicacao com o Arduino.');
```

Listing 4.2 – Código de Teste para Comunicação Serial do Matlab

O resultado obtido ao executar o código pode ser visto na Figura 6, podendo assim validar a comunicação serial entre as duas plataformas.

Figura 6 – Comunicação serial entre Arduino e Matlab



```

Command Window
>> usb
Warning: serial will be removed in a future release. Use serialport instead.
If you are using serial with icdevice, continue using serial in this MATLAB release.
Mensagem recebida do Arduino: Mensagem recebida pelo Arduino: Hello Arduino!
Mensagem recebida do Arduino:
Mensagem recebida do Arduino:

Mensagem recebida do Arduino: Mensagem recebida pelo Arduino: Hello Arduino!
Mensagem recebida do Arduino:
Mensagem recebida do Arduino:
fx Mensagem_recebida_do_Arduino: Mensagem_recebida_pelo_Arduino: Hello Arduino!
```

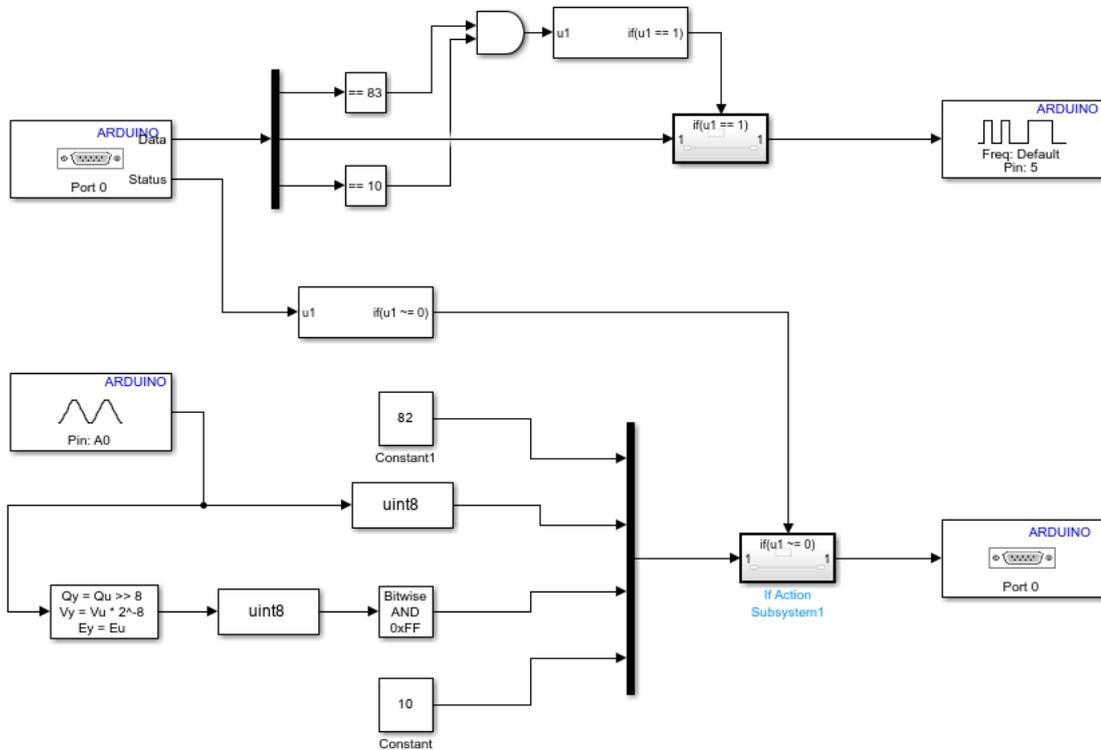
Fonte: Autoria própria.

4.3 HIL Básico

Nesta atividade, o estagiário desenvolveu uma maneira de comunicar as duas plataformas com o objetivo de introduzir o conceito de hardware in the loop. Para isso, um potenciômetro foi utilizado para a geração de um sinal analógico, que foi então enviado através da porta serial para o MATLAB. O MATLAB recebe esses dados, realiza o tratamento necessário e, em seguida, envia um sinal de volta para o Arduino. Este sinal retornado controla a variação do brilho de um LED.

Em vez de escrever o código para configurar o Arduino de maneira tradicional, utilizou-se a ferramenta de geração automática de código Embedded Coder associada ao Simulink. Esta ferramenta não apenas permite a geração de código, mas também facilita a depuração e o upload do código para o dispositivo.

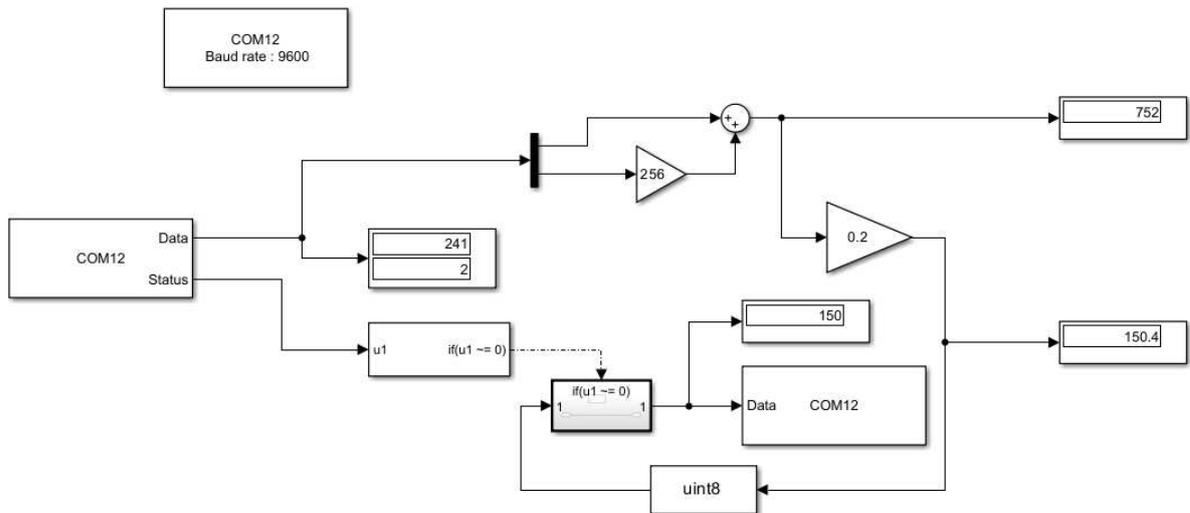
Figura 7 – Modelagem Para Arduino HIL Básico



Fonte: Autoria Própria.

Foi utilizado o modelo a seguir no Simulink para realizar a comunicação com o Arduino. Nele, é possível ver a conexão de entrada e saída dos dados pela porta serial, além do tratamento dos dados. Neste caso, o tratamento de dados envolve principalmente a manipulação de bits e a aplicação de ganhos. Este modelo demonstra a capacidade de interagir com o hardware real, proporcionando um ambiente de teste eficiente e flexível para o desenvolvimento de sistemas embarcados.

Figura 8 – Implementação Matlab HIL Básico



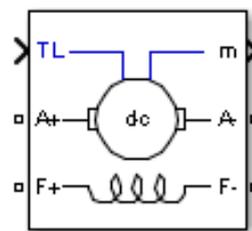
Fonte: Autoria Própria.

4.4 Controlador PI para Motor CC (simulado)

O estagiário desenvolveu um modelo de controle proporcional integral (PI) para uma máquina de corrente contínua (CC), focando na velocidade do eixo como variável de controle. A saída do sistema de controle é uma variável que será utilizada para ajustar a tensão.

Para modelar virtualmente um motor CC, foi utilizado o bloco *DC Machine* (Figura 9) da biblioteca Simscape do MATLAB Simulink. Detalhes sobre a validação deste modelo e as métricas utilizadas podem ser encontrados em [DC Machine](#).

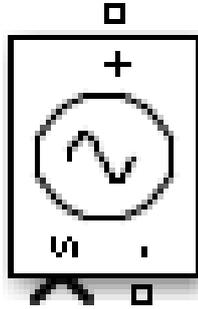
Figura 9 – DC Machine



Fonte: (MathWorks 2024)

Para controlar a tensão de armadura da maquina CC, foi utilizado o bloco Controlled Voltage Source (Figura 10), que gera uma tensão proporcional ao sinal de entrada do bloco.

Figura 10 – Controlled Voltage Source

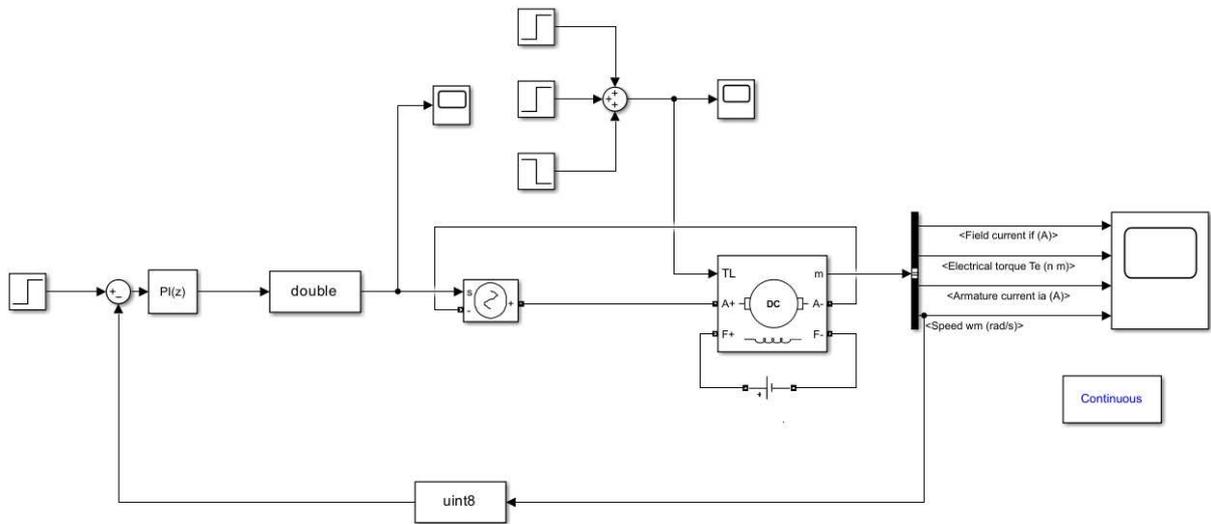


Fonte: (MathWorks 2024)

Na Figura 11 é possível visualizar o diagrama utilizado para modelar o experimento. Nesse diagrama, é fornecido um sinal de referência em forma de degrau para definir a velocidade desejada para o motor. A diferença entre esse sinal de referência e a velocidade real do motor é considerada o erro, que serve como entrada para o controlador PI. O controlador, por sua vez, ajusta a tensão da fonte de alimentação do motor.

Para simular o acréscimo de cargas no eixo do motor e introduzir perturbações na velocidade, foram adicionados degraus de carga na entrada TL do bloco. Os parâmetros do controlador PI foram calculados utilizando a ferramenta PID Tuner, disponível no Simulink. Esta ferramenta permite selecionar características específicas do sistema e, em seguida, gera automaticamente os valores para K_p , K_i e K_d . Na saída do bloco do motor, são fornecidas informações sobre o seu desempenho, como velocidade, corrente de armadura, corrente de campo e torque elétrico.

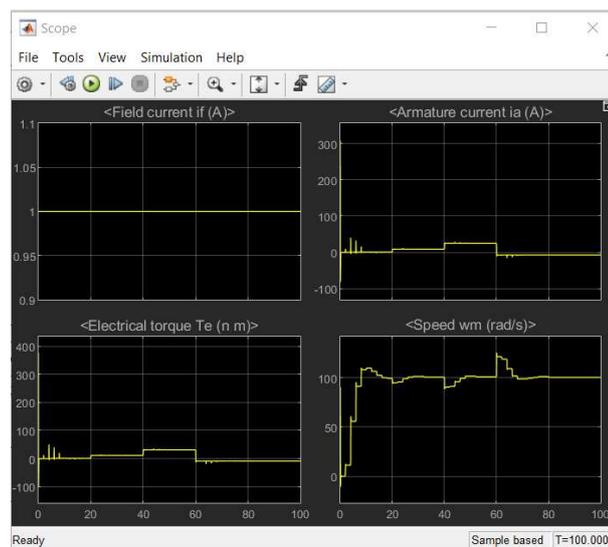
Figura 11 – Diagrama de Simulação Controle



Fonte: Autoria Própria.

O resultado do sistema (Figura 12) de controle foi satisfatório, possibilitando o controle da velocidade do motor com resposta em espaço de tempo aceitável. No entanto, ocorreram alguns overshoots na velocidade e na corrente de armadura. Se esses overshoots forem problemáticos na aplicação real, ajustes nas constantes do controlador podem ser realizados para mitigá-los. A resposta a seguir foi obtida durante a execução da simulação:

Figura 12 – Resultados da malha de controle



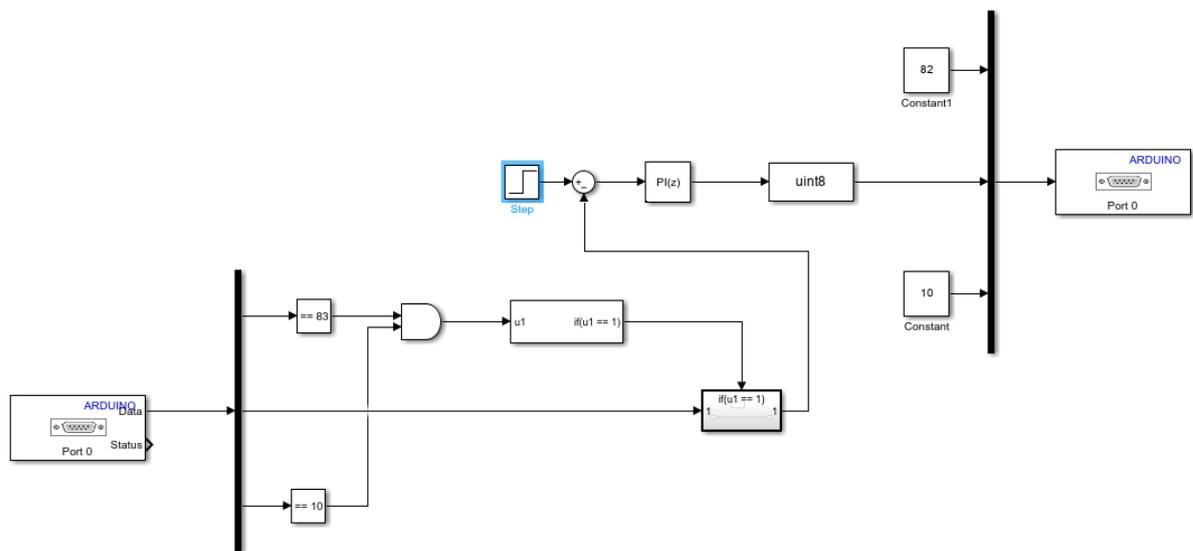
Fonte: Autoria Própria.

4.5 Controlador PI para Motor CC (HIL)

Nesta etapa foi feito o seccionamento do projeto anterior, com o objetivo de fazer com que o sistema de controle fosse executado no microcontrolador, e controlasse a planta que se encontra no matlab.

O diagrama da Figura 13 foi criado, com o objetivo de implementar o sistema de controle no microcontrolador arduino:

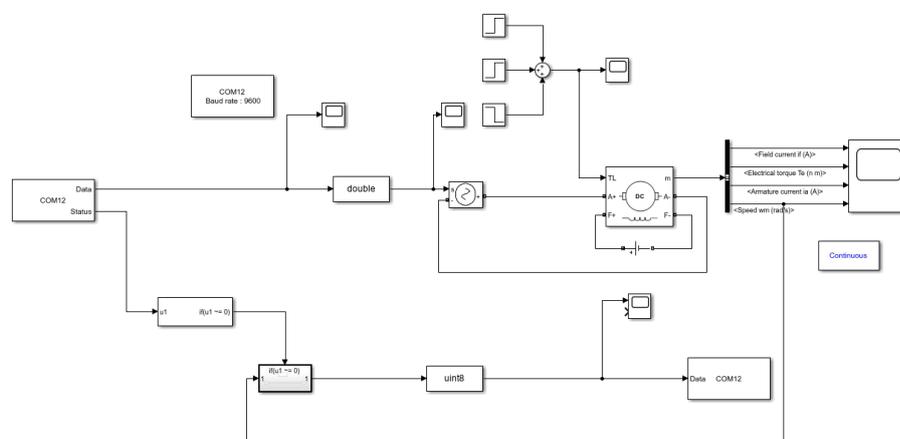
Figura 13 – Sistema de Controle implementado ao microcontrolador Arduino



Fonte: Autoria Própria.

A planta de teste (Figura 14) foi implementada, idêntica ao modelo completamente simulado, mantendo os paramentos do controlador os mesmos.

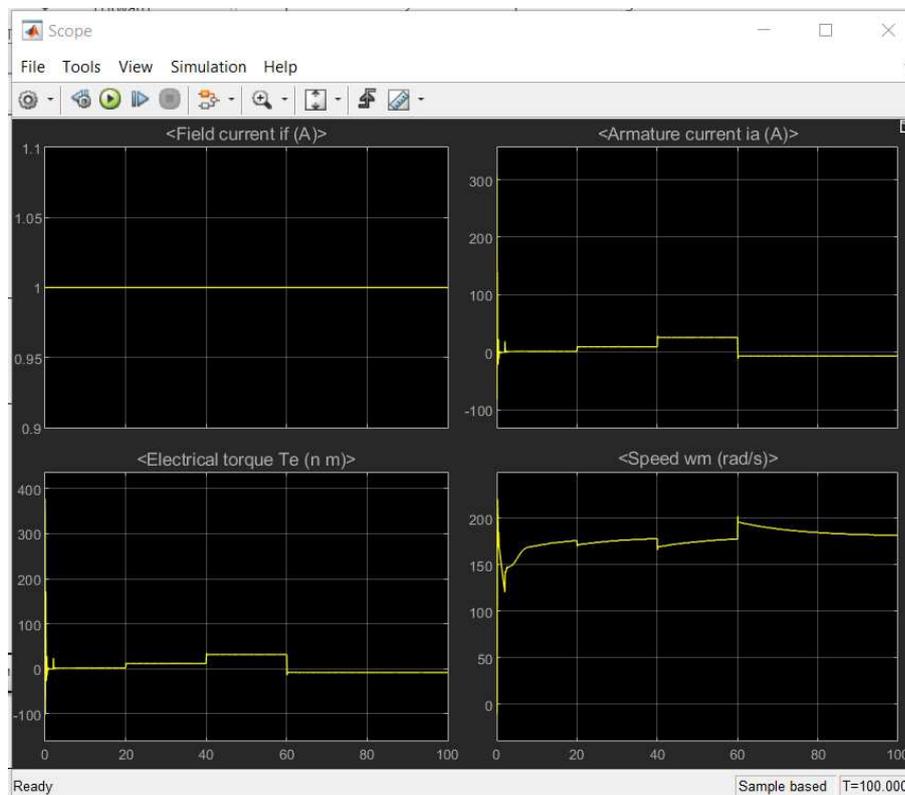
Figura 14 – Planta de motor CC



Fonte: Autoria Própria.

Os resultados utilizando o microcontrolador seguiram semelhante ao modelo simulado (Figura 15).

Figura 15 – Resultados de controle usando microcontrolador



Fonte: Autoria Própria.

5 Considerações Finais

Este relatório apresentou as atividades realizadas durante o Estágio Supervisionado no Laboratório Embedded, como parte integrante da disciplina curricular. Durante o período, o estagiário cumpriu a carga horaria predefinida, realizando as atividades que lhe foram propostas.

As áreas abordadas incluem, sistemas embarcados, Model-Based Design, Hardware In the Loop, envolvendo sistemas de controle para máquinas cc.

Além disso, o uso da ferramenta Embedded Coder permitiu a geração automática de código, facilitando a implementação e depuração dos sistemas de controle no Arduino. Este processo mostrou-se eficiente e reduziu significativamente o tempo de desenvolvimento, além de minimizar erros manuais.

O estágio permitiu ao estagiário desenvolver uma compreensão dos desafios e soluções na área de sistemas embarcados e controle. As disciplinas da graduação foram de grande importância, proporcionando a base de conhecimento necessária para a execução das atividades. Entre elas, destacam-se Controle Digital, Máquinas Elétricas e Acionamentos Elétricos. A experiência prática obtida durante este período será de grande valor para a carreira profissional, preparando o aluno para enfrentar demandas e projetos no mercado de trabalho. A integração de teoria e prática proporcionada pelo estágio é essencial para a formação de engenheiros capacitados e prontos para contribuir de maneira significativa na indústria tecnológica.

Referências

- Arduino.cc ARDUINO.CC. <<https://www.arduino.cc/en/Guide/ArduinoUno>>. Acesso em: 18 jun. 2024. 4
- Arduino.cc 2024 ARDUINO.CC. *Fotografia do Arduino UNO*. 2024. Acessado em: 18 jun. 2024. Disponível em: <<https://store.arduino.cc/products/arduino-uno-rev3>>. 5
- Fitzgerald e Jr. 2003 FITZGERALD, A. E.; JR., C. K. *Máquinas Elétricas*. 5. ed. Porto Alegre: Bookman, 2003. 7
- Han 2015 HAN, S. *Analytical Design of PID Controllers*. 1. ed. Cham: Springer, 2015. 8
- Heath 2002 HEATH, S. *Embedded Systems Design*. [S.l.]: Newnes, 2002. 3
- Jacobina 2005 JACOBINA, C. B. *Sistemas de Acionamento Estático de Máquina Elétrica*. [S.l.], 2005. Disponível em: <<https://pt.scribd.com/document/371652647/Curso-de-Acionamento-JACOBINA>>. 8, 9
- Li e Ang 2006 LI, Y.; ANG, K. H. Pid control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, v. 13, n. 4, p. 559–576, 2006. 7
- MathWorks MATHWORKS. *Simulink Documentation*. <<https://www.mathworks.com/help/simulink/>>. Acesso em: 22 mai. 2024. 7
- MathWorks 2024 MathWorks. *Controlled Voltage Source (Simscape Electrical)*. 2024. Accessed: 03-mai-2024. Disponível em: <https://www.mathworks.com/help/sps/powersys/ref/controlledvoltage.html?s_tid=doc_ta>. 16
- MathWorks 2024 MathWorks. *DC Machine (Simscape Electrical)*. 2024. Accessed: 05-mai-2024. Disponível em: <<https://www.mathworks.com/help/sps/powersys/ref/dcmachine.html>>. 15
- Texas Instruments 2024 Texas Instruments. *LAUNCHXL-F280049C: C2000 F28004x Piccolo LaunchPad Development Kit*. 2024. Accessed: 15-mai-2024. Disponível em: <<https://www.ti.com/product/LAUNCHXL-F280049C/part-details/LAUNCHXL-F280049C>>. 5