

**UNIVERSIDADE DO ESTADO DO AMAZONAS – UEA
ESCOLA SUPERIOR DE TECNOLOGIA – EST
CURSO DE ENGENHARIA ELÉTRICA**

EDWARD PINTO PIMENTA JUNIOR

**ELABORAÇÃO DE AMBIENTES VIRTUAIS PARA AUXILIAR A
AUTOMAÇÃO DE PRÁTICAS LABORATORIAIS DE SISTEMAS DE
CONTROLE EM ENGENHARIA ELÉTRICA E ELETRÔNICA**

Manaus

2023

EDWARD PINTO PIMENTA JUNIOR

**ELABORAÇÃO DE AMBIENTES VIRTUAIS PARA AUXILIAR A
AUTOMAÇÃO DE PRÁTICAS LABORATORIAIS DE SISTEMAS DE
CONTROLE EM ENGENHARIA ELÉTRICA E ELETRÔNICA**

Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia de Engenharia Elétrica da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro Eletricista.

Orientador: Daniel Guzmán Del Río

Manaus

2023

Universidade do Estado do Amazonas - UEA
Escola Superior de Tecnologia - EST

Reitor:

André Luiz Nunes Zodaib

Vice-Reitor:

Kátia do Nascimento Coureiro

Diretor da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo

Coordenador do Curso de Engenharia Elétrica:

Israel Gondres Torné

Banca Avaliadora composta por:

Data da defesa: 28/03/2023.

Prof. Daniel Guzmán Del Río, Dr. (Orientador)

Prof. Antonio Luiz Alencar Pantoja, Dr. (Avaliador 1)

Prof. Israel Gondres Torné, Dr. (Avaliador 2)

CIP – Catalogação na Publicação

Junior, Edward

Ambientes Virtuais Sistemas de Controle/ Edward Pinto
Pimenta Junior; [orientado por] Prof. Orientador: Daniel
Guzmán Del Río, Dr. – Manaus: 2023.
p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica).
Universidade do Estado do Amazonas, 2023.

1. Ambientes Virtuais. 2. MATLAB. 3. Laboratórios. 4. Sistemas de
controle.
I. Guzmán, Daniel.

EDWARD PINTO PIMENTA JUNIOR

**ELABORAÇÃO DE AMBIENTES VIRTUAIS PARA AUXILIAR A
AUTOMAÇÃO DE PRÁTICAS LABORATORIAIS DE SISTEMAS DE
CONTROLE EM ENGENHARIA ELÉTRICA E ELETRÔNICA**

Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia de Engenharia Elétrica da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro Eletricista.

Nota obtida: 9,2

Aprovada em 28 / 03 / 2023

BANCA AVALIADORA:



Prof. Daniel Guzman Del Río, Dr.

Antonio Pantoja

Prof. Antonio Luiz Alencar Pantoja, Dr.



Prof. Israel Gondres Torné, Dr.

Dedicatória

À minha mãe, Maria Cristina, ao meu pai, Edward Pimenta, à minha irmã, Roberta Kelly, minha tia, Katia Maria, por todos os seus esforços, dedicação e investimento em tudo que eu tenho e hoje sou. Por todo o apoio, parceria e cumplicidade ao longo de todos esses anos, eu dedico a vocês essa conquista porque ela é nossa, eu amo vocês.

Agradecimentos

Agradeço aos meus familiares e amigos, por todo o apoio e incentivo. Agradeço meu Orientador Daniel Guzmán Del Río, por ter aceitado meu convite e me orientado. Agradeço a todos meus colegas e professores da EST que me ensinaram e me ajudaram a chegar ao fim de mais uma de muitas conquistas. E por fim, agradeço a Universidade do Estado do Amazonas e a Escola Superior de Tecnologia que me permitiu ter a perspectiva de um futuro melhor.

RESUMO

Devido à não existência de recursos econômicos disponíveis nas faculdades e escolas públicas ou mesmo particulares, ou à falta de instrumentação real para a realização das práticas laboratoriais de forma física, por causa do alto custo que os equipamentos e componentes possuem atualmente, foi idealizado que é possível a realização e elaboração de ambientes virtuais utilizando o MATLAB, para entregar uma solução para a ausência ou dificuldades na execução de práticas laboratoriais, e haver uma ferramenta de aprendizagem, simulação e automação, adequada para a aquisição das habilidades e experiências requeridas no curso de Engenharia Elétrica, Eletrônica, em relação à sistemas de controle, a qual é a área de enfoque do projeto. Além disso, será possível quebrar as barreiras que o ensino à distância pode criar nas práticas laboratoriais.

Palavras-chave: Ambientes Virtuais, MATLAB, Laboratórios, Sistemas de controle.

ABSTRACT

Due to the non-existence of economic resources available in colleges and public or even private schools, or the lack of real instrumentation to carry out laboratory practices in a physical way, because of the high cost that equipment and components currently have, it was idealized that it is possible the realization and elaboration of virtual environments using MATLAB, to deliver a solution for the absence or difficulties in the execution of laboratory practices, and to have a learning, simulation and automation tool, adequate for the acquisition of the skills and experiences required in the course of Electrical Engineering, Electronics, in relation to control systems, which is the focus area of the project. In addition, it will be possible to break down the barriers that distance learning can create in laboratory practices.

Keywords: Virtual Environments, MATLAB, Laboratories, Control systems.

LISTA DE FIGURAS

Figura 1 – Sistema com retroação.....	16
Figura 2 – Modelo Função Transferência	18
Figura 3 – Modelo Controlador PID.....	22
Figura 4 – Valores Método Ziegler e Nichols	22
Figura 5 – Interface inicial App Designer	30
Figura 6 – Página principal do App Designer	31
Figura 7 – “Home Page” do app montada inicialmente	31
Figura 8 – Código Fonte dos botões tipo “run”.....	32
Figura 9 – Código Fonte dos botões tipo “winopen”	32
Figura 10 – Sistema de Controle utilizado no relatório.....	33
Figura 11 – circuito para simulação equação 1.....	34
Figura 12 – Sistema de Controle utilizado no relatório.....	34
Figura 13 – Circuito para simulação equação 1.....	35
Figura 14 – Sistema de nível de líquido	35
Figura 15 – Circuito da questão com entrada degrau	36
Figura 16 – Sistema de nível de líquido questão 1.....	37
Figura 17 – Circuito 1 questão 3	37
Figura 18 – Circuito 2 questão 5	38
Figura 19 – Circuito 1 questão 2	39
Figura 20 – Circuito 2 PID	39
Figura 21 – Tela Inicial.....	41
Figura 22 – Laboratório 1: Função Transferência	42
Figura 23 – Laboratório 2: Espaço-Estado	43
Figura 24 – Laboratório 3: Estabilidade	44
Figura 25 – Laboratório 4: LGR 40.....	45
Figura 26 – Laboratório 5: Controladores PID	46
Figura 27 – Pasta do aplicativo	47
Figura 28 – Opção “Share”	48
Figura 29 – Compilador de Aplicativo.....	48
Figura 30 – Instalação do aplicativo	49

LISTA DE ABREVIATURAS E SIGLAS

EAD – *Ensino à distância*

APP – *Application*

UI – *User Interface*

PID – *Controlador proporcional integral derivativo*

LGR – *Lugar Geométrico das Raízes*

GCF – *Get Current Figure*

GCA – *Get Current Axis*

MEE – *Modelos Espaço-Estado*

FT – *Função Transferência*

VE – *Variáveis de Estado*

ZN – *Ziegler e Nichols*

HTML – *HyperText Markup Language*

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 TEMA	13
1.2 PROBLEMA DE PESQUISA	13
1.3 HIPÓTESE.....	13
1.4 OBJETIVOS	13
1.5 JUSTIFICATIVA.....	14
2 REFERENCIAL TEÓRICO	15
2.1 TIPOS DE SISTEMAS DE CONTROLE	15
2.2 CONCEITOS FUNDAMENTAIS DE SISTEMA DE CONTROLE.....	17
2.3 AMBIENTE VIRTUAL.....	23
3 METODOLOGIA	29
4 IMPLEMENTAÇÃO DO PROJETO.....	30
4.1 CONSTRUÇÃO DO APP	30
4.2 CONSTRUÇÃO DOS LABORATÓRIOS E SEUS ROTEIROS.....	33
4.3 VERSÃO FINAL DO APP E INSTRUÇÕES.....	40
4.4 FINALIZAÇÃO E COMPACTAÇÃO PARA DISTRIBUIÇÃO DO APP	47
5 CONCLUSÃO	50
6 REFERÊNCIAS BIBLIOGRÁFICAS.....	51
7 FONTES CONSULTADAS	52

1 INTRODUÇÃO

Por conta da falta de recursos econômicos das faculdades e escolas públicas ou mesmo particulares, a falta de instrumentação real para a realização das práticas laboratoriais de forma física, tudo isso junto do alto custo que os equipamentos e componentes possuem atualmente, foi proposto que é possível a elaboração de ambientes virtuais utilizando o MATLAB, para entregar uma solução para a ausência ou dificuldades na execução de práticas laboratoriais, e haver uma ferramenta de aprendizagem, simulação e automação, adequada para a aquisição das habilidades e experiências requeridas no curso de Engenharia Elétrica, Eletrônica, em relação à sistemas de controle, a qual é a área de enfoque do projeto.

A principal motivação exposta resolverá problemas presentes do ensino total presencial, podendo assim então, servir de complemento para este, assim como uma nova forma de desenvolver as atividades EAD aplicadas em sistemas de controle, ou seja, o projeto propõe quebrar as barreiras que o ensino à distância pode criar nas práticas laboratoriais.

Os pontos importantes do projeto serão apresentados no trabalho e seus objetivos orbitarão em analisar as formas de desenvolver ambientes e laboratórios virtuais utilizando MATLAB, e desenhar uma plataforma interativa em MATLAB para o desenvolvimento de ambientes virtuais, de maneira prática e direta, o projeto é construir um aplicativo que possa ser disponibilizado nos ambientes citados que farão uso deste, para estudo didático para alunos, assim como ferramenta de utilização para professores.

1.1 TEMA

Elaboração de Ambientes Virtuais utilizando o MATLAB para auxiliar a automação de práticas laboratoriais de sistemas de controle em Engenharia Elétrica e Eletrônica.

1.2 PROBLEMA DE PESQUISA

A não existência de recursos econômicos disponíveis, instrumentação real para a realização das práticas laboratoriais de forma física, devido ao alto custo que os equipamentos e componentes possuem atualmente. Além disso, quebrar as barreiras que o ensino à distância pode criar nas práticas laboratoriais, problema que pode ser visto durante a pandemia de COVID 19.

1.3 HIPÓTESE

É possível a realização e elaboração de ambientes virtuais utilizando o MATLAB, para assim entregar a solução para a ausência ou dificuldades na execução de práticas laboratoriais reais, e então haver uma ferramenta de aprendizagem, simulação e automação, adequada para a aquisição das habilidades e experiências requeridas no curso de Engenharia Elétrica, Eletrônica, em relação à sistemas de controle.

1.4 OBJETIVOS

1.4.1 Objetivo Geral

Criar um ambiente virtual ou laboratório virtual utilizando MATLAB, e ferramentas deste, para auxiliar a automação de práticas laboratoriais de sistemas de controle em Engenharia Elétrica, Eletrônica.

1.4.2 Objetivos Específicos

- a) Analisar as formas de desenvolver ambientes e laboratórios virtuais utilizando MATLAB.
- b) Desenhar uma plataforma interativa em MATLAB para o desenvolvimento de ambientes virtuais.
- c) Desenhar práticas laboratoriais de sistemas de controle em um ambiente virtual.
- d) Utilizar a ferramenta App Designer para a realização do projeto.
- e) Tornar o ambiente virtual intuitivo e de fácil acesso para que assim estudantes e professores consigam trabalhar e desenvolver nos laboratórios virtuais criados.

1.5 JUSTIFICATIVA

O projeto pode ser justificado por apresentar características relevantes socialmente como por exemplo:

- a) Viabilizar a prática laboratorial em faculdades escolas e outros ambientes de ensino que não possuem recursos econômicos disponíveis para tal;

Um dos maiores problemas encontrados atualmente nos ambientes de estudo e ensino é justamente a prática laboratorial, por conta do alto custo de equipamentos e componentes, necessários para uma aula de laboratório, ou até mesmo encontrar um ambiente propício à prática, ou seja, ter um lugar disponível e montar um laboratório ideal tem as suas dificuldades, por isso, mesmo o projeto tendo como enfoque práticas laboratoriais de sistemas de controle, é possível utilizar a plataforma interativa criada e desenvolver práticas pra outras áreas na própria, sendo assim, o projeto tem um amplo campo de utilidade se for desenvolvido além da proposta inicial.

- b) Viabilizar o ensino à distância no que se refere à práticas laboratoriais;

O ensino à distância já era uma alternativa ao ensino muito utilizada, porém depois da pandemia de COVID 19 em que não era permitido aglomeração devido ao alto contágio que o vírus possui, aulas presenciais se tornaram o normal durante o período, logo, o projeto provê uma alternativa real para o EAD (ensino à distância), para práticas laboratoriais de sistemas de controle e, como explicado anteriormente, uma possível alternativa de modo geral para práticas em laboratórios.

Assim como o projeto tem relevância social, também possui relevância acadêmica como:

- a) Conclusão do curso e obtenção do diploma de engenheiro eletricitista;
- b) Aprimorar os conhecimentos com a ferramenta MATLAB e na área de sistemas de controle;
A aplicação do projeto de certo ajudará na expansão de conhecimento e ferramentas da plataforma MATLAB.
- c) Auxílio na automação de práticas laboratoriais de sistemas de controle para alunos e professores;
O projeto será de grande importância para professores de sistemas de controle aplicarem suas práticas e alunos também aplicarem os seus conhecimentos em sistemas de controle seguindo roteiros de laboratórios de controle.

2 REFERENCIAL TEÓRICO

A ideia original é criar um app, que possa ser um ambiente virtual interativo, e deste ambiente serão exibidos os laboratórios com seus roteiros já prontos, os elementos e componentes serão selecionados e utilizados na simulação da prática laboratorial. Para poder montar os roteiros dos laboratórios, é necessário um punhado de conceitos de engenharia de controle e sistemas de controle. Por isso serão abordados alguns conceitos básicos, que serão o esqueleto fundamental das práticas de laboratórios criados, e a partir destes, feita a simulação quando o usuário a quiser iniciar. A maior parte dos conceitos mostrados nessa seção são do autor Katsuhito Ogata do livro “Engenharia de Controle Moderno”, sua 5ª edição, sendo sua 1ª edição originalmente publicada em 1997, mas também há citações e referências de outros autores e livros.

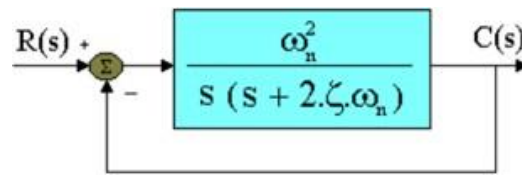
2.1 TIPOS DE SISTEMAS DE CONTROLE

Nesta seção será feito um resumo das características de alguns dos principais tipos de sistemas de controle existentes e que serão trabalhados durante o desenvolvimento das práticas laboratoriais.

2.1.1 Sistemas de Controle com Retroação

Um sistema que mantém uma relação direta entre a grandeza de saída e a grandeza de referência, fazendo uma comparação entre essas grandezas e utilizando diferença delas para fazer o controle. Um exemplo disso seria o sistema de controle de temperatura de um ambiente. Medindo-se a temperatura do ambiente e comparando-a com a temperatura de referência (temperatura desejada), o termostato aciona o equipamento de calefação ou de refrigeração, ligando ou desligando cada um deles, de tal sorte que a temperatura do ambiente permaneça na faixa de conforto estabelecida, a despeito das condições externas (OGATA, 1997, p.5). Os sistemas de controle com retroação são frequentemente referidos como sistemas de controle a malha fechada. Na prática, os termos, controle com retroação e controle a malha fechada são usados indistintamente. Num sistema de controle a malha fechada, o sinal atuante de erro, que é a diferença entre o sinal de entrada e o sinal de retroação (que pode ser o próprio sinal de saída ou uma função do sinal de saída e de suas derivadas e integrais), excita o controlador de modo a reduzir o erro e trazer o valor do sinal de saída para o valor desejado. A expressão controle a malha fechada acarreta sempre o uso da retroação a fim de reduzir o erro do sistema (OGATA, 1997, p.5).

Figura 1 - Sistema com retroação.



Fonte: K. Ogata

Apesar de os sistemas de controle que serão trabalhados no projeto serem, na maior parte, sistemas de malha fechada, é importante também saber como funcionam os sistemas a malha aberta. Sistemas a malha aberta são sistemas nos quais o sinal de saída não afeta a ação de controle são chamados sistemas de controle a malha aberta. Em outras palavras, num sistema de controle a malha aberta, não se mede o sinal de saída nem tampouco este sinal é enviado de volta para comparação com o sinal de entrada. Um exemplo prático disto é o da máquina de lavar roupa. As operações de colocar de molho, lavar e enxaguar numa lavadora são executadas numa sequência programada em função do tempo. A máquina não mede o sinal de saída, isto é, a limpeza das roupas.

Nos sistemas de controle a malha aberta o sinal de saída não é comparado com o sinal de referência na entrada. Assim, a cada sinal de referência na entrada corresponde uma condição de operação fixa: como resultado, a exatidão do sistema depende de uma calibração. Na presença de distúrbios, os sistemas de controle a malha aberta não desempenham a tarefa desejada. Na prática, os sistemas de controle a malha aberta são usados somente quando as relações entre entrada e saída do processo a controlar forem conhecidas e quando não existirem distúrbios internos e externos (OGATA, 1997, p.5).

2.1.2 Sistemas lineares e não-lineares

A maior parte dos sistemas que serão simulados nos laboratórios os chamados sistemas lineares. Um sistema é chamado linear se a ele se aplica o princípio da superposição. O princípio da superposição estabelece que a resposta produzida pela aplicação simultânea de duas excitações diferentes é igual à soma das duas respostas individuais a cada uma das excitações. Conseqüentemente, para os sistemas lineares, a resposta para várias entradas pode ser calculada considerando-se uma única entrada de cada vez e adicionando-se os resultados. É este princípio que permite a construção de soluções complicadas para equações diferenciais lineares a partir de soluções simples. Em uma investigação experimental de um sistema dinâmico, se causa e efeito forem proporcionais, levando, assim, a que o princípio da superposição seja aplicável, então o sistema pode ser considerado linear (OGATA, 1997, p.49).

Uma equação diferencial é linear se os coeficientes forem constantes ou funções apenas da variável independente. Sistemas dinâmicos que são compostos de componentes lineares a parâmetros concentrados e invariantes no tempo podem ser descritos por equações diferenciais lineares (a coeficientes constantes) e invariantes no tempo. Tais sistemas são chamados sistemas lineares invariantes no tempo (ou lineares a coeficientes constantes). Os sistemas representados por equações diferenciais cujos coeficientes sejam funções do tempo são chamados sistemas lineares variáveis no tempo (OGATA, 1997, p.49).

Já um sistema é não-linear se a ele não se aplica o princípio da superposição. Assim, nos sistemas não-lineares a resposta a duas entradas não pode ser calculada tratando-se uma entrada de cada vez e adicionando-se os resultados (OGATA, 1997, p.49).

Em teoria de controle, as funções ditas funções de transferência são comumente usadas para caracterizar as relações de entrada-saída de componentes ou sistemas que podem ser descritos por equações diferenciais lineares invariantes no tempo (OGATA, 1997, p.50).

2.2 CONCEITOS FUNDAMENTAIS DE SISTEMA DE CONTROLE

Outros conceitos fundamentais são o de função transferência, modelos de diagramas de blocos, a teoria de controle moderno, funções e modelos espaço-estado, método do lugar das raízes, estabilidade de um sistema e controladores PID, os quais serão abordados a seguir.

2.2.1 Função de transferência

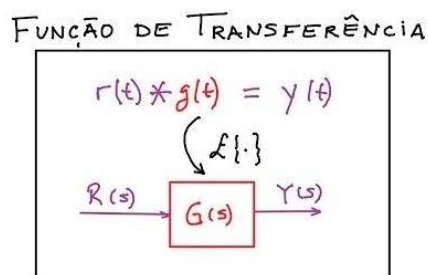
Funções transferências (FT) são a base das simulações nas práticas laboratoriais, sendo assim, conhecer o conceito de função de transferência é fundamental. A função de transferência de um sistema representado por equações diferenciais lineares invariantes no tempo é definida como a relação entre a transformada de Laplace do sinal de saída (função resposta) e a transformada de Laplace do sinal de entrada (função excitação), na hipótese de que todas as condições iniciais são nulas. A aplicabilidade do conceito da função de transferência é limitada a equações diferenciais lineares invariantes no tempo. O método das funções de transferência, portanto, é amplamente usado na análise e no projeto de tais sistemas. (OGATA, 1997, p.50). A função de transferência de um sistema é um modelo matemático no sentido de que constitui um método operacional de expressar a equação diferencial que relaciona a variável de saída à variável de entrada (OGATA, 1997, p.50).

$$G(s) = \frac{L[\text{saída}]}{L[\text{entrada}]} \text{ condições iniciais nulas. Equação (2.2.1)}$$

A função de transferência é uma propriedade intrínseca do sistema, independentemente da magnitude e da natureza do sinal de entrada ou função de excitação (OGATA, 1997, p.50). A função de transferência inclui as unidades necessárias para relacionar o sinal de entrada ao sinal de saída: no entanto, ela não fornece qualquer informação concernente à estrutura física do sistema. (As funções de transferência de muitos sistemas fisicamente diferentes podem ser idênticas) (OGATA, 1997, p.51). Se a função de transferência de um sistema for conhecida, a saída ou resposta pode ser estudada para várias formas de entradas com vistas ao entendimento da natureza do sistema (OGATA, 1997, p.51).

Se a função de transferência de um sistema for desconhecida, ela pode ser estabelecida experimentalmente introduzindo-se sinais de entrada conhecidos e estudando-se o sinal de saída do sistema. Uma vez estabelecida, a função de transferência fornece uma descrição completa das características dinâmicas do sistema, tão precisas quanto aquelas obtidas a partir de sua descrição física (OGATA, 1997, p.51).

Figura 2 - Modelo Função Transferência.



Fonte: Luis Antonio Aguirre - Youtube

2.2.2 Diagrama de blocos de um sistema a malha fechada

Um modelo que representa muito bem um sistema é o diagrama de blocos, principalmente sistemas a malha fechada, nos diagramas de blocos a saída é geralmente representada por um sinal qualquer $C(s)$ e o sinal de entrada por $R(s)$, e a função transferência por $G(s)$. A saída $C(s)$ retroage ao ponto de soma, onde é comparada com o sinal de entrada de referência $R(s)$. O sinal de saída do bloco, $C(s)$ neste caso, é obtido pela multiplicação da função de transferência $G(s)$ pelo sinal de entrada no bloco. Qualquer sistema de controle linear pode ser representado por um diagrama de blocos constituído por blocos, pontos de soma e pontos de junção. Quando o sinal de saída retroage ao ponto de soma para comparação com a entrada, é necessário converter sua natureza física na mesma natureza do sinal de

entrada. Por exemplo, em um sistema de controle de temperatura, o sinal de saída é usualmente a temperatura controlada. O sinal de saída, que tem a dimensão de temperatura, deve ser codificado como força, posição ou tensão elétrica. O papel do elemento de retroação é modificar a natureza do sinal de saída antes que este seja comparado com o sinal de entrada. (Na maioria dos casos, o elemento de retroação é um sensor que mede o valor da grandeza de saída do processo a controlar. O sinal de saída do sensor é comparado com o sinal de entrada, gerando-se o sinal de erro atuante) (OGATA, 1997, p.53).

2.2.3 Teoria de controle moderno

A tendência atual dos sistemas de engenharia é no sentido de aumentar sua complexidade em função, principalmente, da necessidade de realizar tarefas complexas e com requisitos de boa precisão. Sistemas complexos podem ter muitas entradas e muitas saídas e podem ser variantes no tempo. A necessidade de satisfazer requisitos cada vez mais rigorosos quanto ao desempenho de sistemas de controle, o aumento de complexidade dos sistemas e a facilidade de acesso aos computadores de grande porte ensejaram o desenvolvimento da teoria de controle moderno, iniciada por volta de 1960, como uma nova forma de analisar e projetar sistemas de controle complexos (OGATA, 1997, p.58).

Esta nova abordagem é baseada no conceito de estado. A teoria de controle moderno contrasta com a teoria de controle convencional no sentido de que a primeira é aplicável a sistemas com entradas e saídas múltiplas, lineares ou não-lineares, variantes ou invariantes no tempo, enquanto a última é aplicável apenas aos sistemas monovariáveis (uma única entrada e uma única saída), lineares e invariantes no tempo. Além disso, a teoria de controle moderno é uma abordagem centrada essencialmente no domínio do tempo, enquanto a teoria de controle convencional adota um enfoque no domínio de frequência complexa. Antes de se prosseguir, devem ser definidos os termos estado, variáveis de estado, vetor de estado e espaço de estados (OGATA, 1997, p.58).

2.2.4 Modelos Espaço-Estado

Os modelos espaço-estado, são outra forma de representar um sistema através das equações espaço-estado e suas matrizes (MEE). O estado de um sistema dinâmico é o menor conjunto de valores de variáveis (chamadas variáveis de estado) de modo que o conhecimento destes valores em junto com o conhecimento dos valores do sinal de entrada para determina completamente o comportamento do sistema em qualquer instante. As variáveis de estado (VE) de um sistema dinâmico são as grandezas cujo conjunto de valores determina o estado do sistema. As variáveis de estado não precisam ser

grandezas fisicamente mensuráveis ou observáveis (OGATA, 1997, p.58). Variáveis que não representam grandezas físicas e aquelas que não são nem mensuráveis nem observáveis podem ser escolhidas como variáveis de estado.

Tal liberdade na escolha das variáveis de estado é uma vantagem dos métodos de espaço de estados. Falando em termos práticos, no entanto, é conveniente escolher grandezas facilmente mensuráveis para variáveis de estado, se isto for de todo possível, pois as leis de controle ótimo requerem a retroação de todas as variáveis de estado com ponderação adequada (OGATA, 1997, p.58).

O espaço n-dimensional cujos eixos coordenados consistem em n-eixos, é então chamado espaço de estados, sendo assim, qualquer estado pode ser representado por um ponto no espaço de estados.

2.2.5 Método do lugar das raízes (LGR)

O desempenho de um sistema com retroação pode ser descrito em função da localização das raízes características em dado plano “s” genérico.

Sabe-se que a resposta de um sistema com retroação a malha fechada pode ser ajustada para se alcançar o desempenho desejado através da seleção judiciosa de um ou mais parâmetros do sistema. É muito útil determinar como as raízes da equação característica se deslocam ao longo do plano s à medida que se varia o valor de um parâmetro. (BISHOP;DORF, 1998, p.262).

O lugar das raízes no plano “s” pode ser determinado por meio de um método gráfico. O gráfico do lugar das raízes à medida que varia um parâmetro do sistema é conhecido como o diagrama do lugar das raízes. O lugar das raízes é uma ferramenta poderosa para se analisar e projetar sistemas de controle com retroação (BISHOP;DORF, 1998, p.262).

A estabilidade relativa e o desempenho de um sistema de controle a malha fechada estão diretamente relacionados com a localização das raízes da equação característica a malha fechada no plano “s”. Frequentemente é necessário ajustar um ou mais parâmetros a fim de se obter o posicionamento adequado das raízes. (BISHOP;DORF, 1998, p.262). Portanto, vale a pena determinar como as raízes da equação característica de um dado sistema se deslocam pelo plano “s” à medida que os parâmetros mudam de valor; ou seja, é útil determinar o lugar das raízes no plano “s” à medida que um parâmetro tenha o seu valor modificado. O método do lugar das raízes foi introduzido por Evans em 1948 e tem sido desenvolvido e utilizado extensamente na prática da engenharia de controle. A técnica do lugar das raízes é um método gráfico de esboçar, no plano “s”, o lugar geométrico das raízes à medida que um parâmetro é variado. De fato, o método do lugar das raízes fornece ao engenheiro uma medida da sensibilidade das raízes do sistema a uma variação de um parâmetro sob consideração. A técnica do

lugar das raízes pode ser usada com grande vantagem em conjunto com o critério de Routh-Hurwitz. O método do lugar das raízes fornece informação gráfica, por conseguinte, pode-se usar um esboço aproximado no sentido de obter informação qualitativa referente à estabilidade e ao desempenho do sistema. Além disto, pode-se investigar o lugar das raízes da equação característica de um sistema com múltiplas malhas tão rapidamente quanto de um sistema com uma única malha. Se a localização das raízes não for satisfatória, os ajustes de parâmetro necessários quase sempre podem ser pronta mente determinados com exatidão a partir do lugar das raízes. (BISHOP;DORF, 1998, p.263).

2.2.6 Estabilidade de um sistema

Ao se considerar o projeto e a análise de sistemas de controle com retroação, a estabilidade é da maior importância. De um ponto de vista prático, um sistema de controle a malha fechada que seja instável é de pouco valor. Muitos sistemas físicos são inerentemente instáveis a malha aberta. Com o uso da retroação é possível estabilizar processos instáveis e, assim, mediante a escolha judiciosa de valores de parâmetros, ajustar o desempenho em regime transitório.

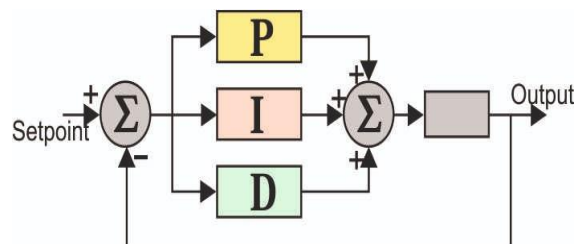
Para processos estáveis a malha aberta utiliza-se ainda a retroação para ajustar o desempenho a malha fechada de modo a alcançar as especificações de projeto. Estas especificações tomam a forma de erros de acompanhamento em estado estacionário, ultrapassagem percentual, tempo de assentamento, tempo de pico entre outros índices (BISHOP;DORF, 1998, p.229).

Pode-se dizer que um sistema com retroação a malha fechada é estável ou não. Este tipo de caracterização estável/não estável é referida como estabilidade absoluta. Um sistema que possua estabilidade absoluta é dito um sistema estável - abandonando-se o rótulo absoluto. Dado um sistema a malha fechada que seja estável, é possível caracterizar adicionalmente o grau de estabilidade. Isto é referido como estabilidade relativa. (BISHOP;DORF, 1998, p.230). Um sistema estável é definido como um sistema com resposta limitada (sem exceder valores de magnitude limitados). Isto é, se o sistema for submetido a uma excitação limitada ou a uma perturbação e a resposta for limitada em magnitude, o sistema é dito estável (BISHOP;DORF, 1998, p.230). Um sistema estável é então, um sistema dinâmico com uma resposta limitada a uma entrada limitada (BISHOP;DORF, 1998, p.230).

2.2.7 Controle PID de um processo

Uma forma de controlador usado amplamente no controle de processos industriais, é chamado de controlador de três termos ou controlador PID. O controlador oferece um termo proporcional, um termo de integração e um termo derivativo (BISHOP;DORF, 1998, p.289).

Figura 4 - Modelo Controlador PID



Fonte: pcbheaven.com/wikipages/PID_Theory

Ziegler e Nichols propuseram regras para se determinarem os valores do ganho proporcional K , do tempo de integral I , e do tempo derivativo T , com base nas características da resposta transitória de um determinado processo a controlar. Tal determinação de parâmetros, ou sintonia, dos controladores PID pode ser feita na própria instalação através de experimentos sobre o processo. A partir da proposta de ZN, um grande número de outras regras para a sintonia de controladores PID tem sido surgido (OGATA, 1997, p.545).

Figura 4 – Valores Método Ziegler e Nichols.

Controller Type	From step response			From frequency response		
	K_p	T_i	T_d	K_p	T_i	T_d
P	$1/a$			$0.5K_c$		
PI	$0.9/a$	$3L$		$0.4K_c$	$0.8T_c$	
PID	$1.2/a$	$2L$	$L/2$	$0.6K_c$	$0.5T_c$	$0.12T_c$

Fonte: K. Ogata

2.3 AMBIENTE VIRTUAL

Para a construção do ambiente virtual proposto e baseado nas teorias de sistemas de controle já apresentados, será necessário um conhecimento básico de como utilizar a plataforma App Designer dentro do MATLAB para a construção do app que será o ambiente virtual proposto que conterà os laboratórios, roteiros e componentes dentro deste. Para isso, foi utilizado como base para este trabalho de pesquisa, o documento guia para a criação de app utilizando o App Designer do MATLAB da Mathworks, a empresa proprietária e criadora do MATLAB, a seguir tem-se então, um resumo de alguns conceitos considerados importantes para a aplicação do projeto, conceitos e noções chave da ferramenta App Designer para a criação de apps, os quais foram primariamente retirados do guia “creating_guis: MATLAB® App Building” da Mathworks.

2.3.1 Construção de apps com o App Designer do MATLAB

Existem pelo menos duas maneiras de usar o MATLAB para a construção de interfaces a nível usuário. Os Apps são interfaces que performam operações baseadas nas interações de usuário. Podem ser construídos interativamente utilizando o App Designer, com formato de arquivo “m.lapp”, sendo que as opções de compartilhamento variam entre distribuir a interface principal ,suportar arquivos diretamente, possui pacotes em um único arquivo, pode rodar como um web app em qualquer web browser, mas nesse caso necessita de um compilador de MATLAB, pode também ser criado uma aplicação de desktop, que também precisará de um compilador de MATLAB. Apps podem ser também construídos através de programação, usando funções de MATLAB, com o tipo de arquivo sendo “.m (tipo utilizado para um script MATLAB, função ou classe)”, e as opções de compartilhamento variam entre distribuir a interface principal e suportar arquivos diretamente, pacotes em um único arquivo, pode ser criado uma aplicação de desktop, que precisará de um compilador de MATLAB (MATLAB® App Building, 2000-2022, p.21, tradução nossa).

Já as tarefas de editor são interfaces que podem ser embutidas em um script ao vivo e que podem ser geradas ao mesmo tempo que o usuário explora os parâmetros. Podem ser construídos através de programação usando a classe base “matlab.task.LiveTask”. O tipo de arquivo sendo “.m (classe MATLAB)”, e a opções de compartilhamento são a distribuição do arquivo da interface principal e suporte direto de arquivos. (MATLAB® App Building, 2000-2022, p.21, tradução nossa). O que se quer construir neste projeto são os apps, um app simulando um laboratório, então a pesquisa de projeto se baseia primeiramente em como se construir um app utilizando o MATLAB, para depois serem aplicados os conceitos de sistemas de controle e os laboratórios serem construídos.

A principal ferramenta do projeto é o App Designer, App Designer é um rico ambiente de interação, introduzido no MATLAB na versão R2016a, principal ambiente recomendado para a construção de apps utilizando MATLAB. Ela inclui uma versão totalmente integrada do MATLAB Editor. O design do layout e as visualizações de códigos são extremamente conectadas, de tal forma que as mudanças que são feitas em um ponto imediatamente afetam outro ponto (MATLAB® App Building, 2000-2022, p.21, tradução nossa).

2.3.2 Criação de um simples app usando o App Designer

App Designer provê um tutorial que guia durante o processo de criação de um simples app contendo um “plot” e um “slider”. O slider controla a amplitude da função plotada. O tutorial inicial explica alguns conceitos básicos, porém, importantíssimos para o objetivo desta pesquisa, então alguns dos passos deste tutorial serão listados e explicados, para assim, se poder ter um primeiro contato e noção de como o laboratório virtual será construído e também saber como funciona o App Designer.

Primeiramente, para rodar o tutorial é necessário abrir a página inicial do App Designer e clicar em “Show Examples” na seção de “Apps”. E então selecionar a opção “Interactive Tutorial”. O App Designer tem duas seções para criação de apps: “Design View” e “Code View”. O “Design View” é usado para criar componentes de interface de usuário e interativamente fazer o app. Já o “Code View” é usado para programar como o app irá se comportar. Pode ser trocado de uma aba pra outra usando os botões na parte superior direita do App Designer. Para criar uma “Axes Component” ou Componente de Eixo, é necessário clicar em “Design View”, criar componentes de interface de usuário e então modificar sua aparência interativamente. A biblioteca de componentes ou “Component Library” contém todos os componentes, caixas, e ferramentas que podem ser acrescentados no app interativamente.

Os componentes podem ser arrastados da “Component Library” na aba do app. Então a aparência pode ser modificada juntamente com as propriedades em “Component Browser”, ou editando alguns certos aspectos da componente, como por exemplo o tamanho do componente ou texto que pode ser inserido. O “slider” pode também ser tirado da mesma biblioteca de componentes. Pode-se então verificar como fica o Design final no “Design View” A navegação feita no “Code View” proporciona a programação direta do app.

Quando os componentes são adicionados ao app no “Design View”, o App Designer automaticamente gera o código que irá ser executado quando o app for rodado. Este código configura a aparência do app pra ficar igual ao que é visto na aba de visualização, este código não é editável e é mostrado num background cinza.

Como parte deste código gerado, o App Design cria alguns objetos de programação para serem usados quando o comportamento do app estiver sendo desenvolvido. Será necessário conhecer quais são os principais tipos de “Objetos” que são trabalhados no App Designer. Existe o objeto de app, este objeto guarda todos os dados do app criado, como por exemplo os componentes de interface de usuário e qualquer dado que foi especificado nas suas propriedades. Todas as funções do app que requerem esse objeto como primeiro argumento. Esse padrão permite que o acesso aos componentes sejam garantidos, assim como as propriedades dentro das funções. (MATLAB® App Building, 2000-2022, p.32, tradução nossa).

Outro tipo de objeto são os “objetos de componentes” (Component Object), a qualquer momento que seja necessário adicionar um componente no “Design View”, o App Designer guarda o componente como um objeto nomeado usando a forma “app.ComponentName”. Este nome de componente pode ser visto e modificado no “Component Browser” ou Navegador de Componentes. Pra acessar e atualizar as propriedades dos componentes de dentro do código de app, será necessário usar o padrão “app.ComponentName.Property”. (MATLAB® App Building, 2000-2022, p.32, tradução nossa).

O comportamento do app pode ser primariamente programado usando funções “callback” ou funções de retorno, que é uma função dentro de outra função como argumento, pode-se chamar a função interna através da externa quando for preciso. Pode ser acrescentada então uma função que executa sempre que o usuário ajusta o valor do slider, então uma função dentro de outra. Quando uma função de retorno é acrescentada a um componente, o App Designer posiciona o cursor no corpo da função, então o objeto de app passa como argumento primário da função “callback”, isto serve para permitir o acesso de componentes e suas propriedades. Quando uma “função de gráfico” é chamada no App Designer, é necessário especificar o rótulo ou “label” do gráfico ou o objeto como argumento da função. Pra rodar o app é necessário clicar em “Run”, ele então será salvo e rodado logo em seguida. Após salvar as mudanças, o app fica pronto para rodar novamente no App Designer, ou simplesmente digitando seu nome, sem a extensão “.mlapp”, na janela de comando do MATLAB. Quando o app for rodado no prompt de comando, o arquivo deve estar na pasta atual ou no caminho próprio do MATLAB.

2.3.3 Mostrar Gráficos no App Designer

Nesta seção será explicado como exibir gráficos resultantes, que serão amplamente utilizados na criação do app, e serão as respostas visuais das simulações feitas no laboratório virtual. É preciso saber que muitas das funções que geram gráfico, ou tipicamente chamadas de “funções de gráfico”, no MATLAB, tem um argumento pra especificar os eixos alvos, conhecidos como “target axes” e até mesmo tem como característica o parentesco entre objetos, grupos de objetos chamados “parent objects”,

que possuem uma propriedade em comum, e podem ser chamados de uma vez indicando tal característica. Esse argumento é opcional na maioria das vezes, mas quando essas funções são chamadas no App Designer, esse argumento deve ser especificado.

A razão para isso é que na maioria dos casos, o MATLAB usa por padrão funções “gcf” ou “gca”, (grupos de funções bases do MATLAB para gerar figuras), pra pegar o objeto alvo e utilizá-lo em uma operação. Mas essas funções dependem da propriedade “HandleVisibility” (uma propriedade própria do MATLAB para objetos gráficos), dependem do “HandleVisibility” da figura estando “On”, porém no App Designer, esta propriedade, vem por padrão com o estado “Off”, isso significa que as funções “gcf” e “gca” não funcionam normalmente, como resultado, omitir “target axes” e “parente object” pode produzir resultados inesperados.

Há várias maneiras de especificar um “target component” ou “Componente Alvo” para funções de gráfico. Pode-se especificar eixos como primeiro argumento, muitas funções de gráfico tem uma primeira entrada opcional pra especificar os eixos alvos, por exemplo tanto a função “plot” e a como a função “hold” tomam um objeto de “target axes” dessa maneira. Pra plotar duas linhas com um conjunto de eixos na aba de design, é necessário especificar o nome dos objetos eixos como o primeiro argumento.

2.3.4 Construindo Componentes de App

O App Designer e as suas figuras de interface de usuário suportam um largo set de componentes pra apps, com todas as funcionalidades possíveis. A lista dos componentes disponíveis pode ser vista abaixo:

- 1) Componentes Comuns – incluem componentes que respondem à interações, como botões, “sliders”, abas de deslizamento (pra cima ou pra baixo) e árvores ou “trees”;
- 2) Eixos ou “Axes” – Incluem eixos pra gerar visualizadores de dados e exploração;
- 3) Caixas (Containers) e ferramentas de figura – Incluem painéis e tabelas pra agrupar componentes, também barras de menus;
- 4) Componentes de Instrumentação – Incluem funcionalidades pra visualizar status, também chaves ou “switches” pra selecionar parâmetros de entrada;
- 5) Componentes extensíveis – Incluem “toolboxes” de componentes de interface de usuário de outros autores, requer então uma licença de “toolbox” extra e sua instalação;
- 6) Todos os componentes estão disponíveis para programação. A maioria dos componentes estão na biblioteca de componentes do App Designer, pra arrastar e serem adicionados na aba de design. (MATLAB® App Building, 2000-2022, p.50, tradução nossa).

2.3.5 Layout do Aplicativo

A aba “Design View” no App Designer provê um rico conjunto de ferramentas de “layout” para apps a serem criados. Também provê uma extensa biblioteca de componentes de interface de usuários, então é possível criar funcionalidades de maneira interativa. Qualquer mudança que é feita no “Design View” é automaticamente refletida no “Code View”. Então é possível configuração de vários aspectos do app sem nem sequer escrever qualquer código. (MATLAB® App Building, 2000-2022, p.76, tradução nossa).

Pra adicionar um componente ao app, existem dois métodos possíveis, tem como arrastar um componente da biblioteca e colocar direto na aba de design. A outra opção é clicar no componente na biblioteca de componentes e mover o cursor sob a aba de design, o cursor muda pro formato de uma cruz. Então é possível com o mouse para adicionar o componente à aba de design no tamanho padrão, ou clicar, arrastar e determinar o tamanho do componente. Alguns componentes podem ser adicionados em seu tamanho padrão. (MATLAB® App Building, 2000-2022, p.76, tradução nossa).

O nome do componente então deve aparecer no “Component Browser” depois que é adicionado na aba de design. Pode-se selecionar componentes tanto na aba de design quanto no “Component Browser”, esta seleção ocorre em ambos os lugares simultaneamente.

Alguns componentes, como campos de edição ou “sliders”, são agrupados por rótulos, ou as chamadas “labels”, quando eles são arrastados pra aba de design, esses rótulos não aparecem no “Component Browser” por padrão, mas é possível adicioná-los à lista ao clicar em qualquer lugar no “Component Browser” e selecionando a opção “Include component labels in Component Browser”.

Se o usuário não quiser que o componente tenha um rótulo, pode-se excluí-lo pressionando e segurando a tecla “Ctrl” ao arrastar o componente para a aba de design. Se um componente tem um rótulo, e o texto do rótulo é alterado, o nome do componente no “Component Browser” muda pra se alinhar com o texto alterado, A customização do nome do componente ocorre dando um duplo clique e digitando o novo nome na aba de texto. A customização da aparência dos componentes pode ser feita selecionando o item e editando suas propriedades na aba de componente do “Component Browser”, pode ser tomado como exemplo a aba “Button”, nela é possível alterar o alinhamento do texto que é mostrado em um botão.

2.3.6 Programação no App

A seção “Code View” do App Designer, provê a maior parte do que as funcionalidades de programação que o “MATLAB editor” já possui. Também dispõe de um rico conjunto de

funcionalidades que ajudam na navegação pelo código e evitam tarefas desnecessárias. Por exemplo, se for preciso procurar uma função “callback”, pode-se digitar as primeiras letras do nome na barra de procura. Clicando em “Search”, nela será possível encontrar a definição daquele “callback” específico, se o nome deste “call-back” for alterado, o App Designer automaticamente alterará todas as referências no próprio código.

O Code View tem três painéis que ajudam no gerenciamento de diferentes aspectos do código principal. O “Component Browser”, o “Code Browser” e “App Layout”. O “Component Browser” tem quatro partes principais, o “Context Menu”, nele é possível clicar em um componente da lista pra mostrar um “menu de contexto”, menu que tem opções para deletar um componente específico ou renomeá-lo, adicionar um “callback”, ou mostrar a aba de ajuda. Se selecionar a opção “Include Component Labels in Component Browser”, ou em português, “incluir um rótulo de componente no Navegador de Componentes”, é possível mostrar um grupo de rótulo de componentes.

A “Search Bar”, ou barra de procura, serve para se encontrar rapidamente um componente digitando suas iniciais na barra. A “Aba de Componente” é utilizada para ver ou alterar propriedades dos componentes que estão selecionados, procura pela propriedade pode ser feita digitando o nome na barra de procura no topo desta aba. E por último existe a “Aba de Callbacks”, através desta aba é possível gerenciar call-backs para componentes que estão selecionados (MATLAB® App Building, 2000-2022, p.96, tradução nossa).

O “Code Browser” ou “Navegador de Código” tem duas partes principais, as “Abas de Callbacks, Funções e Propriedades”, que são usadas para adicionar, deletar, renomear um dos callbacks, funções de ajuda, ou customizar propriedades do app. Clicando em um item nas abas de “Callbacks” ou “Functions” leva até a seção correspondente no código. Pode-se alterar a ordem dos callbacks selecionando o callback que será movido e então selecionar, arrastar e colocá-lo na sua nova posição na lista. E a “Search Bar”, ou barra de procura, que serve pra rapidamente localizar um “callback”, uma função de ajuda, ou propriedade digitando as iniciais do nome na barra. (MATLAB® App Building, 2000-2022, p.96, tradução nossa). E por último, existe o “App Layout”, o qual tem a funcionalidade “App thumbnail”, que é o uso de uma imagem “thumbnail” ou ícone pra localizar componentes em apps largos e complexos que tem vários componentes. Selecionando um componente na pela “thumbnail” seleciona automaticamente o componente no “Component Browser”. (MATLAB® App Building, 2000-2022, p.97, tradução nossa). Uma observação importante é que no editor do “Code View”, algumas seções do código são editáveis e algumas não são. Seções que não são editáveis são geradas e gerenciadas pelo próprio App Designer, onde quer que seção corresponda, pode ser o corpo das funções que são definidas (callbacks e funções de ajuda), ou definições de propriedade customizadas. Então seguindo o esquema padrão de cor, seções não editáveis do código são cinzas e seções editáveis são brancas.

3 METODOLOGIA

O presente trabalho é uma pesquisa aplicada, e o objetivo deste, é a realização de uma pesquisa exploratória e explicativa do material fonte do assunto e após isso a aplicação direta dos conhecimentos adquiridos. Os procedimentos técnicos que foram utilizados são os de pesquisa bibliográfica e experimental. O método de abordagem utilizado foi o hipotético-dedutivo e a elaboração seguiu o método de procedimento monográfico. Inicialmente, foram realizadas pesquisas bibliográficas na área de Sistemas de Controle aplicado à engenharia elétrica, foram então abordados os assuntos relacionados à funções transferência, modelos e equações espaço-estados, sistemas de primeira e segunda ordem, estabilidade de sistemas de controle, sistemas LGR e controladores PID, entre outros.

Além disso, foi também abordado como se utiliza a plataforma App Designer dentro do MATLAB, grande parte da pesquisa foi o aprendizado de como se utilizou a ferramenta para criar um app dentro do MATLAB.

A primeira etapa de aplicação da pesquisa, ou seja, aplicação do projeto foi a construção de um app que foi então a base para o ambiente virtual que se quis criar, o ambiente virtual é a “home page” do app, que contém uma aba com os laboratórios que se desejará simular, ou seja, primeiramente foi necessário construir no app o ambiente inicial que aparece quando se entra no app, logo após isso, foi necessário criar também a aba com as opções de laboratórios que podem ser simulados, sendo, a quantidade planejada de laboratórios, a de 5 simulações que giram em torno de conceitos que foram abordados neste trabalho.

A segunda etapa então foi a construção interna dos laboratórios inicialmente exibidos como opção, então quando o laboratório escolhido é selecionado, abrirá uma nova interface, que apresentará um passo a passo do que se fazer na prática, o roteiro de laboratório, e indicativos de como interagir com a plataforma, quais elementos ou componentes que podem ser escolhidos e quais as alterações de propriedades destes que podem ser alterados para atender à aplicação do laboratório.

A terceira etapa foi a implementação e construção de elementos de laboratório de controle para apresentar a resposta esperada, fazer um comparativo entre a saída atual simulada e qual a saída inicial e resposta inicial com o sistema sem alterações.

Por fim, foi feita uma bateria de testes de utilização do app ou ambiente virtual, para a verificação de possíveis bugs durante a utilização ou resultados que podiam ser exibidos e que não eram esperados, ou seja, uma última verificação de qualidade do projeto criado antes de poder ser apresentado como trabalho final e de chegar a ser utilizado pelos usuários finais, estudantes e professores de Sistemas de Controle.

4 IMPLEMENTAÇÃO DO PROJETO

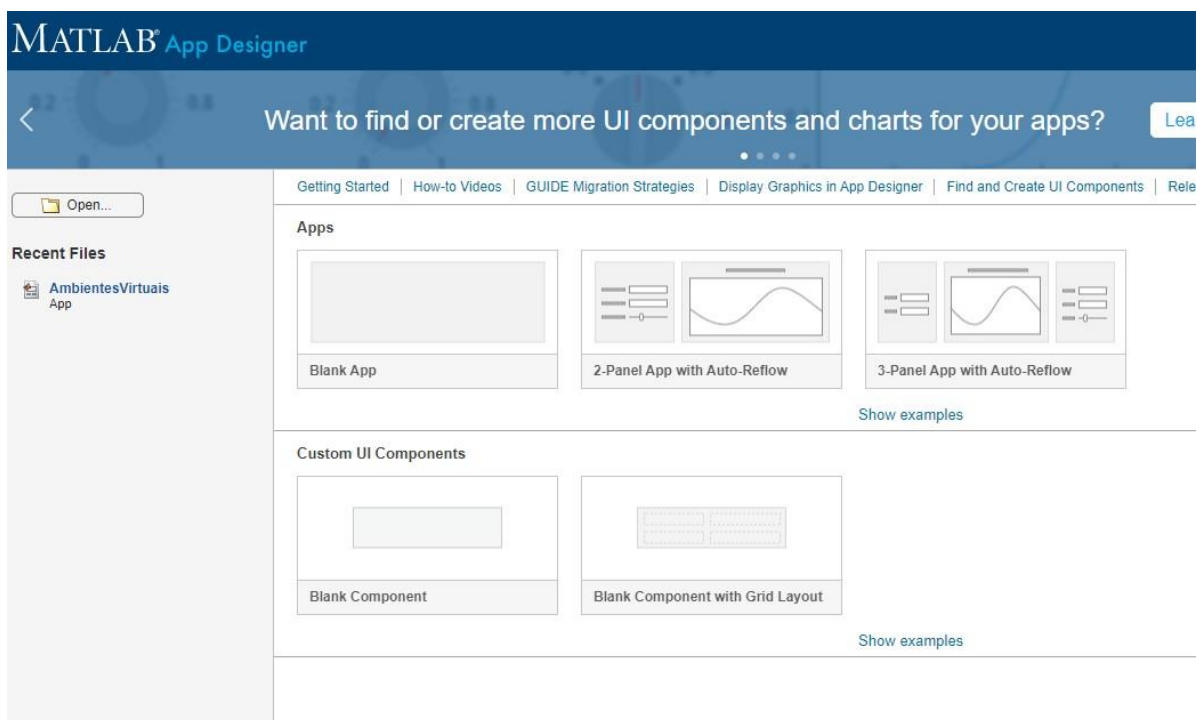
O objetivo desse tópico é aprofundar a descrição de como foi constituído cada etapa do projeto. Iniciando a partir das informações de como a Home Page do App foi construída, passando por toda a construção de cada laboratório e a implementação do app em si, com a interação da Home Page e cada laboratório.

4.1 CONSTRUÇÃO DO APP

O MATLAB possui algumas abas e funcionalidades, o App Designer se encontra na opção “APPS” existente na parte superior da tela principal, e nela a opção “Design App”. Assim pode-se encontrar a principal ferramenta utilizada para o Projeto.

A primeira interface do App Designer pode ser vista abaixo, nela pode-se abrir outros arquivos “mlapp”, assim como escolher qual tipo de app será inicialmente aberto, um “Blank App” ou app em branco para criação, “2 Panel App with Auto-Reflow” ou um app com duas abas, em que os dados de uma refletem na outra que pode possuir um gráfico por exemplo para a saída ou “3 Panel App with Auto-Reflow”, sendo três abas interativas que interagem umas com as outras, assim como uma aba dedicada à customização de componentes de UI (User Interface).

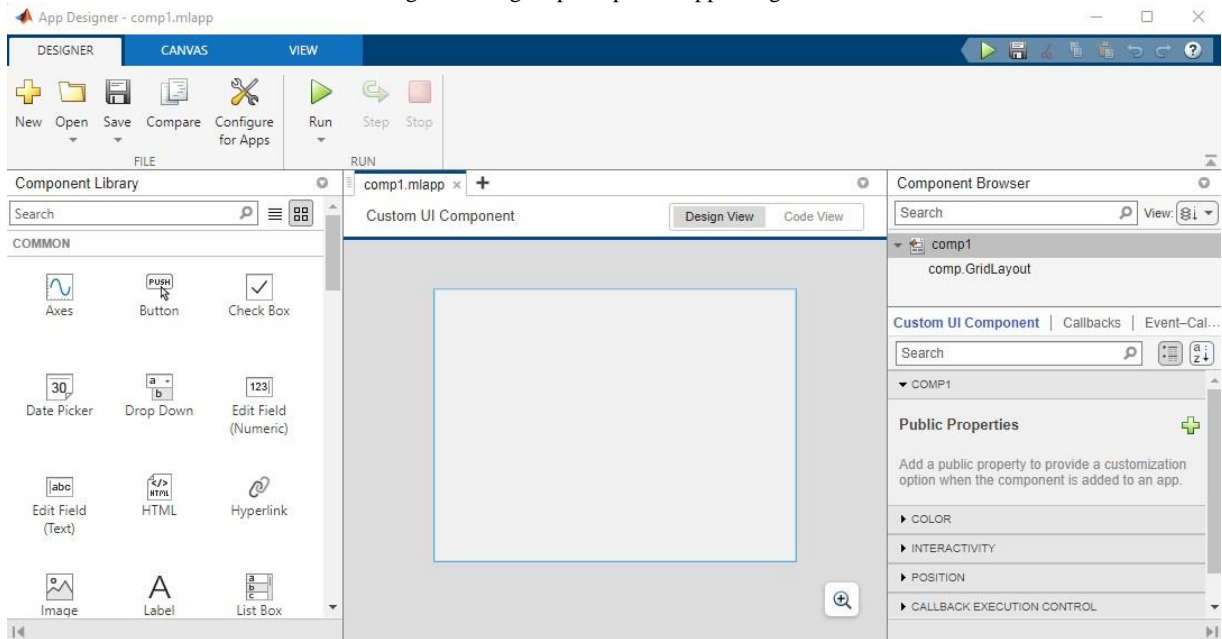
Figura 5 - Interface inicial App Designer.



Fonte: MATLAB.

A figura abaixo retrata a tela principal do App Designer, com as opções de abrir arquivos, salvar e rodar o atual arquivo aberto, assim como as duas abas “Design View” e “Code View” amplamente utilizadas no desenvolvimento do projeto. Pode-se ver as abas com os componentes ou “Component Library”, assim como a aba onde pode-se modificar os objetos “Component Browser”.

Figura 6 - Página principal do App Designer.



Fonte: MATLAB.

Os componentes podem então serem arrastados para a aba de “Designer View” e logo depois suas funções Callbacks podem ser definidas na aba “Code View” bastando selecionar o componente e logo após em “Função Callback” do componente. Os principais componentes utilizados foram então os componentes de imagem e botões, sendo que para os componentes imagens basta selecionar uma imagem armazenada no computador e para os botões basta programar a sua função.

Figura 7 - “Home Page” do app montada inicialmente.



Fonte: Autor.

Ao clicar nos botões da Home Page, os botões são programados para terem um call-back que retorna com o comando “run”, isso inicia outro app com o respectivo nome do botão que é o laboratório em si.

Figura 8 - Código Fonte dos botões tipo “run”.

```

end

% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: Laboratrio4Button
    function Laboratrio4ButtonPushed(app, event)
        run("app1.mlapp");
    end

    % Button pushed function: Laboratrio1Button
    function Laboratrio1ButtonPushed(app, event)

end

end

% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

```

Fonte: Autor

Também foi programado um botão chamado “Ajuda” que ao ser clicado abre um arquivo chamado “Ajuda”, neste arquivo foram colocadas o manual de instruções do aplicativo para o usuário ler e se familiarizar com o aplicativo. Já a programação deste botão e outros botões de interação dentro dos laboratórios utilizaram a função “winopen” que abre qualquer tipo de arquivo do computador e escolhe um leitor específico para cada tipo, se for um arquivo “html”, abrirá um navegador, se for um arquivo pdf, abrirá um leitor pdf por exemplo.

Figura 9 - Código Fonte dos botões tipo “winopen”.

```

    % Button pushed function: AjudaButton
    function AjudaButtonPushed(app, event)
        winopen("Ajuda.pdf");
    end
end

```

Fonte: Autor.

De maneira geral então segue que para botões que abrem arquivos pdf e links são sempre do tipo “winopen”, já botões que abrem imagem de resultados das simulações ou abrem outras abas, como os próprios laboratórios, são os botões do tipo “run”.

4.2 CONSTRUÇÃO DOS LABORATÓRIOS E SEUS ROTEIROS

Nesta seção será apresentada como ficou a atividade proposta em cada um dos cinco roteiros de laboratório escolhidos, seus temas e exatamente o que será lido pelo usuário em cada um dos roteiros de seus respectivos laboratórios, ou seja, os passos a serem tomados pelos estudantes como atividade, e logo após isso, as instruções de como é possível fazer a comparação com o feito manualmente pelo aluno e a automação da atividade laboratorial que é dada pelo aplicativo, a comunicação entre o roteiro proposto e a UI.

4.2.1 Laboratório 1: Função Transferência

O seguinte experimento tem como objetivos simular sistemas de controle e desenvolver as equações diferenciais na forma de função de transferência para obter a resposta no tempo das mesmas para entradas tipo degrau e para entrada tipo rampa.

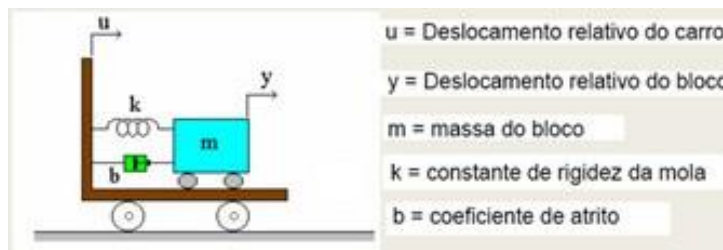
1 - Observando a equação 1 dada e o sistema na figura 1 a seguir, faça o que se pede.

Equação 1:

$$m\ddot{y} + b\dot{y} + ky = u$$

A equação 1 faz referência ao seguinte sistema de controle:

Figura 10 - Sistema de Controle utilizado no relatório.

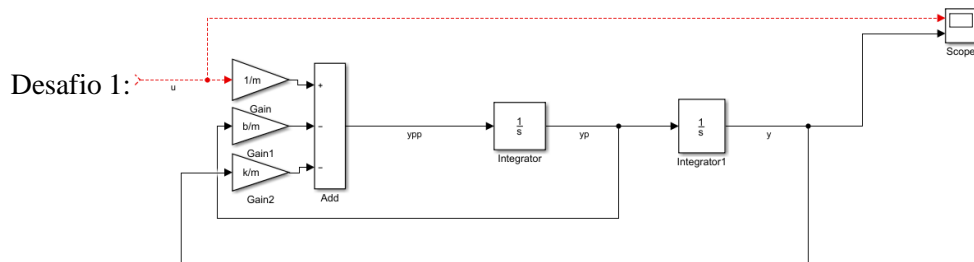


Fonte: K. Ogata

2 - Considere valores para m, k e b aleatórios para estudo, para a resposta do laboratório 1 foram utilizados: $m = 1 \text{ kg}$, $k = 500 \text{ N/m}$, $b = 0.3$.

3 - Monte o seguinte circuito da equação no Simulink para respostas ao degrau e rampa, isso compare com a resposta dada pelo laboratório 1 e tire as suas conclusões.

Figura 11 - Circuito para simulação equação 1.



Fonte: Autor.

Monte o circuito da equação abaixo no Simulink para respostas ao degrau e rampa, após isso compare com o circuito desafio do laboratório, observe a resposta do item 3 e tire as suas conclusões:

$$\ddot{y} + 6\dot{y} + 11y = 6u$$

4.2.2 Laboratório 2: Espaço-Estado

O seguinte experimento tem como objetivos simular sistemas de e desenvolver as equações diferenciais na forma de equações matriciais para espaço estado, obtendo a resposta no tempo das mesmas para entradas tipo degrau e para entrada tipo rampa.

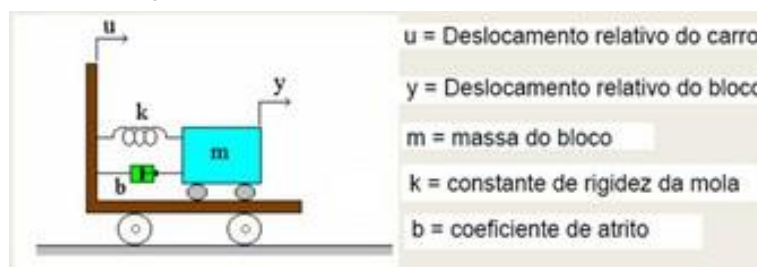
1 – Observando a equação 1 dada e o sistema na figura 1 a seguir, faça o que se pede:

Equação 1:

$$m\ddot{y} + b\dot{y} + ky = u$$

A equação 1 faz referência ao seguinte sistema de controle:

Figura 12 - Sistema de Controle utilizado no relatório.

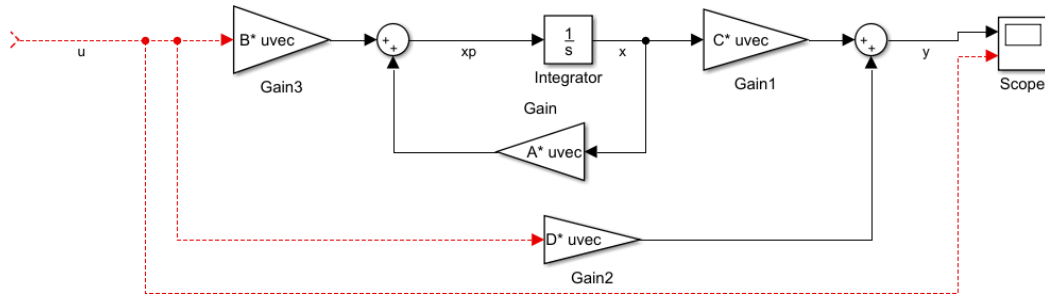


Fonte: K. Ogata

2 – Calcule as matrizes espaço-estado (A, B, C, D) derivadas da conversão da função transferência dada.

3 – Monte o seguinte circuito da equação no Simulink para respostas ao degrau e rampa, após isso compare com a resposta dada pelo laboratório 2 e tire as suas conclusões.

Figura 13 - Circuito para simulação equação 1.



Fonte: Autor.

Desafio 2:

Calcule as matrizes espaço-estado e monte o circuito da equação abaixo no Simulink para respostas ao degrau e rampa, após isso compare com a resposta do item 3 e tire as suas conclusões:

$$\ddot{y} + 6\dot{y} + 11y = 6u$$

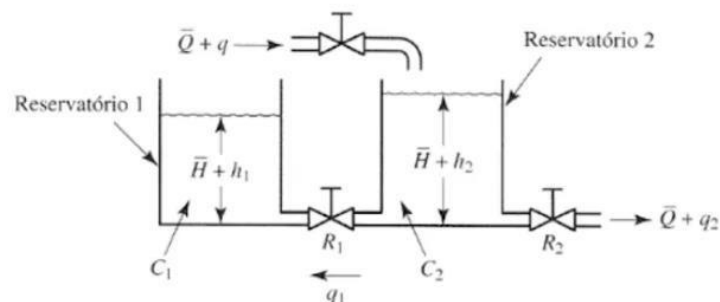
4.2.3 Laboratório 3: Estabilidade

O seguinte experimento tem como objetivos simular sistemas de controle e determinar a função de transferência, a estabilidade do sistema, obtendo a resposta para as malhas abertas e malhas fechadas e os parâmetros da resposta transitória.

1 – A figura 1 demonstra o sistema escolhido para se fazer as análises em relação a estabilidade e a resposta transitória.

Figura 14 - Sistema de nível de líquido.

$$R_1 C_1 R_2 C_2 \frac{d^2 h_1}{dt^2} + (R_1 C_1 + R_2 C_2 + R_2 C_1) \frac{dh_1}{dt} + h_1 = R_2 q$$



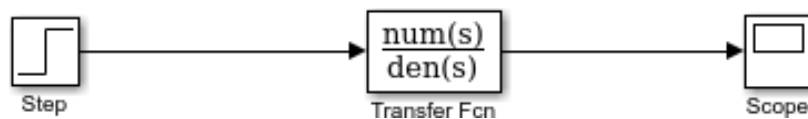
Fonte: Autor.

- 2 – Encontre a equação da função transferência do sistema dado para malha aberta.
- 3 – Monte a tabela de Rote.
- 4 – Determine se o sistema é estável, se sim, para qual condição.
- 5 – Encontre a equação da função transferência do sistema dado para malha fechada com realimentação unitária.
- 6 – Determine se o sistema é estável, se sim, para qual condição.
- 7 – Encontre a equação da função transferência do sistema dado para malha fechada com uma função de transferência de primeira ordem $H(s)$:

$$H(s) = \frac{1}{s}$$

- 8 – Determine se o sistema é estável, se sim, para qual condição.
- 9 – Considere $W_n^2 = 4$, sendo assim os valores de R (resistência hidráulica) e C (capacidade) são aleatórios. Encontre a equação de transferência de 2º grau e determine que tipo de resposta o sistema terá a uma entrada degrau.
- 10 – Encontre os parâmetros da resposta transitória (t_r , t_p , t_s , M_p).
- 11 - Monte o seguinte circuito equação no Simulink com a equação transferência de 2º grau encontrada para resposta ao degrau e após isso compare com a resposta dada pelo laboratório 3, observe os parâmetros na saída e tire as suas conclusões.

Figura 15 - Circuito da questão com entrada degrau.



Fonte: Autor.

Desafio 3:

Atribua o valor de $W_n^2 = 16$ e repita todos os passos para a nova função transferência encontrada.

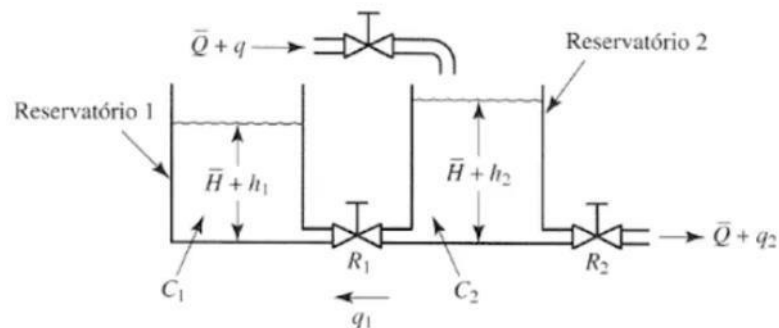
4.2.4 Laboratório 4: LGR

O seguinte experimento tem como objetivos simular sistemas de controle e determinar graficamente o lugar geométrico das raízes para funções de transferência em as malhas abertas e malhas fechadas, assim como determinar o comportamento desses sistemas.

1 – A figura 1 demonstra o sistema escolhido para se fazer as análises em relação a estabilidade e a resposta transitória.

Figura 16 - Sistema de nível de líquido questão 1.

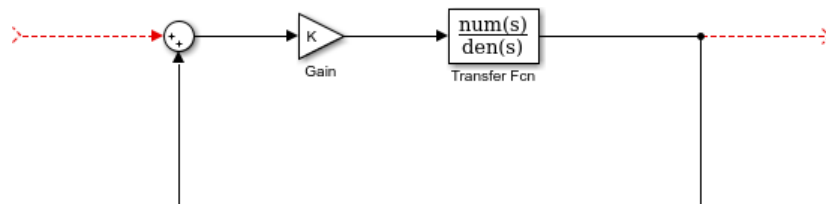
$$R_1 C_1 R_2 C_2 \frac{d^2 h_1}{dt^2} + (R_1 C_1 + R_2 C_2 + R_2 C_1) \frac{dh_1}{dt} + h_1 = R_2 q$$



Fonte: Autor.

- 1 – Encontre a equação da função transferência do sistema dado para malha aberta.
- 2 – Encontre a equação da função transferência do sistema dado para malha fechada com realimentação unitária.
- 3 - Encontre a equação da função transferência do sistema dado:

Figura 17 - Circuito 1 questão 3.



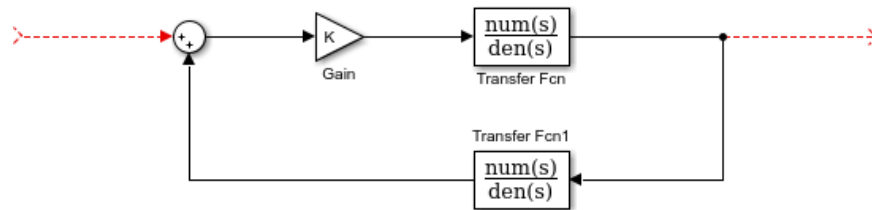
Fonte: Autor.

- 4 – Encontre a equação da função transferência do sistema dado para malha fechada com uma função de transferência de primeira ordem $H(s)$:

$$H(s) = \frac{1}{s}$$

5 – Encontre a equação da função transferência do sistema dado (Fcn1 = H(s)):

Figura 18 - Circuito 2 questão 5.



Fonte: Autor.

6 – Considere valores aleatórios de $R_1 = R_2 = C_1 = C_2 = 1$, para que assim seja possível encontrar o lugar geométrico das raízes para os sistemas.

7.1 – Para o sistema do item 3, encontre as raízes da equação para $K = 0$, $K = 1$, $K = 5/4$ (Exemplo de valores de K indicados para chegar na resposta dada pelo laboratório).

7.2 – Determine se o sistema é estável, se sim, para qual condição.

7.3 – Simule a resposta no MATLAB utilizando a rltool, para $K = 1$.

7.4 – Descreva qual o tipo é o tipo de sistema, compare com a resposta dada pelo laboratório 4 e tire as suas conclusões.

8.1 – Para o sistema do item 5, encontre as raízes da equação para $K = 0$, $K = 1$ (Exemplo de valores de K indicados para chegar na resposta dada pelo laboratório).

8.2 – Determine se o sistema é estável, se sim, para qual condição.

8.3 – Simule a resposta no MATLAB utilizando a rltool, para $K = 1$.

8.4 – Descreva qual o tipo é o tipo de sistema, compare com a resposta dada pelo laboratório 4 e tire as suas conclusões.

Desafio 4:

Considere desta vez a função transferência abaixo como a equação principal para o item 3 e repita os itens 3, 4, 5, 7.1 - 7.3 e 8.1 - 8.3. Após isso compare todos os resultados e tire suas conclusões.

$$\frac{4}{s^2 + 2s + 4}$$

4.2.5 Laboratório 5: PID

O seguinte experimento tem como objetivos simular sistemas de controle e determinar graficamente as características e condições de operação dos controladores PID nos mais diversos tipos de sistemas de controle.

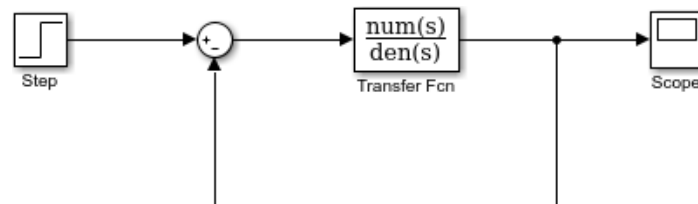
1 – Observando a equação de transferência de primeiro grau dada faça o que se pede.

Equação:

$$\frac{1}{s + 1}$$

2 – Monte o seguinte circuito da equação no Simulink para resposta ao degrau e observe a saída da simulação.

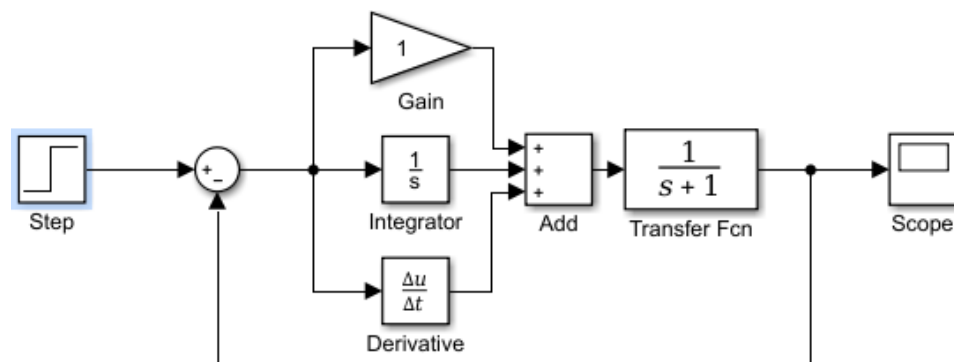
Figura 19 - Circuito 1 questão 2.



Fonte: Autor.

3 - Monte o seguinte circuito da equação com um controlador PID básico (considerando $K_p = K_d = K_i = 1$) no Simulink para resposta ao degrau e observe a saída da simulação.

Figura 20 - Circuito 2 PID.



Fonte: Autor.

4 – Encontre a equação do controlador PID.

5 – Compare as duas saídas, com a resposta dada pelo laboratório 5 e tire as suas conclusões.

6 – Modifique os valores de K_p , K_d e K_i , observe as saídas e tire suas conclusões.

Desafio 5:

Do sistema turbina-gerador, sistema Desafio do laboratório 5, encontre a equação resultante dos três blocos, aplique um controlador PID e compare a saída com e sem o controlador PID aplicado.

4.3 VERSÃO FINAL DO APP E INSTRUÇÕES

Cada parte do app montado, todas as instruções e funções do aplicativo encontram-se abaixo, e todas elas estão também no arquivo ajuda que é aberto quando se clica no botão “Ajuda” em qualquer um dos laboratórios ou “Home Page”.


1 – Para o aplicativo “Ambientes Virtuais” funcionar corretamente é preciso ter instalado na máquina o programa “MATLAB”.

2 – Para seguir alguns roteiros de laboratórios será necessária a utilização do “SIMULINK” ou “RLTOOL” do “MATLAB”.

3 – Para interagir no “Ambientes Virtuais” é preciso clicar nos botões indicados.

4 – Alguns botões abrem outras abas e outros botões abrem documentos.

5 – Para conseguir abrir documentos é preciso ter na máquina instalado um leitor PDF.

6 – Para fechar as abas basta fechar no botão: 

7 – Para maximizar as abas basta clicar no botão: 

8 – Para minimizar as abas basta clicar no botão: 

4.3.1 Ambientes Virtuais “Home Page”

Figura 21 - Tela Inicial.



Fonte: Autor.

Instruções:

- 1 – Clique no botão “Laboratório 1” para abrir o laboratório 1.
- 2 – Clique no botão “Laboratório 2” para abrir o laboratório 2.
- 3 – Clique no botão “Laboratório 3” para abrir o laboratório 3.
- 4 – Clique no botão “Laboratório 4” para abrir o laboratório 4.
- 5 – Clique no botão “Laboratório 5” para abrir o laboratório 5.
- 6 – Clique no botão “Ajuda” para abrir o documento “Ajuda”.

4.3.2 Ambientes Virtuais “Laboratório 1: Função Transferência”

Figura 22 - Laboratório 1: Função Transferência.

Roteiro do Laboratório 1

O seguinte experimento tem como objetivos simular sistemas de controle e desenvolver as equações diferenciais na forma de função de transferência para obter a resposta no tempo das mesmas para entradas tipo degrau e para entrada tipo rampa.

A equação faz referência ao seguinte sistema de controle:

$$m\ddot{y} + b\dot{y} + ky = u$$

u = Deslocamento relativo do carro
y = Deslocamento relativo do bloco
m = massa do bloco
k = constante de rigidez da mola
b = coeficiente de atrito

3 **Degrau**

4 **Rampa**

5 **Aulas**

6 **Cálculos**

Equação e Sistema para o Desafio 1:

$$\ddot{y} + 6\dot{y} + 11y = 6u$$

Fonte: Autor.

Instruções:

- 1 – Clique no botão “Roteiro do laboratório” para abrir o documento do Roteiro do Laboratório.
- 2 – Clique no botão “Ajuda” para abrir o documento “Ajuda”.
- 3 – Clique no botão “Degrau” para a resposta do sistema ao degrau.
- 4 – Clique no botão “Rampa” para a resposta do sistema à rampa.
- 5 – Clique no botão “Aulas” para abrir um link para um drive com aulas da matéria “Sistemas de Controle I” ministradas pelo professor *Dr. Daniel Guzmán Del Río* para auxílio nas práticas laboratoriais.
- 6 – Clique no botão “Cálculos” para abrir o documento com os cálculos e respostas.

4.3.3 Ambientes Virtuais “Laboratório 2: Espaço-Estado”

Figura 23 - Laboratório 2: Espaço-Estado.

Laboratório 2: Espaço-Estado

Roteiro do Laboratório 1

O seguinte experimento tem como objetivos simular sistemas de controle e desenvolver as equações diferenciais na forma de equações matriciais para espaço estado, obtendo a resposta no tempo das mesmas para entradas tipo degrau e para entrada tipo rampa.

A equação faz referência ao seguinte sistema de controle:

$$m\ddot{y} + b\dot{y} + ky = u$$

u = Deslocamento relativo do carro
y = Deslocamento relativo do bloco
m = massa do bloco
k = constante de rigidez da mola
b = coeficiente de atrito

3 **Degrau**

4 **Rampa**

Aulas 5

Cálculos 6

Equação para o Desafio 2:

$$y''' + 6\ddot{y} + 11\dot{y} + 6y = 6u$$

$$y''' = 6u - 6\ddot{y} - 11\dot{y} - 6y$$

State-space equations shown in the block diagram:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Fonte: Autor.

Instruções:

- 1 – Clique no botão “Roteiro do laboratório” para abrir o documento do Roteiro do Laboratório.
- 2 – Clique no botão “Ajuda” para abrir o documento “Ajuda”.
- 3 – Clique no botão “Degrau” para a resposta do sistema ao degrau.
- 4 – Clique no botão “Rampa” para a resposta do sistema à rampa.
- 5 – Clique no botão “Aulas” para abrir um link para um drive com aulas da matéria “Sistemas de Controle I” ministradas pelo professor *Dr. Daniel Guzmán Del Río* para auxílio nas práticas laboratoriais.
- 6 – Clique no botão “Cálculos” para abrir o documento com os cálculos e respostas.

4.3.4 Ambientes Virtuais “Laboratório 3: Estabilidade”

Figura 24 - Laboratório 3: Estabilidade.

Roteiro do Laboratório 1

O seguinte experimento tem como objetivos simular sistemas de controle e determinar a função transferência, a estabilidade do sistema, obtendo a resposta para as malhas abertas e os parâmetros da resposta transitória.

A figura abaixo demonstra o sistema escolhido para se fazer as análises em relação a estabilidade e a resposta transitória:

$$R_1 C_1 R_2 C_2 \frac{d^2 h_1}{dt^2} + (R_1 C_1 + R_2 C_2 + R_2 C_1) \frac{dh_1}{dt} + h_1 = R_2 q$$

Reservatório 1, Reservatório 2, $\bar{Q} + q$, $\bar{H} + h_1$, $\bar{H} + h_2$, C_1 , R_1 , C_2 , R_2 , q_1 , $\bar{Q} + q_2$

Circuito para a simulação:

Parâmetros da resposta transitória:

Degrau 2 num(s) den(s) Transfer Fcn Scope

Aulas 3

Cálculos 4

5 6 7 8 9

tr 7

tp 8

ts

Mp

Ajuda

Fonte: Autor.

Instruções:

- 1 – Clique no botão “Roteiro do laboratório” para abrir o documento do Roteiro do Laboratório.
- 2 – Clique no botão “Degrau” para a resposta do sistema ao degrau.
- 3 – Clique no botão “Aulas” para abrir um link para um drive com aulas da matéria “Sistemas de Controle I” ministradas pelo professor *Dr. Daniel Guzmán Del Río* para auxílio nas práticas laboratoriais.
- 4 – Clique no botão “Cálculos” para abrir o documento com os cálculos e respostas.
- 5 – Clique no botão “ts” para a resposta do parâmetro transitório “ts”.
- 6 – Clique no botão “Mp” para a resposta do parâmetro transitório “Mp”.
- 7 – Clique no botão “tr” para a resposta do parâmetro transitório “tr”.
- 8 – Clique no botão “tp” para a resposta do parâmetro transitório “tp”.
- 9 – Clique no botão “Ajuda” para abrir o documento “Ajuda”.

4.3.5 Ambientes Virtuais “Laboratório 4: LGR”

Figura 25 - Laboratório 4: LGR.

Laboratório 4: LGR

Roteiro do Laboratório 1

O seguinte experimento tem como objetivos simular sistemas de controle e determinar graficamente o lugar geométrico das raízes para funções de transferência em as malhas abertas e malhas fechadas, assim como determinar o comportamento desses sistemas.

A figura abaixo demonstra o sistema escolhido para se fazer as análises em relação a estabilidade e a resposta transitória:

$$R_1 C_1 R_2 C_2 \frac{d^2 h_1}{dt^2} + (R_1 C_1 + R_2 C_2 + R_2 C_1) \frac{dh_1}{dt} + h_1 = R_2 q$$

Reservatório 1
Reservatório 2

2 **Degrau1**

3 **Degrau2**

4 **Aulas**

5 **Cálculos**

6 **Ajuda**

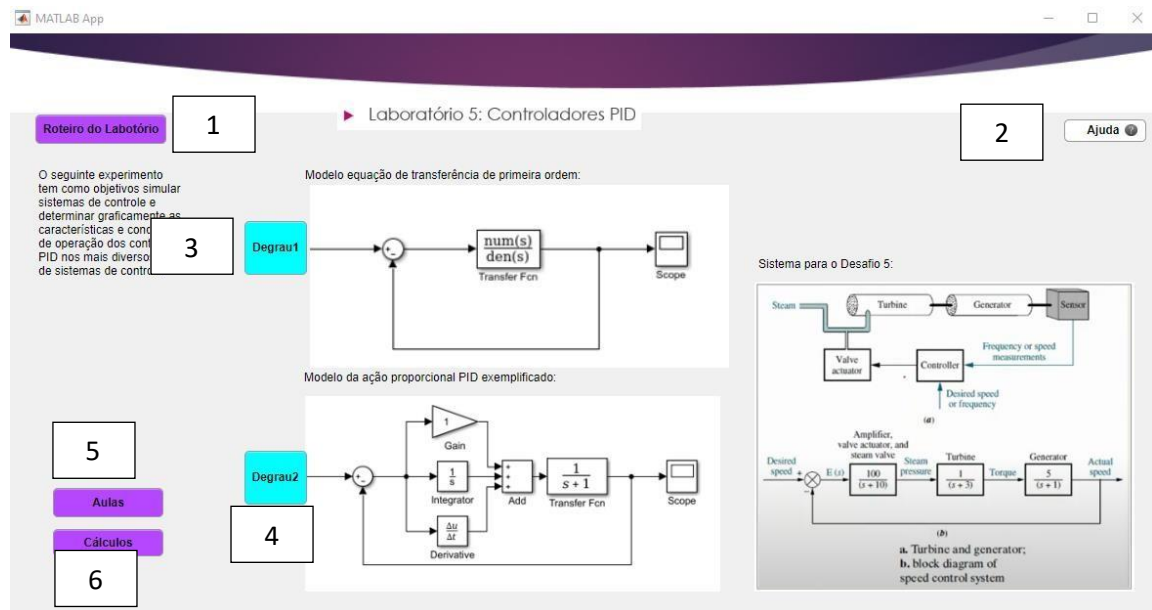
Fonte: Autor.

Instruções:

- 1 – Clique no botão “Roteiro do laboratório” para abrir o documento do Roteiro do Laboratório.
- 2 – Clique no botão “Degrau1” para a resposta do primeiro sistema ao degrau.
- 3 – Clique no botão “Degrau2” para a resposta do segundo sistema ao degrau.
- 4 – Clique no botão “Aulas” para abrir um link para um drive com aulas da matéria “Sistemas de Controle I” ministradas pelo professor *Dr. Daniel Guzmán Del Río* para auxílio nas práticas laboratoriais.
- 5 – Clique no botão “Cálculos” para abrir o documento com os cálculos e respostas.
- 6 – Clique no botão “Ajuda” para abrir o documento “Ajuda”.

4.3.6 Ambientes Virtuais “Laboratório 5: Controladores PID”

Figura 26 - Laboratório 5: Controladores PID.



Fonte: Autor.

Instruções:

- 1 – Clique no botão “Roteiro do laboratório” para abrir o documento do Roteiro do Laboratório.
- 2 – Clique no botão “Ajuda” para abrir o documento “Ajuda”.
- 3 – Clique no botão “Degrau1” para a resposta do primeiro sistema ao degrau.
- 4 – Clique no botão “Degrau2” para a resposta do segundo sistema ao degrau.
- 5 – Clique no botão “Aulas” para abrir um link para um drive com aulas da matéria “Sistemas de Controle I” ministradas pelo professor *Dr. Daniel Guzmán Del Río* para auxílio nas práticas laboratoriais.
- 6 – Clique no botão “Cálculos” para abrir o documento com os cálculos e respostas.

4.4 FINALIZAÇÃO E COMPACTAÇÃO PARA DISTRIBUIÇÃO DO APP

Para a compactação do app final todos os arquivos, desde arquivos “mlapps” gerados, até figuras, arquivos “html” e arquivos pdf utilizados, foram todos colocados em uma única pasta com o nome do aplicativo “Ambientes Virtuais”, para serem compactados pelo próprio MATLAB com a função que o App Designer provê para isso, sendo assim, esta é a pasta a ser instalada na máquina do usuário. A forma final do aplicativo é um “Setup” básico como qualquer outro programa, que será o instalador do aplicativo.

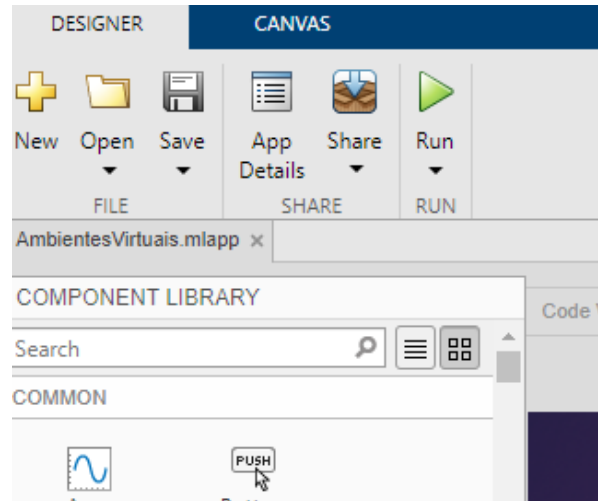
Figura 27 - Pasta do aplicativo.

Nome	Data de modificação	Tipo	Tamanho
Ajuda	20/03/2023 23:36	Documento do A...	732 KB
AmbientesVirtuais	20/03/2023 23:06	MATLAB App	224 KB
AV1	08/02/2023 17:08	Arquivo PNG	53 KB
AV2	15/03/2023 22:51	Arquivo PNG	155 KB
AV3	15/03/2023 22:58	Arquivo PNG	2 KB
AV4	27/02/2023 22:49	Arquivo PNG	25 KB
AV5	15/03/2023 23:14	Arquivo JPG	105 KB
Lab1	20/03/2023 23:03	MATLAB App	96 KB
Lab1aa	12/02/2023 23:40	Arquivo PNG	29 KB
Lab1b	12/02/2023 22:05	Arquivo PNG	37 KB
Lab1c	12/02/2023 22:06	Arquivo PNG	3 KB

Fonte: Autor.

Na opção “Share” é possível selecionar “Standalone Desktop App” que gerará o pacote de instalação para outros usuários de MATLAB.

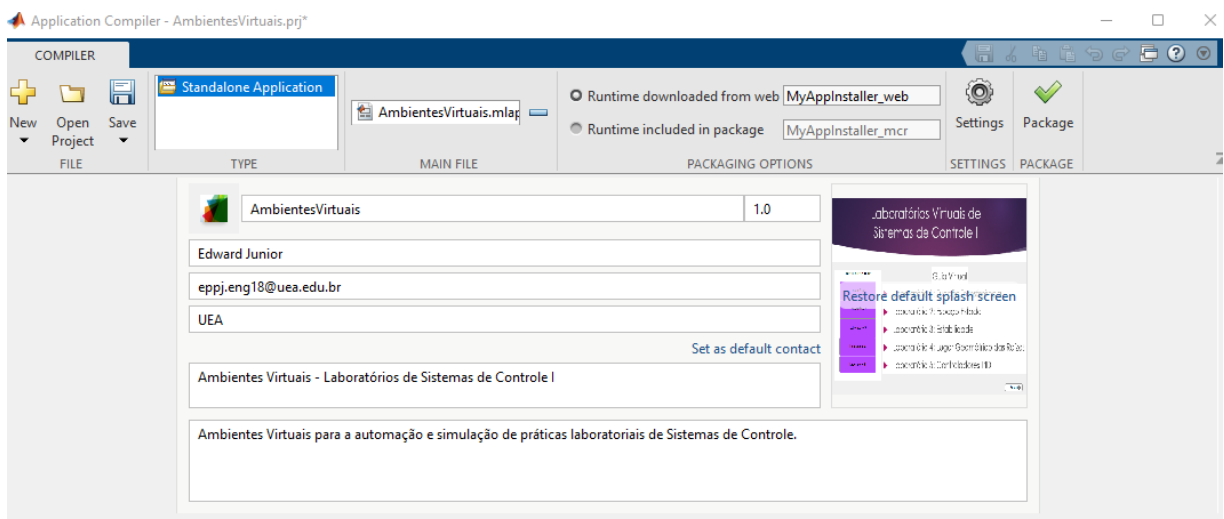
Figura 28 - Opção “Share”.



Fonte: Autor.

A aba “Package App” é então aberta, e nela pode-se inserir todas as informações necessárias para o app, desde o nome do autor até quais arquivos devem ser compactados junto com o app principal.

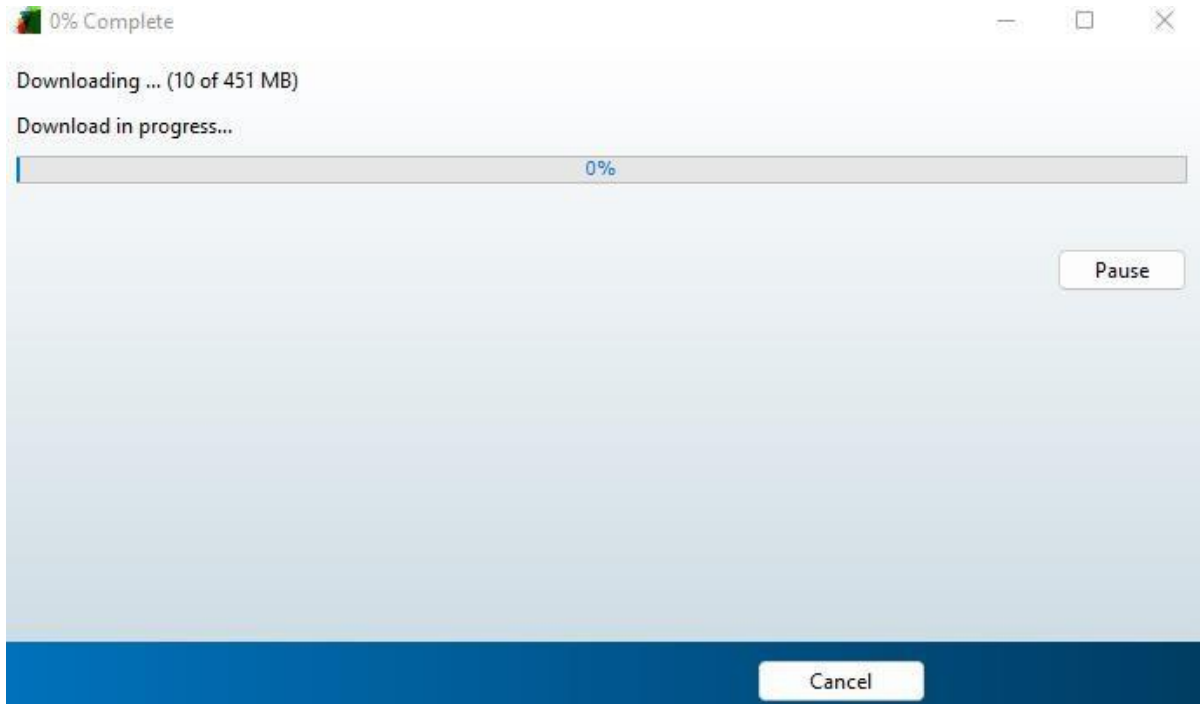
Figura 29 - Compilador de Aplicativo.



Fonte: Autor.

O arquivo gera um instalador e a saída para instalar na pasta raiz do computador do usuário fica deste modo:

Figura 30 – Instalação do aplicativo.



Fonte: Autor.

5 CONCLUSÃO

Pode-se observar que é sim possível a realização e elaboração de ambientes virtuais utilizando o MATLAB, assim como entregar uma possível solução para a ausência ou dificuldades na execução de práticas laboratoriais reais, através do projeto foi possível também a criação de uma ferramenta de aprendizagem, simulação e automação, adequada para a aquisição das habilidades e experiências requeridas no curso de Engenharia, isso relacionado à sistemas de controle.

Foi também o intuito deste trabalho lutar contra o problema que afeta as principais universidades e escolas públicas, a falta de recursos econômicos, instrumentação real para a realização das práticas laboratoriais de forma física, assim como o alto custo de equipamentos e componentes, sendo possível alcançar o objetivo inicial, já que o aplicativo é funcional, nele se pode acompanhar roteiros de ensino utilizados em aulas reais de laboratórios de Sistemas de Controle, a nível graduação de engenharia, e também acompanhar a saída real de uma automação de prática laboratorial que é gerada como resposta em cada um dos laboratórios propostos, e comparar a solução própria com a solução correta que também é provida pelo aplicativo em forma de cálculos e observações. Além disso, é visto que este aplicativo pode quebrar as barreiras que o ensino à distância cria em uma elaboração de uma prática laboratorial.

Através do projeto concluído, pode-se enfim analisar as formas de desenvolver ambientes e laboratórios virtuais utilizando MATLAB, pois, assim como os cinco laboratórios que foram desenhados facilmente utilizando a ferramenta App Designer, e neste caso são ambientes para estudo e práticas da matéria Sistemas de Controle I, podem também serem construídos outros ambientes para outras matérias e práticas, a fim de alcançar a maior abrangência possível na grade de Engenharia Elétrica e Eletrônica.

O projeto não é contido em apresentar de maneira direta uma solução ao problema proposto, mas também é uma fonte para possíveis implementações futuras, já que este é um aplicativo, um aplicativo pode ter infinitas versões, então esta pode ser considerada uma primeira versão finalizada, que resolve a incógnita a qual ele foi confrontado, mas assim como ele resolve isto, ele pode também através de atualizações resolver outros problemas e ser bem mais desenvolvidos em futuros projetos, além deste Trabalho de Conclusão de Curso.

Este trabalho é então, um passo para a evolução das ferramentas para a integração da sociedade e a tecnologia na vida moderna, integrando o ensino a curta ou à longa distância, facilitando a comunicação direta entre alunos que almejam se aprofundar nos assuntos de Sistemas de Controle e suas práticas, assim como os professores que atuam na área.

6 REFERÊNCIAS BIBLIOGRÁFICAS

OGATA, K. - Engenharia de Controle Moderno, Prentice-Hall, 5ª. ed., 2010.

FRIEDLAND, B.-Control System Design: An Introduction to State Space Methods, McGraw-Hill, 1987.

FRANKLIN, G.; Powell, J.D.; Naeini, A.E. – Feedback Control of Dynamic Systems, 3rd Edition, Addison-Wesley, 1994.

DAZZO J.J. & HOUPIS, C. H. Análise de Projeto de Sistemas de Controle Lineares, 2ª Edição, Editora Guanabara, 1984.

NISE, N. S. Engenharia de Sistemas de Controle, 3ª Edição, LTC, 2002.

DORF, R. C & BISHOP, R. H. Sistemas de Controle Moderno, Addison Wesley Longman, 13ª Edição, 2018.

KUO, B.C. – Automatic Control Systems, 7th Edition, Prentice Hall, 1995.

BOLTON, W. Control Engineering , Longman Scientific & Technical ,1992, UK.

KWAKERNAAK, H. & SIVAN, R. Linear optimal control systems, Wiley Interscience, N. York, 1972.

MATHWORKS. creating_guis: *MATLAB® App Building*” da Mathworks. Disponível em: <https://www.mathworks.com/help/pdf_doc/matlab/creating_guis.pdf>. Acesso em 27/03/2023.

7 FONTES CONSULTADAS

MOZILLA. DEVELOPER, Callback function. Disponível em: https://developer.mozilla.org/pt-BR/docs/Glossary/Callback_function. Acesso em 28/08/2022.

BLENDER. Parent Objects and properties. Disponível em: https://docs.blender.org/manual/pt/2.80/scene_layout/object/properties/relations/parents.html. Acesso em 02/09/2022.

ACERVOLIMA Matlab functions. Disponível em: <[https://acervolima.com/matplotlib-pyplot-gcf-em-python/#:~:text=gcf\(\)%20%C3%A9%20usado%20principalmente,ajuda%20da%20figure\(\)%20fun%C3%A7%C3%A3o](https://acervolima.com/matplotlib-pyplot-gcf-em-python/#:~:text=gcf()%20%C3%A9%20usado%20principalmente,ajuda%20da%20figure()%20fun%C3%A7%C3%A3o)>. Acesso em 13/09/2022.

MATHWORKS. Preventing access to figure and axes. Disponível em: <https://www.mathworks.com/help/matlab/creating_plots/preventing-access-to-figures-and-axes.html> Acesso em 28/03/2023.

Aguirre, L.A Aguirre. Disponível em: <[youtube.com/Luis_Antonio_Aguirre](https://www.youtube.com/Luis_Antonio_Aguirre)>. Acesso em 27/03/2023.

PID_THEORY. Disponível em: pcbheaven.com/wikipages/PID_Theory. Acesso em 27/03/2023.