

## Laboratório virtual de Controle de Sistemas utilizando VPython: Uma análise da efetividade no ensino de Engenharia e perspectivas para o futuro

### INSTRUÇÕES PARA A PREPARAÇÃO E SUBMISSÃO DE TRABALHOS À COMISSÃO CIENTÍFICA DO 51º CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA E VI SIMPÓSIO INTERNACIONAL DE EDUCAÇÃO EM ENGENHARIA – COBENGE 2023

- Observe as instruções e formate seu artigo de acordo com este padrão (*template*). Recomenda-se, para isso, o uso dos estilos de formatação definidos neste documento: **basta copiar e colar o seu texto diretamente em uma cópia deste template** e de acordo com as seções. O arquivo não deve ultrapassar 12 (doze) páginas e deve ser submetido em formato PDF. Lembre-se de que o uso adequado da linguagem é indispensável para a boa avaliação. O trabalho deve atender à temática da Educação em Engenharia.
- **Não inserir informações dos autores e afiliações, resumo e palavras-chave no corpo do trabalho.** Essas informações devem ser inseridas somente no formulário *online* de submissão.
- Não será permitida a inclusão ou exclusão de autores após o término do prazo de submissão.
- **Trabalhos fora dos padrões de formatação estabelecidos serão rejeitados.**

(RETIRAR O TEXTO ACIMA PARA A SUBMISSÃO DO SEU TRABALHO)



## ESTRUTURA ATUAL

—  
**INTRODUÇÃO (OK - PRECISA DE REVISÃO)**

**METODOLOGIA**

**MÓDULO 1**

**Modelagem dos sistemas físicos**

**Controle**

**Simulação numérica**

**MÓDULO 2**

**Modelagem dos sistemas físicos**

**Controle**

**Simulação numérica**

**CONCEITOS UTILIZADOS EM AMBOS OS MÓDULOS**

**Implementação computacional (Python, bibliotecas utilizadas).**

**Animação (VPython, classes criadas, descrição do código)**

**Proposta pedagógica**

**RESULTADOS**

**MÓDULO 1**

**MÓDULO 2**

**CONCLUSÃO**

**AGRADECIMENTOS**

**REFERÊNCIAS**

—  
**RESUMO**  
—

## 1. INTRODUÇÃO

O ensino de disciplinas teóricas nos cursos de engenharia é muitas vezes visto com maus olhos por grande parte dos acadêmicos, pois, há a dificuldade de relacionar o conteúdo com o seu comportamento físico. A introdução de tecnologias voltadas para a apresentação e o desenvolvimento de protótipos miniaturizados, acabou desmistificando esse preconceito e despertando o interesse dos discentes, estimulando assim a assimilação do conteúdo visto em sala de aula. No entanto, esse processo ainda é lento, devido a uma série de fatores, dentre eles: a falta de estrutura, de atualização das metodologias de ensino e de apoio socioeconômico por parte das instituições. Visando acelerar o processo, uma das formas que tem se tornado bastante atraente nos últimos anos, foi a inserção das linguagens de programação nos conteúdos ministrados para esses alunos, dado a simplificação dessas ferramentas, como é o caso do Python, por exemplo.

Este trabalho visa apresentar uma nova maneira de transmitir o conteúdo em sala de aula ou fora dela. Para atingir este objetivo, serão utilizados os conceitos de modelagem de sistemas físicos através de ferramentas computacionais para criar um conjunto de simulações e animações, que além de propiciar um apelo visual diferenciado ao usuário, também irá fornecer a possibilidade de interação e controle desses sistemas.

Em um primeiro momento, foram construídas duas animações voltadas para as disciplinas de controle do curso de engenharia elétrica do campus da UFPA de Tucuruí no Pará. Uma delas é a simulação de um Aero pêndulo, que foi modelado com o uso das bibliotecas *Numpy* e *Control* da linguagem Python. Para a animação foi usada a biblioteca *Vpython*, uma poderosa ferramenta quando se trata de simulações e animações no ambiente computacional. A mesma linha de raciocínio foi utilizada para a outra animação, o Levitador Magnético.

A distribuição desses sistemas será feita através da plataforma *Jupyter Notebook*, pois, além de uma interface simples e didática é possível combinar códigos e textos em só um lugar. A distribuição desses sistemas será realizada por meio da plataforma Jupyter Notebook, que oferece uma interface intuitiva e didática, permitindo a combinação de códigos e textos em um único lugar. Com essa abordagem, os usuários terão acesso a uma ferramenta interativa e integrada, que facilita o aprendizado dos conceitos fundamentais de controle dos sistemas MAGLEV e Aero Pêndulo, e dos demais que serão adicionados ao laboratório virtual.

## 2. METODOLOGIA

As simulações desenvolvidas para a aplicação como ferramentas de ensino de temas pertinentes à disciplina de controle na engenharia, foram baseadas em sistemas físicos reais elaborados em outros trabalhos. Dito isto, será apresentado sucintamente os conceitos e técnicas empregados em cada módulo (simulação), a abordagem computacional utilizada e as formas de inserir essas ferramentas no âmbito educacional.

### 2.1 Módulo 1 - MAGLEV (Magnetic Levitator)

O sistema MAGLEV (*Magnetic Levitator*) é uma tecnologia que permite a suspensão de um corpo ferromagnético no ar através da utilização de uma força magnética gerada por um eletroímã. Essa força, contrária ao peso do objeto, estabelece uma relação de equilíbrio, conforme explica Milhomem (2010). Com essa técnica, é possível manter objetos suspensos sem a necessidade de apoios físicos, o que traz diversas aplicações práticas em áreas como transporte eletromagnético e levitação magnética.

#### **Modelagem do sistema físico**

Devido à natureza não linear da dinâmica do modelo em estudo, a resolução analítica deste problema se torna complexa com as técnicas ensinadas na graduação. Para lidar com sistemas não lineares como este, é necessário utilizar técnicas mais abrangentes, como as equações de estado. Segundo Nise (2012), as equações de estado fornecem uma representação matemática completa do sistema, permitindo sua análise e simulação de forma mais eficaz. Neste trabalho, as equações de estado serão utilizadas para modelar e analisar a dinâmica do sistema, permitindo avaliar seu comportamento e desempenho em diferentes condições operacionais.

As equações de estado utilizadas neste trabalho foram retiradas do estudo de Costa, Silva e Teixeira (2012), que propuseram um modelo analítico baseado no conceito de circuitos magnéticos para a simulação de sistemas de levitação magnética. Além disso, os parâmetros de uma bancada MAGLEV real, que será utilizada como base para a simulação, foram obtidos a partir deste mesmo estudo. As equações de estado estão expressas na Equação (1) e Equação (2).

$$\dot{x}_1 = x_2 \quad (1)$$

$$\dot{x}_2 = g - \frac{kl^2}{m(x_1 - \mu)^2} \quad (2)$$

A técnica de equações de estado é essencial para tratar sistemas não lineares como este, pois permite a modelagem e análise do sistema de forma mais geral e precisa.

Por meio da simulação com os dados da bancada MAGLEV real, é possível validar as equações de estado e verificar a eficácia do modelo proposto.

Para melhorar a análise do modelo descrito pelas Equações (1) e (2), foi aplicada a técnica de linearização, que consiste em aproximar o modelo não linear por um modelo linear em torno de um ponto de equilíbrio. Segundo Nise (2012), essa técnica permite analisar a dinâmica do sistema com maior facilidade e precisão. Para definir o ponto de equilíbrio, foram realizados ensaios experimentais, baseados na distância entre o eletroímã e o corpo flutuante, seguindo os procedimentos descritos no estudo de Costa, Silva e Teixeira (2012). Estes ensaios permitiram determinar as condições em que o sistema opera de forma estável, fornecendo uma referência para a linearização e análise da dinâmica em torno desse ponto de equilíbrio.

A partir desse ponto de equilíbrio, as equações de estado foram linearizadas usando a expansão em séries de Taylor. As equações de estado linearizadas, descritas na Equação (3) e Equação (4), consideram como ponto de equilíbrio os valores de posição e corrente do sistema ( $x_0, I_0$ ). Com o objetivo de obter uma análise mais precisa do comportamento do sistema em torno do ponto de equilíbrio, foram utilizadas equações de estado simplificadas, as quais foram propostas por Milhomem (2010) em sua pesquisa sobre a construção de uma bancada didática de levitação magnética. Essas simplificações permitiram uma maior compreensão da dinâmica do sistema e sua resposta em diferentes situações.

$$\Delta \dot{x}_1 = \Delta x_2 \quad (3)$$

$$\Delta \dot{x}_2 = k_0 \Delta I + \lambda^2 \Delta x_1 \quad (4)$$

Em que  $k_0$  e  $\lambda^2$  estão expressos na Equação (5) e Equação (6).

$$\lambda^2 = -\frac{1}{m} \frac{\partial F(x_0, I_0)}{\partial x} = \frac{2kI_0^2}{m(x_0 + \mu)^3} \quad (5)$$

$$k_0 = -\frac{1}{m} \frac{\partial F(x_0, I_0)}{\partial I} = \frac{2kI_0}{m(x_0 + \mu)^2} \quad (6)$$

A técnica de linearização é comumente usada em sistemas não lineares para simplificar a análise e tornar o modelo mais manejável. No presente trabalho, ela foi aplicada com sucesso ao modelo descrito pela Equação (1), permitindo a obtenção das

equações de estado linearizadas e a análise do comportamento do sistema em torno do ponto de equilíbrio

### Controle aplicado

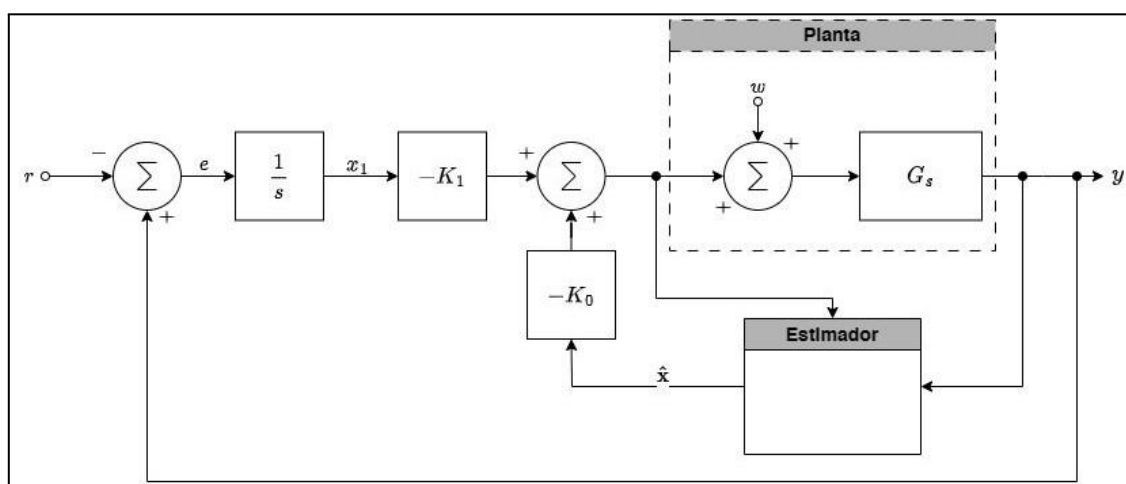
O controle via espaço de estados é uma técnica utilizada para projetar controladores para sistemas dinâmicos, permitindo uma modelagem mais geral e flexível do processo. No caso do modelo em questão, que apresenta não-linearidades, as equações de estado são uma ferramenta poderosa para a descrição do sistema (NISE, 2012).

Nesse contexto, o projeto de controle escolhido para o modelo foi um regulador integral, que apresenta vantagens como a capacidade de zerar o erro em regime permanente para uma entrada degrau e rejeição de distúrbios, aumentando a robustez do processo (FRANKLIN, 2013, p. 436).

Os pólos do regulador estão localizados a uma distância de três vezes o valor dos pólos dominantes do sistema em malha aberta. Dado que, a localização dos pólos do regulador em relação aos pólos dominantes da planta é um fator importante na estabilidade do sistema em malha fechada. Quando o polo do regulador está localizado a uma distância adequada em relação ao pólo dominante da planta, o controlador é capaz de atender aos requisitos de desempenho e estabilidade do sistema em malha fechada. Caso contrário, o sistema pode apresentar comportamento instável ou inadequado em malha fechada (FRANKLIN, 2013, p. 413).

A estrutura do controlador é apresentada na Figura 1.

Figura 1 – Estrutura de controle integral.



Fonte: Adaptado de FRANKLIN (2013)

Na Figura 1 ainda é ilustrado um observador de ordem completa, que tem como objetivo estimar as variáveis de estado que não podem ser obtidas a partir da planta (NISE, 2012). Os pólos do observador estão localizados a uma distância de oito vezes o valor do pólo dominante da planta. Foi considerado que os pólos do observador devem ser mais rápidos do que os do controlador, a fim de garantir que o erro de estimação dos

estados convirja para zero. Como resultado, espera-se que os pólos do controlador dominem a resposta do sistema (OGATA, 2010, p. 687).

A técnica de Ackermann foi empregada para realizar a alocação dos pólos do sistema de controle, com o objetivo de alcançar um desempenho desejado em malha fechada.

A equação de estados do controlador é apresentada na Equação (7).

$$\dot{\mathbf{z}} = \begin{bmatrix} -2q & 1 & 0 \\ -\lambda^2 - 3p^2 - q^2 & -3p & p^3 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 2q & 0 \\ -\lambda^2 + q^2 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ r \end{bmatrix} \quad (\text{X})$$

Portanto, dado que a planta em estudo é de segunda ordem, o sistema em malha fechada com regulador integral com três ganhos e observador com dois ganhos é de quinta ordem em espaço de estados.

### **Simulação numérica**

A simulação numérica de sistemas de controle é uma etapa importante no processo de projeto e análise de sistemas (OGATA, 2010, p. 668). Neste artigo, a simulação foi realizada usando o `solver_ivp` pertencente a sub biblioteca `scipy.integrate` disponível na biblioteca `SciPy`. De acordo com a documentação oficial da biblioteca `SciPy`, esse `solver` implementa o método de integração numérica chamado "IVP" (sigla em inglês para Problema de Valor Inicial), que é um método de passo variável baseado em Runge-Kutta (SciPy, 2023).

Além disso, foram adicionadas algumas funcionalidades extras para tornar a simulação mais realista. Primeiro, foi adicionado um ruído aleatório à saída do sistema para simular a incerteza presente em sistemas reais. Segundo, foram adicionados limitadores à entrada do controlador para garantir que a entrada do sistema permaneça dentro de limites físicos.

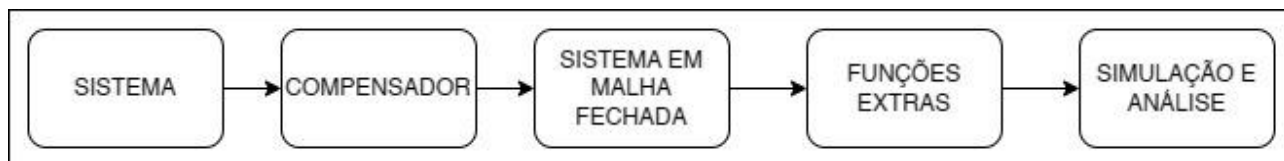
Para simular o sistema em malha fechada com regulador integral e observador em espaço de estados, as equações diferenciais do modelo foram definidas usando funções concebidas a partir de cálculos algébricos utilizando Python. Em seguida, foi utilizado o `solver_ivp` para integrar as equações diferenciais ao longo do tempo e obter a resposta do sistema.

Os ruídos foram gerados aleatoriamente usando a função `random` da biblioteca `NumPy`. Foram incluídos limitadores com o objetivo de simular a queda do cilindro quando ele saía da região de equilíbrio, permitindo que retornasse ao ponto inicial. Essa estratégia tornou a simulação mais eficiente em termos de processamento computacional.

A Figura 2 apresenta um fluxograma dos passos realizados para a criação da simulação.



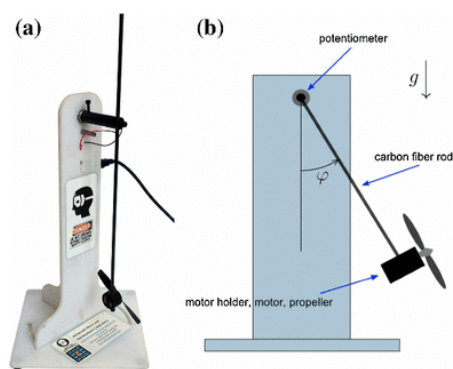
Figura 2 – Fluxograma dos passos até a simulação.



Fonte: Elaborado pelos autores.

## 2.2 Módulo 2 - Aeropêndulo

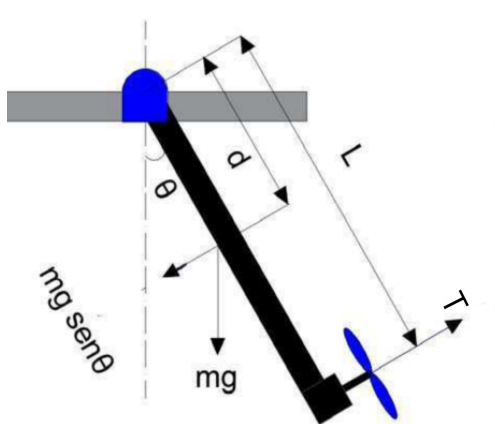
O aeropêndulo é um dispositivo utilizado em experimentos de física que combina os princípios de um pêndulo e da aerodinâmica. Ele consiste em uma haste suspensa por um ponto fixo em um suporte, com uma asa em uma das extremidades da haste.



Fonte: Habib at al, (2023)

Quando o aeropêndulo é movido para um lado, a força da gravidade faz com que a haste comece a oscilar em torno do ponto de suspensão. A asa no final da haste também gera uma força aerodinâmica que pode afetar o movimento do pêndulo. A interação entre as forças gravitacionais e aerodinâmicas faz com que o aeropêndulo execute movimentos complexos e interessantes.

### Modelagem do sistema físico



Fonte: xxx xxx at al, (2023)



Usando as leis de Newton e momentos angulares podemos encontrar o modelo matemático que descreve a dinâmica do aeropêndulo, a equação que modela o sistema foi retirada de MOHAMMADBAGHERI et al.

$$T = J\ddot{\theta} + c\dot{\theta} + mgd \sin(\theta) \quad (xx)$$

Para controlar o ângulo o braço do aeropêndulo a partir da tensão aplicada aos terminais do motor CC, deve se obter uma relação entre a tensão  $V$  com o empuxo  $T$  gerado pelas hélices do motor, essa relação é não linear, porém é possível aproximar por uma relação linear, como mostra a expressão (xx).

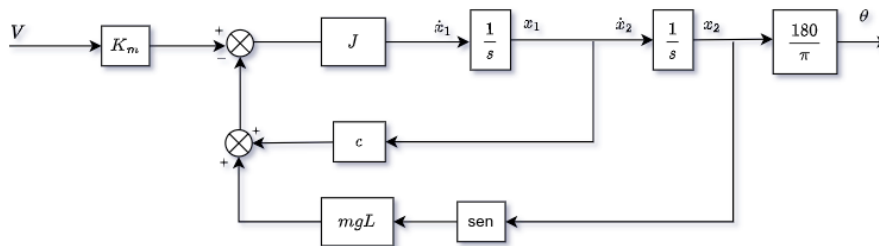
$$T \approx K_m V \quad (xx)$$

Assim, ao substituir em (xx1), se obtêm a expressão (xx) que modela o sistema tendo como entrada a tensão nos terminais e como saída o ângulo do aeropêndulo.

$$K_m V = J\ddot{\theta} + c\dot{\theta} + mgd \sin(\theta) \quad (xx)$$

Diagrama de blocos do modelo.

Figura xx: Diagrama de blocos do modelo do Aeropêndulo.



Fonte: Autor, (2023)

A partir do diagrama de blocos do modelo é possível encontrar as equações de estados do sistema, assim, se obtêm as expressões (xx1), (xx2) e (xx3).

$$\dot{x}_2 = x_1 \quad (xx1)$$

$$x_1 J = -x_1 c - mgd \sin(x_2) + V K_m \quad (xx2)$$

$$\dot{x}_1 = \frac{-x_1 c - mgd \sin(x_2) + V K_m}{J} \quad (xx3)$$

Em que:

- **T**: Empuxo gerado pela hélice;
- **J**: Momento de inércia;
- **$\theta$** : posição angular do Aeropêndulo;
- **c**: coeficiente de amortecimento viscoso;
- **m**: peso do Aeropêndulo;
- **d**: a distância entre o centro de massa e o ponto de pivô;
- **V**: Tensão de Entrada do Motor CC Série;
- **Km**: Relação entre o torque e a tensão;
- **x1** e **x2**: Estados do Sistema;

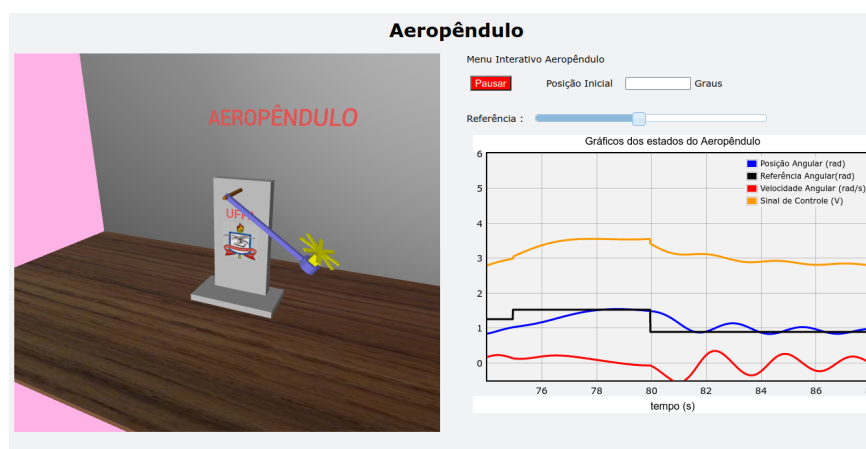
### Função de Transferência do Controlador

Para ser possível realizar projetos de controle usando técnicas da área de Sistemas de Controle é preciso linearizar o modelo e aplicar a transformada de Laplace. Uma das formas de linearizar o aeropêndulo é considerar **sen( $\theta$ )** sendo o próprio ângulo  **$\theta$** , essa técnica de linearização é válida para descrever o sistema para pequenas variações em torno de  **$\theta$** .

### Simulação Numérica e Gráfica

O processo de simulação tem duas partes, a gráfica e a numérica, sendo que a parte gráfica usa os dados numéricos para realiza a dinâmica do sistema, isso é interessante para mostrar de uma forma visual o comportamento de um sistema real a partir de cálculos computacionais.

Figura xx: Simulador Aeropêndulo.



Fonte: Autor, (2023)

A parte de simulação gráfica, tem dois componentes principais, sendo a animação 3D e os gráficos dos estados do sistema. Dessa forma, é possível observar o comportamento do sistema simulado a partir de diferentes perspectivas. Além disso, existe a simulação numérica, para essa parte foi criada uma Classe do Python que implementa as equações de estados do sistema e resolve numericamente.

### 3 CONCEITOS UTILIZADOS EM AMBOS OS MÓDULOS

A simulação virtual é uma ferramenta cada vez mais utilizada para auxiliar no ensino de conceitos complexos em diversas áreas, e na engenharia de controle não é diferente. A integração de teoria e prática é essencial para um bom aprendizado, e a simulação virtual permite essa união de forma eficiente e dinâmica. Nesse contexto, este capítulo apresenta a implementação de módulos de simulação virtual em Python para auxiliar no ensino de conceitos de controle em disciplinas de graduação em engenharia.

Foram utilizadas diversas bibliotecas, como NumPy, SciPy, Control e Vpython, para desenvolver as animações em 3D interativas, que permitem uma melhor visualização dos conceitos e sua aplicação em problemas reais. O capítulo apresenta a linguagem utilizada (Python) e suas bibliotecas, além de discutir a efetividade da utilização de simulações virtuais no ensino de controle em engenharia.

#### 3.1 Linguagem Python e suas bibliotecas

De acordo com Python.org (2023), a linguagem de programação Python foi criada na década de 1990 por Guido van Rossum enquanto trabalhava no Stichting Mathematisch Centrum (CIT) nos Países Baixos. Trata-se de uma linguagem de alto nível, sua simplicidade e flexibilidade tornam-na uma escolha popular para diversas aplicações em ciência de dados, inteligência artificial, análise de dados, entre outras áreas. Além disso, a grande quantidade de bibliotecas disponíveis para Python permite que os usuários tenham acesso a uma ampla gama de ferramentas prontas para uso, o que agiliza o desenvolvimento de projetos e aumenta a produtividade. Essas características fazem do Python uma das linguagens de programação mais populares e amplamente utilizadas em todo o mundo (MENEZES, 2010).

Neste trabalho, serão exploradas e destacadas algumas bibliotecas que foram utilizadas no desenvolvimento dos módulos para a implementação de soluções e simulações nos sistemas em estudo. Daremos especial ênfase às bibliotecas:

**NumPy:** é frequentemente utilizada para realizar operações matemáticas sobre conjuntos de dados, tais como cálculo de estatísticas, álgebra linear, processamento de sinais, entre outras aplicações. Ela também é utilizada em conjunto com outras bibliotecas, como Matplotlib, SciPy e Pandas, para realizar análises e visualizações de dados (NUMPY, 2023).

**SciPy:** é uma ferramenta essencial para aqueles que trabalham com análise de dados, processamento de sinais e resolução de equações diferenciais. Sua ampla gama de funções permite resolver problemas complexos de maneira eficiente e escalável (SCIPY, 2023).

**Control:** é uma extensão da linguagem Python que fornece ferramentas para análise e design de sistemas de controle. Ela é amplamente utilizada em aplicações de engenharia, em particular na área de controle automático de processos (PYTHON-CONTROL, 2023).

**SymPy:** é frequentemente utilizada em aplicações de matemática, física e engenharia, onde cálculos simbólicos são necessários para modelagem e análise de sistemas complexos (SYMPY, 2023).

**Matplotlib:** é uma das bibliotecas de visualização de dados mais utilizadas na linguagem de programação Python. Ela fornece ferramentas para a criação de gráficos 2D e 3D, histogramas, diagramas de dispersão, entre outros tipos de visualizações (MATPLOTLIB, 2023).

Adicionalmente, além de comandos nativos da linguagem, foram criadas funções e classes customizadas que auxiliaram na implementação e simulação dos modelos.

### 3.2 A biblioteca VPython

O VPython é uma biblioteca de visualização em 3D para a linguagem de programação Python, que permite a criação de animações e interações com objetos em tridimensionais em tempo real (GLOWSCRIPT, 2023). Com sua ampla gama de recursos e suporte para interações do usuário, o VPython é uma ferramenta valiosa para a exploração e compreensão de dados em 3D. A Figura 3 ilustra a interface do MAGLEV criada com o VPython.

Figura 3 – Interface MAGLEV.



Fonte: Elaborado pelos autores.

As interfaces gráficas dos módulos foram projetadas de forma a permitir ao usuário interagir diretamente com o sistema simulado. Além dos botões para alterar os parâmetros de simulação, foram incluídos gráficos para análises mais detalhadas do comportamento do sistema. Dessa forma, o usuário pode observar o sistema em ação e, ao mesmo tempo, visualizar os dados gerados pela simulação. Essa abordagem possibilita uma compreensão mais clara e dinâmica dos conceitos abordados durante as disciplinas de controle na engenharia.

### 3.3 Proposta pedagógica

Segundo os Indicadores de Fluxo da Educação Superior do INEP, os cursos de engenharia da UFPA apresentaram uma taxa média de desistência acumulada de 11% no período entre 2017 e 2021 (INEP, 2023). Esses dados indicam a presença de diversos fatores que podem influenciar na evasão dos estudantes de graduação, tais como baixo desempenho escolar, condições socioeconômicas desfavoráveis (que afetam especialmente os representantes das classes sociais mais baixas) e pertencer a minorias étnicas.

Frequentemente, os estudantes investem significativo tempo e esforço em seus estudos, porém, nem sempre alcançam resultados satisfatórios em termos de aprendizado, o que pode levar à frustração e afetar sua autoestima (BUNGÄU, POP, BORZA, 2017).

Diante desse cenário, a utilização de simulações pode ser uma importante ferramenta pedagógica para facilitar a compreensão de conceitos abstratos e complexos, especialmente na área de controle na engenharia. Nesse sentido, a biblioteca Vpython pode ser utilizada para criar simulações visuais e interativas de sistemas físicos, contribuindo para o aprimoramento do processo de ensino. (UTIEL; KIRNER, 2009)

Através da simulação, os estudantes poderão observar o comportamento de sistemas reais sob diferentes condições e parâmetros, permitindo uma análise mais clara e dinâmica do comportamento desses sistemas. Além disso, a interface gráfica da biblioteca Vpython possibilita a interação direta do usuário com o sistema simulado, permitindo a alteração de parâmetros e condições em tempo real e observando os efeitos dessas mudanças na resposta do sistema.

Outra vantagem do uso da biblioteca Vpython é a facilidade de programação, já que ela possui uma sintaxe simples e intuitiva, permitindo aos estudantes aprenderem a criar simulações rapidamente, sem a necessidade de conhecimentos avançados de programação.

Por fim, o uso de simulações de sistemas físicos com a biblioteca Vpython pode auxiliar na formação de profissionais mais capacitados e preparados para lidar com problemas reais na área de controle na engenharia, além de tornar o ensino mais atrativo e dinâmico para os estudantes. (VARGAS; ALVARADO, 2021)

Dessa forma, a proposta pedagógica consiste em utilizar a biblioteca Vpython para criar simulações de sistemas físicos e utilizá-las como ferramenta de ensino nas disciplinas de controle na engenharia, com o objetivo de facilitar a compreensão dos conceitos abordados e formar profissionais mais capacitados para lidar com problemas reais.

## 4 RESULTADOS

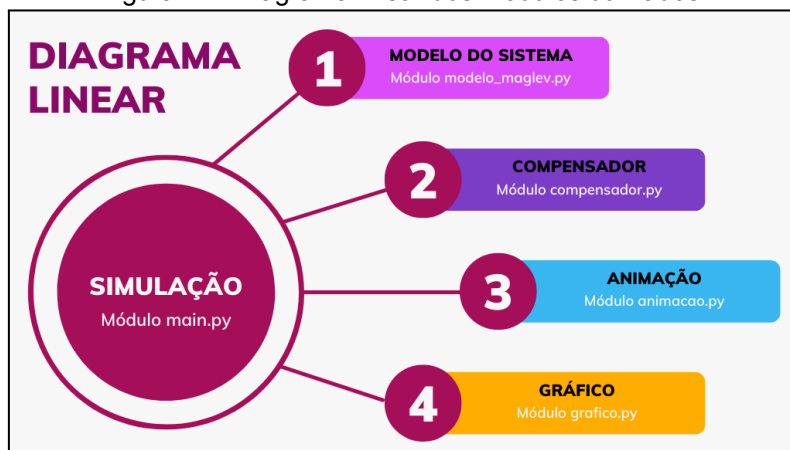
Nesta seção, serão apresentados os resultados obtidos por meio das simulações realizadas com a utilização da biblioteca VPython, a fim de possibilitar o ensino das disciplinas de controle de forma mais interativa e didática. O objetivo dessas simulações é proporcionar aos estudantes uma compreensão mais aprofundada sobre os conceitos teóricos estudados em sala de aula, através da visualização de fenômenos físicos em tempo real.

### 4.1 Módulo 1 - MAGLEV (Magnetic Levitator)

Utilizando a linguagem Python, foi possível estruturar o sistema através de módulos, que têm como objetivo organizar o código e torná-lo mais legível. Dentro desses arquivos são criadas classes, que são estruturas fundamentais para organizar e reutilizar

o código de maneira eficiente. Esses módulos são importados em um documento principal, onde são executados e implementados. Os módulos, ilustrados na Figura 4, desempenham um papel crucial na implementação e execução das simulações, tornando o processo mais eficiente e organizado.

Figura 4 – Diagrama linear dos módulos utilizados.



Fonte: Elaborado pelos autores.

O módulo `modelo_maglev.py` desempenha um papel fundamental na simulação do sistema maglev. Ele contém uma classe que implementa o modelo linearizado em espaço de estados do sistema, utilizando arrays para armazenar as informações necessárias. Além disso, o módulo possui uma função que recebe o modelo e o compensador projetado para o sistema, permitindo a simulação do sistema controlado. Um trecho do código é apresentado na Figura 5.

Figura 5 – Trecho de código do `modelo_maglev.py`.

```
# MODELAGEM -----
# %% Modelagem da Planta do MAGLEV
class Maglev:
    def __init__(self, m, k, mu, I0):
        self.m = m          # Massa em kg
        self.k = k          # Constante magnética em N m^2/A^2
        self.mu = mu        # Constante magnética em m
        self.g = 9.81       # Gravidade, em m/s^2

        self.I0 = I0        # Corrente de equilíbrio, em A

        # Posição de equilíbrio, a partir da corrente
        self.x0 = np.sqrt(k*I0**2/(m*self.g))-mu

        # Parâmetros linearizados
        self.lamda = np.sqrt(2*k*I0**2/(m*(self.x0+mu)**3))
        self.a = 2*k*I0/(m*(self.x0+mu)**2)
        self.A = np.array([[0, 1.], [self.lamda**2, 0]])
        self.B = np.array([[0], [-self.a]])
        self.C = np.array([[1., 0]])
```

Fonte: Elaborado pelos autores.

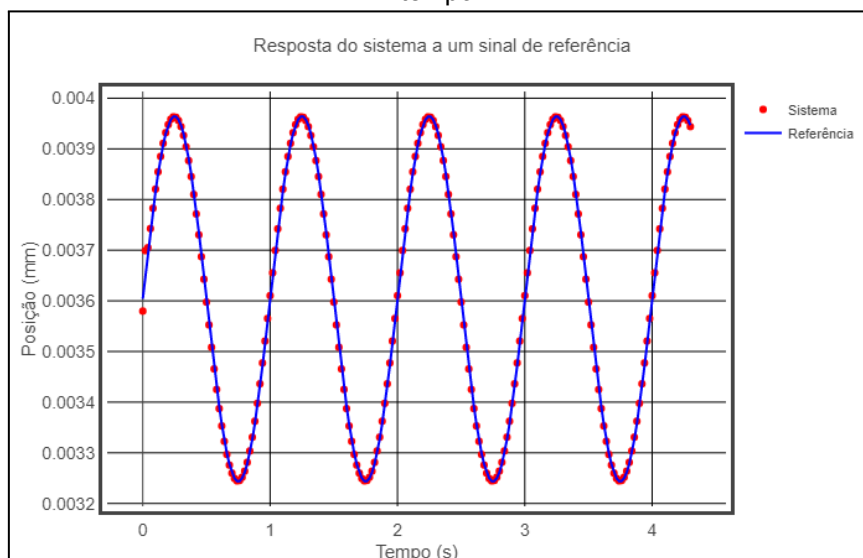
Um dos principais módulos desenvolvidos no projeto é o módulo `compensador.py`, que implementa funções para a declaração das matrizes do regulador integral e do observador. Essas matrizes são calculadas com o suporte da biblioteca `Control` e



utilizadas em conjunto com o modelo linearizado para simular o sistema. Além disso, foram definidos dois sinais de referência para o sistema: uma onda senoidal e uma onda quadrada, com o objetivo de oferecer ao usuário uma simulação mais abrangente e completa. Dessa forma, é possível visualizar como o sistema se comporta em diferentes condições de operação, o que é fundamental para o entendimento dos princípios de controle de sistemas físicos.

O módulo grafico.py foi desenvolvido com o objetivo de exibir um gráfico da posição do cilindro em função do tempo, permitindo que o usuário visualize e analise o comportamento do sistema. Para isso, foram criadas funções que facilitam a exibição dos dados de forma clara e organizada. Além disso, o módulo também é responsável por adicionar uma legenda flutuante na simulação, indicando a posição inicial do cilindro. Com essa funcionalidade, o usuário pode ter uma melhor compreensão sobre a simulação e os resultados obtidos, tornando a análise do sistema mais intuitiva e completa.

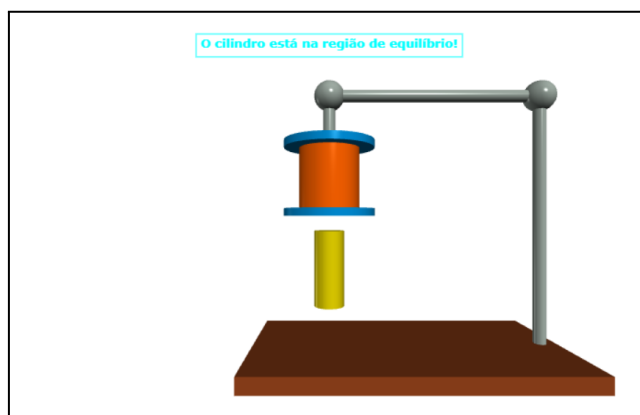
Figura 6 – Resposta do sistema em função do tempo.



Fonte: Elaborado pelos autores.

O módulo animacao.py foi criado com o objetivo de construir o sistema físico que será apresentado ao usuário, utilizando funções que criam objetos como a mesa, a estrutura e o cilindro flutuante, com o auxílio da biblioteca VPython. Adicionalmente, um menu interativo foi implementado na simulação, permitindo que o usuário possa modificar a frequência e a amplitude do sinal, bem como selecionar um sinal de referência. Outra funcionalidade importante que foi adicionada é a possibilidade de inserir um controlador personalizado, por meio do fornecimento das matrizes de estados. Dessa forma, o usuário pode testar diferentes configurações do sistema e avaliar os resultados de forma interativa.

Figura 7 – Bancada MAGLEV 3D.

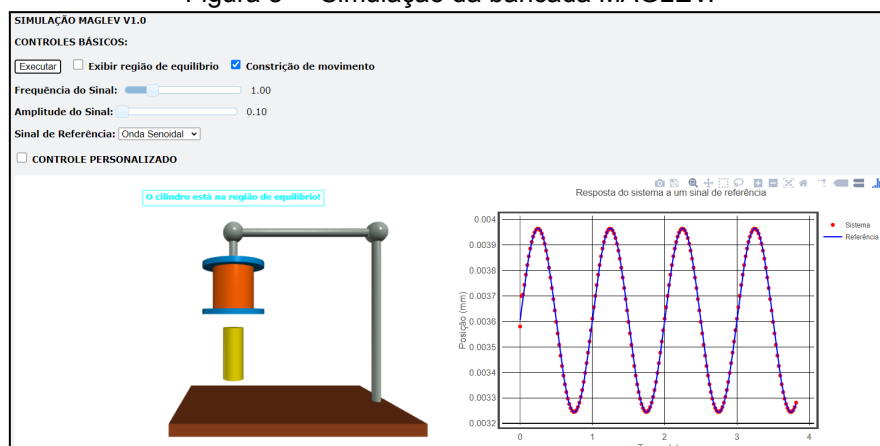


Fonte: Elaborado pelos autores.

O módulo `main.py` tem como função principal importar os outros módulos e executar um loop `while` que contém estruturas condicionais para criar legendas que representam o estado atual do cilindro flutuante na simulação. Nele também são definidos os sinais de referência para rastreamento, que podem ser ondas senoidais ou quadradas, além de ser implementada a função de ajuste das coordenadas do modelo às coordenadas do Vpython, assim como a adição de ruído gaussiano. Para tornar a simulação mais realista, uma função foi incluída para adicionar ruído gaussiano à saída do sistema, utilizando a função `np.random.normal` da biblioteca Numpy. Essa abordagem visa fornecer uma simulação mais precisa e próxima da realidade do sistema maglev. O loop utiliza o `solver_ivp` da biblioteca Scipy, que é responsável por fornecer a posição do cilindro em função do tempo, a partir da resolução do sistema para um sinal de referência escolhido pelo usuário.

O sistema desenvolvido é uma excelente ferramenta didática para o ensino de controle de sistemas físicos. Na simulação do sistema MAGLEV, foi possível observar o comportamento do cilindro flutuante em função do tempo, de acordo com diferentes sinais de referência e controladores implementados pelo usuário. Os resultados obtidos foram coerentes com a teoria estudada, o que demonstra a eficácia da simulação como uma ferramenta de ensino. Na Figura 8 pode ser visto o funcionamento da simulação.

Figura 8 – Simulação da bancada MAGLEV.



Fonte: Elaborado pelos autores.

Com o intuito de disponibilizar essa ferramenta aos discentes, será criado um Jupyter notebook que explicará os conceitos necessários para a compreensão da simulação. Esse notebook será disponibilizado em um ambiente virtual de aprendizado, onde os alunos poderão acessá-lo e executar a simulação em seus próprios computadores.

O Jupyter notebook será organizado em seções que apresentam os conceitos teóricos fundamentais para o entendimento da simulação. Essas seções incluirão explicações sobre o modelo linearizado em espaço de estados, o regulador integral, o observador e o compensador projetado. Além disso, serão apresentados os conceitos básicos de controle de sistemas físicos, como estabilidade, margem de ganho e margem de fase.

O objetivo é que o aluno possa entender a teoria por trás do sistema e, em seguida, aplicar esse conhecimento na simulação. Para isso, serão disponibilizadas instruções passo a passo para executar a simulação, bem como dicas e sugestões para a análise dos resultados obtidos.

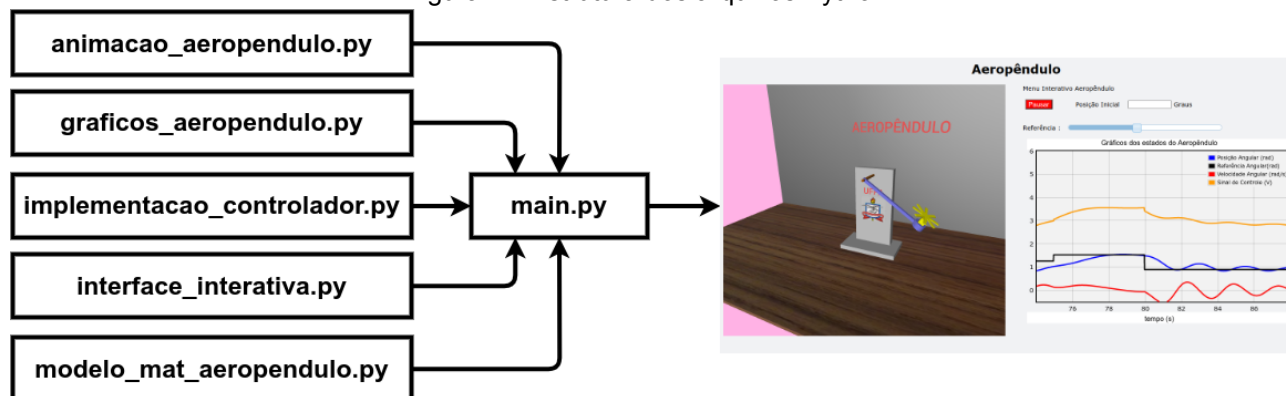
A disponibilização desta ferramenta didática em um ambiente virtual permitirá aos alunos acessá-la em qualquer lugar e a qualquer momento, tornando o aprendizado mais flexível e personalizado. Além disso, a execução da simulação em seus próprios computadores permitirá aos alunos experimentar diferentes configurações do sistema e avaliar os resultados de forma interativa, aprimorando ainda mais seu conhecimento em controle de sistemas físicos.

### 3.2 Módulo 2 - Aeropêndulo

O desenvolvimento do simulador para o aeropêndulo teve êxito. Com as equações obtidas, partiu-se para a implementação dos códigos usando a linguagem Python, para isso, usou-se a estrutura de programação orientada a objeto, em que se criaram classes que implementam as funcionalidades do simulador.

A Figura (XX) permite visualizar a estrutura do projeto, em que, as classes são criadas em arquivos **.py** diferentes sendo importadas em um arquivo principal que orchestra as classes para se obter o algoritmo que implementa o simulador.

Figura xx: Estrutura dos arquivos Python.



Fonte: Autor, (2023)

Descrição dos módulos do simulador:

- **animacao\_aeropendulo.py:** Usa a biblioteca Vpython para construir a estrutura 3D do simulador.
- **graficos\_aeropendulo.py:** Usa a biblioteca Vpython para implementar os gráficos dos estados do sistema.
- **implementacao\_controlador.py:** Módulo responsável por construir os controladores e disponibilizar o sinal de controle.
- **interface\_interativa.py:** Classe responsável por criar o botão, slide, labels e input da interface gráfica que permite a interação do usuário com o simulador.
- **modelo\_mat\_aeropendulo.py:** Módulo que implementa as equações de estados do sistema real, e retorna a solução, que é usada para realizar a dinâmica do sistema e plotar os gráficos dos sinais do sistema.
- **main.py:** Módulo principal que importa todos os outros módulos e realiza a interação entre simulador numérico e gráfico.

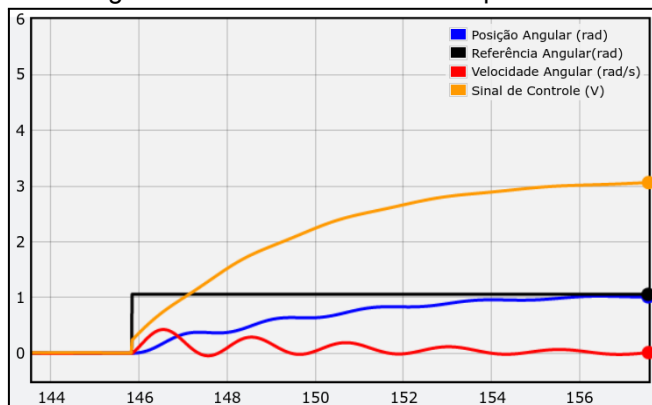
Figura 8 – Simulador 3D do Aeropêndulo.



Fonte: Elaborado pelos autores.

Foi exitosa a construção do simulador 3D, com ele foi possível observar a dinâmica do aeropêndulo assim como no mundo real, dessa forma, possibilitando uma outra maneira de observar a dinâmica de um modelo matemático.

Figura 8 – Simulador 3D do Aeropêndulo.



Fonte: Elaborado pelos autores.

A partir da simulação numérica, os dados dos estados do sistema, assim como, os sinais de controle e referência, foram plotados para visualizar o comportamento dos sinais em função do tempo.

A Figura XX permite visualizar os gráficos para uma simulação em malha fechada com um controlador Proporcional Integral, dessa forma, pode-se observar que o sistema se comporta de forma coerente com a teoria de controle, em que, para controladores PI, o erro em regime permanente é nulo, isso é possível de constatar ao passo que o sinal de saída se desloca em direção do sinal de referência e em regime permanente os sinais se sobrepõem.

### 3. CONCLUSÃO

O uso de simulações é uma maneira eficaz de ensinar conceitos complexos e abstratos em disciplinas de controle. Neste artigo, foram apresentados dois exemplos de simulações desenvolvidas com o objetivo de auxiliar o ensino de controle: o MAGLEV e o Aeropêndulo. Ambas foram implementadas utilizando a biblioteca Vpython e contém recursos como gráficos, animações e interatividade para tornar o aprendizado mais dinâmico e atraente.

O sistema MAGLEV simula um trem levitando magneticamente e possui um controlador para manter a posição do trem constante. Já o Aero Pêndulo simula um pêndulo invertido que é controlado para permanecer em equilíbrio. Ambos os sistemas apresentam desafios interessantes em termos de controle, e as simulações permitem que os alunos experimentem diferentes estratégias de controle e avaliem seus resultados.

Além disso, foi discutido o uso de diferentes módulos para construir as simulações, como os módulos de animação, gráfico e compensador, que permitem a construção de sistemas mais complexos e realistas. A utilização de bibliotecas como a Control, Numpy, Scipy e Vpython, foi fundamental para a implementação das simulações.

Para tornar essas simulações acessíveis aos discentes, foi escolhido o formato de distribuição através de Jupyter Notebooks, que são ambientes interativos de programação e análise de dados. Os Jupyter Notebooks permitem que os usuários visualizem, executem e interajam com as simulações em tempo real, além de oferecerem uma plataforma para explicar os conceitos teóricos necessários para o entendimento das simulações.

Com essa abordagem, os alunos podem ter uma experiência mais prática e visual sobre os conceitos de controle, o que pode ajudar a consolidar o aprendizado e aumentar o engajamento. Dessa forma, a utilização de simulações com a biblioteca Vpython distribuídas em Jupyter Notebooks se apresenta como uma alternativa interessante para o ensino de controle de sistemas físicos, proporcionando uma experiência mais completa e interativa para os alunos.

### 4. AGRADECIMENTOS

### 5. REFERÊNCIAS

Biblioteca Python Control. Disponível em:  
<https://python-control.readthedocs.io/en/0.9.3.post2/>. Acesso em: 05 mai. 2023.

BUNGĂU, Constantin; POP, Adrian Petru; BORZA, Adriana. **Dropout of first year undergraduate students: A case study of engineering students**. In: Balkan Region Conference on Engineering and Business Education. 2017. p. 349-356.

COSTA, Jefferson S.; SILVA, Cleison D.; TEIXEIRA, Raphael B. **Modelagem de sistema de levitação magnética por abordagem analítica baseada no conceito de circuitos magnéticos**. Tucuruí: Cobenge, 2012.

FRANKLIN, Gene F.; POWELL, J. David; EMAMI-NAEINI, Abbas. **Sistemas de controle para engenharia**. Bookman Editora, 2013.

GlowScript Team. GlowScript Documentation. Disponível em:  
<https://www.glowscript.org/docs/VPythonDocs/index.html>. Acesso em: 05 mai. 2023.

INEP: Indicadores de Fluxo da Educação Superior. Disponível em:  
<https://www.gov.br/inep/pt-br/aceso-a-informacao/dados-abertos/indicadores-educacionais/indicadores-de-fluxo-da-educacao-superior>. Acesso em: 06 mai. 2023.

MATPLOTLIB. Homepage. Disponível em: <https://matplotlib.org/>. Acesso em: 06 mai. 2023.

MENEZES, Nilo Ney Coutinho. **Introdução a programação com Python**. São Paulo: Novatec, 2010.

MILHOMEM, Rômulo L. **Construção de uma Bancada Didática de Processo de Levitação Magnética-MAGLEV**. Trabalho de Conclusão de Curso. Universidade Federal do Pará, Tucuruí-PA, 2010.

NISE, N. S. **Engenharia de sistemas de controle**. 6 ed. São Paulo: Editora LTC, 2012.

NumPy. NumPy Documentation. Disponível em: <https://numpy.org/doc/stable/>. Acesso em: 05 mai. 2023.

OGATA, Katsuhiko et al. **Modern control engineering**. Upper Saddle River, NJ: Prentice hall, 2010.

SciPy community. SciPy v1.6.2 Reference Guide. Disponível em:  
<https://docs.scipy.org/doc/scipy/reference/>. Acesso em: 05 mai. 2023.

SYMPY. SymPy 1.9 Documentation. Disponível em: <https://www.sympy.org/en/index.html>. Acesso em: 06 mai. 2023.

UTIEL, Wagner; KIRNER, Claudio. **Realismo Visual em Ambientes Interativos como Ferramentas de Auxílio à Educação**. RENOTE, v. 7, n. 1, 2009.

VARGAS, Francisco Javier Triveno; ALVARADO, Hugo Siles. **Virtual laboratories as strategy for teaching improvement in Math Sciences and Engineering in Bolivia**. Cadernos de Educação Tecnologia e Sociedade, v. 14, n. 2, p. 187-196, 2021.



"ABENGE 50 ANOS: DESAFIOS DE ENSINO, PESQUISA E  
EXTENSÃO NA EDUCAÇÃO EM ENGENHARIA"

18 a 20 de setembro  
Rio de Janeiro-RJ



**COBENGE**

**2023**

*51º Congresso Brasileiro de Educação em Engenharia*

*VI Simpósio Internacional de Educação em Engenharia*

Realização:



Organização:

