

Explorando o potencial do Laboratório Virtual de Controle de Sistemas com VPython no ensino de Engenharia: Análise e perspectivas para o futuro

DOI: 10.37702/2175-957X.COBENGE.2023.4253

Yuri de Oliveira Cota - yuricota9@gmail.com
Universidade Federal do Pará

Oséias Dias de Farias - oseias.farias@tucurui.ufpa.br
Universidade Federal do Pará

Raphael Barros Teixeira - raphaelbt@ufpa.br
Universidade Federal do Pará

Rafael Suzuki Bayma - rafaelbayma@gmail.com
Universidade Federal do Pará

Resumo: O presente artigo descreve o desenvolvimento de simulações de controle de sistemas por meio da biblioteca VPython e da plataforma PyPi. Os resultados das simulações evidenciam a efetividade do uso dessas ferramentas no processo de ensino de Engenharia, fornecendo uma experiência prática e interativa para os alunos. A simulação do MAGLEV e do Aero Pêndulo foram desenvolvidas e seus resultados foram apresentados e discutidos em detalhes. A partir dos resultados obtidos, concluiu-se que as simulações são eficazes para o ensino de controle de sistemas, pois permitem que os alunos visualizem e compreendam de forma prática os conceitos teóricos aprendidos em sala de aula. Futuramente, espera-se expandir a utilização dessas ferramentas para outras áreas do conhecimento, bem como implementar novas simulações para enriquecer ainda mais o processo de ensino-aprendizagem.

Palavras-chave: controle de sistemas, simulação, VPython, PyPi, educação em engenharia.

Explorando o potencial do Laboratório Virtual de Controle de Sistemas com VPython no ensino de Engenharia: Análise e perspectivas para o futuro

1 INTRODUÇÃO

O ensino de disciplinas teóricas nos cursos de Engenharia é tradicionalmente um desafio, pois muitos alunos têm dificuldade em relacionar o conteúdo com o comportamento físico real dos sistemas estudados. Embora a introdução de tecnologias de apoio tenha ajudado a amenizar o problema, ainda existem desafios, como a falta de estrutura operacional e atualização de metodologias de ensino. Uma forma de acelerar este processo é a utilização de simulação computacional auxiliada por animações.

Este trabalho tem como objetivo apresentar uma maneira de explorar conhecimentos de Engenharia sobre controle de sistemas físicos por meio de ferramentas computacionais para criar simulações e animações interativas. Foram desenvolvidas duas animações para sistemas clássicos utilizados em aulas de Engenharia de Sistemas de Controle que fazem parte do currículo de Engenharia Elétrica do Campus Tucuruí da Universidade Federal do Pará: o Aero Pêndulo e o Levitador Magnético. A proposta do trabalho é iniciar um conjunto de módulos programáveis em linguagem Python que possam ser aperfeiçoados e utilizados consistentemente em aulas de modo a melhorar a compreensão dos estudantes sobre o assunto, além de habilidades de programação e trabalho em equipe.

2 METODOLOGIA

As simulações desenvolvidas representam a dinâmica de sistemas físicos reais descritos em outros trabalhos. Neste sentido, são apresentados sucintamente os conceitos e técnicas empregados em cada módulo de simulação, assim como a abordagem computacional utilizada na implementação. O processo de modelagem e simulação foi desenvolvido em linguagem Python, utilizando tanto recursos numéricos, como o módulo NumPy (NUMPY, 2023) como simbólicos - módulo SymPy (SYMPY, 2023).

2.1 Módulo 1 - MAGLEV (*Magnetic Levitator – Levitação Magnética*)

O MAGLEV é um sistema que busca a suspensão de um corpo ferromagnético no ar através da utilização de uma força magnética gerada por um eletroímã. A dinâmica é equivalente à de diversos sistemas dinâmicos instáveis comumente vistos em nível de graduação. O MAGLEV também permite ilustrar aplicações sobre circuitos magnéticos, magnetização e estabilização de sistemas, além de possuir um bom apelo visual.

Modelagem do sistema físico

O modelo utilizado na simulação é do tipo espaço de estados (NISE, 2012). As equações são provenientes do estudo de Costa, Silva e Teixeira (2012), cuja derivação analítica a partir de circuitos magnéticos consegue representar satisfatoriamente o sistema real desenvolvido (MILHOMEM, 2010). Os parâmetros utilizados na simulação correspondem ao sistema MAGLEV citado a priori.

As equações de estado são descritas como Equação (1) e Equação (2).

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

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

Em que, x_1 e x_2 representam a posição e velocidade do corpo flutuante, respectivamente, I é a corrente no eletroímã, g é a aceleração da gravidade e os demais símbolos são constantes.

Para proceder com os métodos clássicos de Engenharia de Controle, o sistema é linearizado em torno de um ponto de equilíbrio (NISE, 2012). As equações de estado linearizadas, Equação (3) e Equação (4), consideram como ponto de equilíbrio os valores de posição e corrente do sistema (x_0, I_0).

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

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

Em que k_0 e λ^2 estão expressos na Equação (5) e Equação (6) (MILHOMEN, 2010).

$$\lambda^2 = -\frac{2kI_0^2}{m(x_0 + \mu)^3} \quad (5)$$

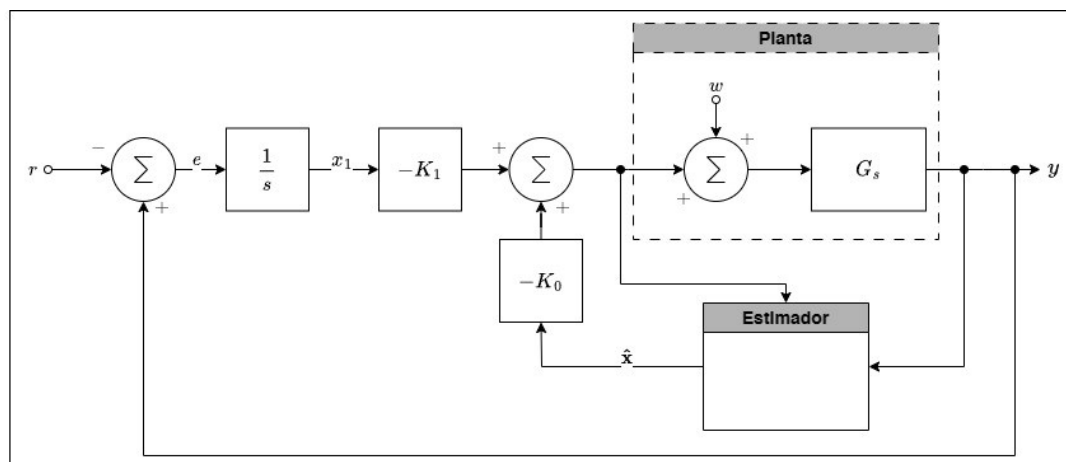
$$k_0 = -\frac{2kI_0}{m(x_0 + \mu)^2} \quad (6)$$

Ao seguir este roteiro teórico o estudante verifica a semelhança dinâmica deste sistema com outros clássicos da Engenharia de Controle, como o pêndulo invertido (OGATA, 2010, p. 63). É interessante pontuar que o todo o roteiro teórico foi desenvolvido utilizando matemática simbólica, que é um adicional importante ao arcabouço de ferramentas de Engenharia.

Controle aplicado

O controlador escolhido para estabilizar o sistema MAGLEV simulado foi um compensador integral (FRANKLIN, 2013) por realimentação de estados, que apresenta vantagens como a erro em regime permanente zero e rejeição de distúrbios do tipo degrau, aumentando a robustez do processo (FRANKLIN, 2013, p. 436). Por uma questão de simplicidade, os pólos do regulador foram escolhidos como sendo reais e iguais em $s = -p$, onde recomenda-se p três vezes maior em módulo que os pólos de malha aberta. Os pólos do estimador de estados foram escolhidos de forma semelhante em $s = -q$, onde q é escolhido em função de p . A estrutura do controlador é apresentada na Figura 1.

Figura 1 – Estrutura de controle integral.



Fonte: Adaptado de FRANKLIN (2013)

Vale frisar que o compensador implementado utiliza um observador de ordem completa (NISE, 2012). Devido ao sistema ser de ordem baixa, o projeto do controlador pôde ser feito de forma analítica, utilizando computação simbólica. A realização do compensador é feita através da equação de estados parametrizada em termos do sistema e das especificações de projeto, de acordo com a Equação (7) e a saída do controlador é dada pela Equação (8).

$$\dot{\mathbf{z}} = \begin{bmatrix} -2q & 1 & 0 \\ -\lambda^2 - 3p^2 - q^2 & -3p & p^3 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 2q & 0 \\ -\lambda^2 + q^2 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ r \end{bmatrix} \quad (7)$$

$$u = - \begin{bmatrix} \frac{-\lambda - 3p^2}{k_0} & \frac{-3p}{k_0} & \frac{p^3}{k_0} \end{bmatrix} \mathbf{z} \quad (8)$$

Em que, \mathbf{z} são os estados do compensador, $x_1 = x_1 - x_0$ é a saída do sistema linearizado para correção de estimativas e r é o sinal de rastreamento. Juntamente com a Equação (1) e Equação (2), a Equação (7) permite simular o sistema em malha fechada.

Simulação numérica

A simulação numérica de Sistemas de Controle é uma etapa importante no processo de projeto e análise na maioria dos trabalhos de Engenharia (OGATA, 2010, p. 668). Neste trabalho, a simulação consiste em resolver numericamente a Equação (1), Equação (2) e Equação (7), e utilizar os resultados para alimentar os módulos de animação gráfica. As funções Python utilizadas pertencem à biblioteca SciPy, que utiliza um solver de integração numérica baseado no algoritmo Runge-Kutta de passo variável (SCIPY, 2023).

Além das equações não-lineares e do controlador, foram adicionados recursos de programação para tornar a simulação um pouco mais realista: ruído branco Gaussiano foi adicionado ao sinal de saída para simular o ruído proveniente dos sensores reais da planta e limitadores de amplitude foram adicionados à entrada do controlador para garantir que a corrente simulada do eletroímã permaneça dentro de valores realistas.

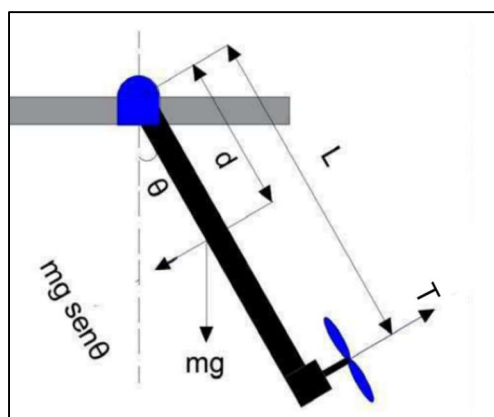
2.2 Módulo 2 – Aero Pêndulo

O Aero Pêndulo combina os princípios de aerodinâmica ao problema do pêndulo simples. Fisicamente, o sistema consiste de uma haste presa a um ponto fixo, com uma hélice em uma das extremidades. A hélice produz empuxo aerodinâmico que propulsiona o pêndulo a oscilar em modo forçado.

Modelagem do sistema físico

A Figura 2 apresenta um esquemático com os parâmetros do Aero Pêndulo.

Figura 2 – Esquemático com os parâmetros do Aero Pêndulo.



Fonte: Adaptado de SILVA (2018)

Usando as leis de Newton pode-se encontrar um modelo matemático que descreve o movimento do Aero Pêndulo, conforme a Equação (9) (MOHAMMADBAGHERI et al., 2011, p. 2).

$$T = J\ddot{\theta} + c\dot{\theta} + mgd \sin(\theta) \quad (9)$$

Em que, T é o empuxo gerado pela hélice, J é o momento de inércia, θ é a posição do Aero Pêndulo, c é o coeficiente de amortecimento viscoso, m é o peso do Aero Pêndulo e d é a distância entre o centro de massa e o ponto de pivô.

A hélice que constitui o atuador do sistema é normalmente conectada a um motor elétrico controlado por uma tensão V , cuja relação com o torque T é não linear, porém é possível aproximá-la razoavelmente por uma equação linear (AHMAD et al., 2021, p.1), como é apresentado na Equação (10).

$$T \approx K_m V \quad (10)$$

Desta forma, combinando a Equação (9) e Equação (10), obtém-se a Equação (11) que descreve o movimento do sistema, entre os sinais de entrada e saída de interesse.

$$K_m V = J\ddot{\theta} + c\dot{\theta} + mgd \sin(\theta) \quad (11)$$

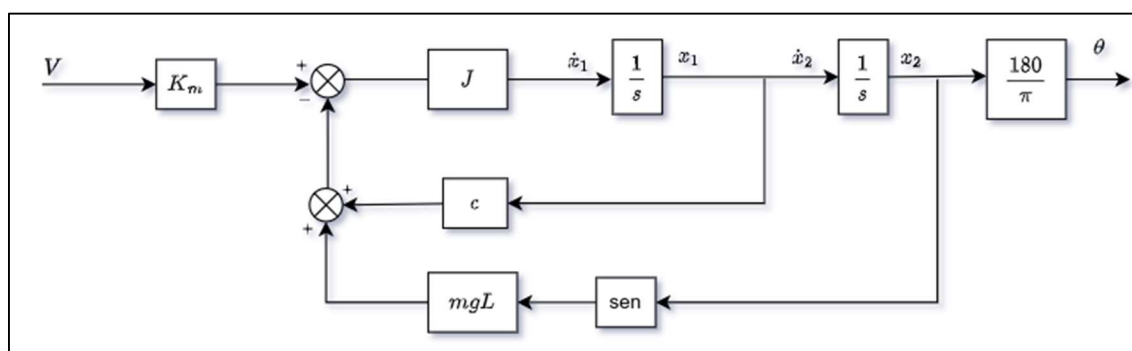
O diagrama de blocos ilustrado na Figura 3 permite a identificação das equações de estados do sistema, apresentadas na Equação (12) e Equação (13), que se mostram importantes para a simulação do sistema utilizando o solver numérico já descrito.

$$\dot{x}_2 = x_1 \quad (12)$$

$$\dot{x}_1 = \frac{-x_1 c - mgd \sin(x_2) + V K_m}{J} \quad (13)$$

Em que, V é tensão de entrada do motor CC série, K_m é a relação entre o torque e a tensão, x_1 e x_2 são os estados do sistema.

Figura 3 – Diagrama de blocos do modelo do Aero Pêndulo.



Fonte: Os Autores

Função de Transferência do Controlador

O modelo linearizado do Aero Pêndulo pode ser encontrado facilmente admitindo-se a aproximação de pequenos ângulos $\sin(\theta) \approx \theta$, o que resulta na Equação (14). A função de transferência do sistema na Equação (15) decorre diretamente da Equação (14).

$$K_m V = J\ddot{\theta} + c\dot{\theta} + mgd\theta \quad (14)$$

$$\frac{\theta(s)}{V(s)} = \frac{K_m/J}{s^2 + sc/J + mgd/J} \quad (15)$$

O controlador para o sistema Aero Pêndulo foi projetado utilizando o método do Lugar da Raízes (FRANKLIN, 2013, p. 190). Observou-se que um controle Proporcional-Integral (PI) simples serviria ao propósito da simulação. A função de transferência obtida é descrita como a Equação (16).

$$C(s) = \frac{0,2126s + 0,7893}{s} \quad (16)$$

Da mesma forma que o MAGLEV, o controlador é revertido para uma equação de estados que, juntamente com a Equação (12) e Equação (13), representam o sistema em malha fechada.

Simulação Numérica e Gráfica

A simulação do módulo Aero Pêndulo é dividida em duas partes: gráfica e numérica. A parte gráfica inclui animação 3D e gráficos dos sinais, permitindo análise do resultado em diferentes perspectivas. A parte numérica é implementada em uma classe Python que lida apenas com a solução numérica das equações de estados.

3 IMPLEMENTAÇÃO COMPUTACIONAL

Até este ponto, os sistemas foram tratados de forma diferente devido à derivação dos aspectos matemáticos serem bastante distintos. No entanto, a implementação computacional dos dois foi basicamente a mesma. A seguir são discutidos os aspectos utilizados para a realização dos módulos.

3.1 Linguagem Python e suas bibliotecas

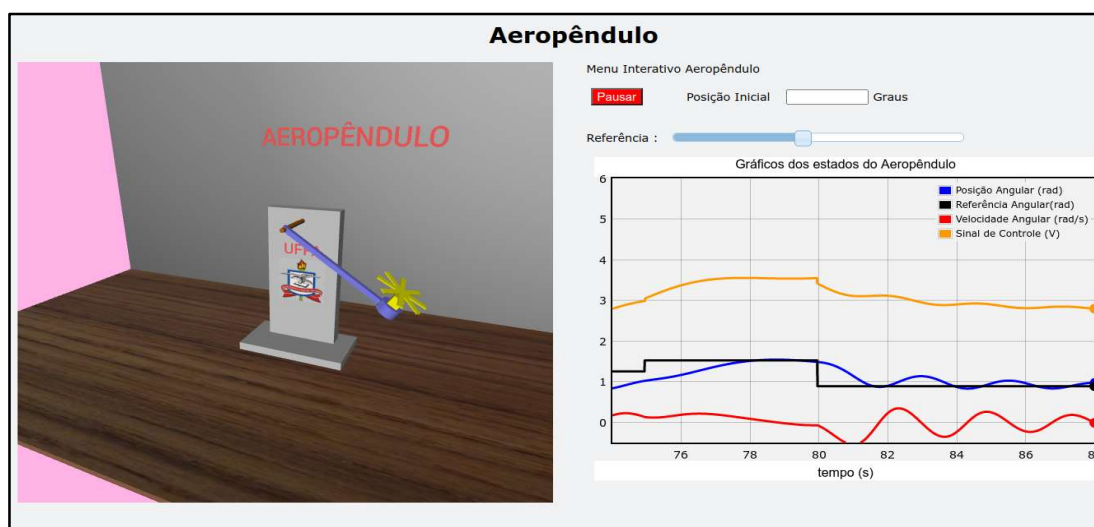
Conforme já comentado, a proposta do laboratório é majoritariamente desenvolvida na linguagem Python, do projeto teórico às animações. A linguagem de programação Python foi criada no início dos anos 90 por Guido van Rossum enquanto trabalhava no Centrum Wiskunde & Informatica (CWI), nos Países Baixos. 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. Essas características fazem do Python uma das linguagens de programação mais populares e amplamente utilizadas em todo o mundo (MENEZES, 2010).

Os principais pacotes utilizados no desenvolvimento deste estudo foram: NumPy, SciPy, SymPy e Control. Além disso, foram criadas funções e classes específicas às necessidades deste projeto, visando maior modularidade e melhor distribuição do código.

3.2 A biblioteca VPython

De particular destaque para o trabalho é a biblioteca VPython, onde foram desenvolvidas as animações 3D para visualização dos resultados de simulação, como pode ser visto na Figura 4.

Figura 4 – Simulador Aero Pêndulo.



Fonte: Os Autores

O VPython é uma biblioteca que permite a criação de animações interativas em tempo real (GLOWSCRIPT, 2023). Com uma ampla variedade de recursos, o VPython é uma ferramenta interessante para simulação de fenômenos físicos para fins pedagógicos.

As interfaces gráficas dos módulos desenvolvidos 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

3.3 Proposta pedagógica

Durante as aulas de Engenharia de Sistemas de Controle, muitos alunos reportam dificuldades de relacionar o conteúdo da disciplina aos sistemas físicos aos quais ela se propõe a discutir. Um exemplo recorrente é a dificuldade de interpretar fisicamente os gráficos cartesianos de sinais de saída e de controle do sistema. Neste sentido, a visualização do fenômeno torna-se ainda mais importante, principalmente quando se trata de sistemas onde os sinais representam movimento.

No entanto, a visualização do processo real normalmente requer disponibilidade de equipamento, o que normalmente esbarra em dificuldades de custo ou escala. Neste caso, a abordagem virtual é uma alternativa atraente, uma vez que permite a todos os alunos da turma visualizar relativamente bem o fenômeno, sem ônus material. A simulação é, obviamente, limitada em complexidade se comparada ao fenômeno real, mas melhorias gráficas progressivas podem torná-la cada vez mais convincente.

Inicialmente, a proposta pedagógica do laboratório virtual é estimular os estudantes a desenvolver módulos de código Python que permitam visualizar os resultados do projeto de controladores através de animações 3D desenvolvidas em VPython. A simplicidade da biblioteca permite que os estudantes realizem a tarefa sem se dedicar excessivamente à programação gráfica, sem perder o foco no assunto relativo à Engenharia. Além disso, os programas serão desenvolvidos em código aberto e ficarão disponíveis através de canais de distribuição acessíveis, por exemplo, o GitHub, o que permite que outros alunos construam versões melhoradas tomando trabalhos anteriores de seus pares como ponto de partida.

Os dois primeiros módulos apresentados até aqui constituem o estágio piloto da proposta. Eles foram desenvolvidos por estudantes que já passaram pelas disciplinas de Sistemas de Controle, mas relataram ainda possuir dúvidas semelhantes ao que já foi exposto. Através da experiência de desenvolvimento, além de reforçarem os conceitos sobre Sistemas de Controle, os alunos tiveram oportunidade de conhecer ferramentas de animação e exercitar habilidades como trabalho em equipe, programação científica e controle de versão. Espera-se que, através da replicação e aperfeiçoamento desta metodologia ao longo dos próximos semestres (possivelmente ofertando aos alunos das turmas em curso), a oferta das disciplinas se torne mais dinâmica no aprendizado do conteúdo.

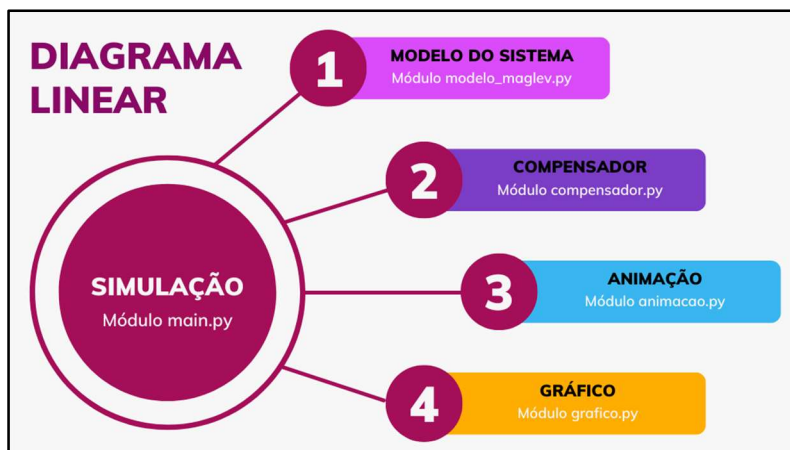
4 RESULTADOS

Nesta seção são apresentados alguns dos resultados obtidos nesta primeira iteração do projeto do laboratório virtual de sistemas controle.

4.1 Módulo 1 - MAGLEV

A simulação do sistema MAGLEV foi desenvolvida em módulos de programas, ilustrados na Figura 5.

Figura 5 – Diagrama linear dos módulos utilizados.



Fonte: Os Autores

O módulo "*modelo_maglev.py*" desempenha um papel fundamental no estudo do sistema MAGLEV, fornecendo o modelo embutido necessário para análises e simulações. Esse módulo encapsula todas as informações relevantes do sistema, como parâmetros, equações e comportamento dinâmico. Ele também é responsável pelo cálculo das equações de estado em malha fechada.

O módulo "*compensador.py*" realiza o projeto numérico de um compensador para um sistema de controle, utilizando técnicas como ganho integral e observador para estabilizar o sistema em malha fechada. Essa classe encapsula o processo de projeto do compensador, fornecendo as matrizes e vetores necessários para a implementação do compensador no sistema de controle.

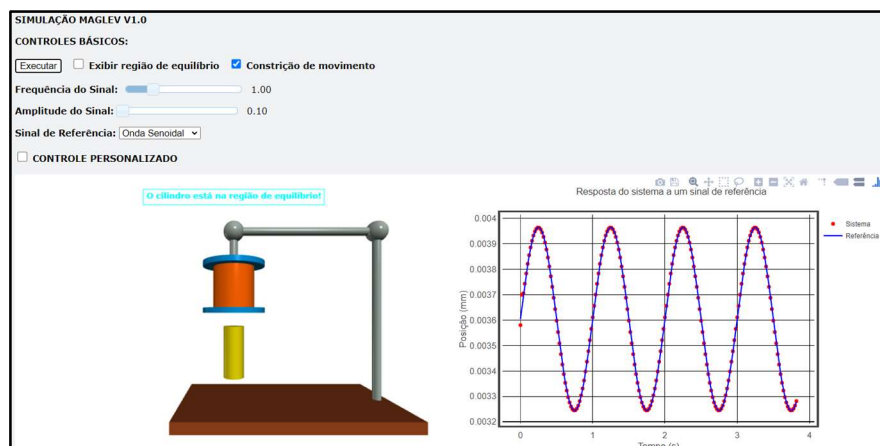
O módulo "*grafico.py*" foi desenvolvido com o objetivo de exibir um gráfico cartesiano do sinal da posição do cilindro em função do tempo, permitindo que o usuário visualize e analise o comportamento do sistema da forma usual como é apresentado em livros textos sobre o assunto.

O módulo "*animacao.py*" foi criado com o objetivo de construir o ambiente 3D que é apresentado ao usuário. O módulo cria objetos como a mesa, a estrutura e o cilindro flutuante, todos componentes da biblioteca VPython. Adicionalmente, diversos controles interativos foram adicionados à interface, permitindo que o usuário possa modificar a frequência e a amplitude do sinal, bem como selecionar diferentes tipos de rastreamento. Outra funcionalidade importante que foi adicionada é a possibilidade de inserir um controlador personalizado, por meio do fornecimento das matrizes de estados.

O módulo "*main.py*" desempenha um papel central no programa, pois é responsável por importar os outros módulos e executar a simulação do sistema. Além disso, ele também define os sinais de referência utilizados para o rastreamento, incluindo um sinal senoidal e uma onda quadrada. Uma funcionalidade importante implementada nesse módulo é a adição de ruído à saída do sistema, a fim de tornar a simulação mais realista. Essa função permite introduzir variações aleatórias na saída do sistema, simulando as perturbações e incertezas presentes em situações reais.

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. Na Figura 6 pode ser visto o funcionamento da simulação.

Figura 6 – Simulação da bancada MAGLEV.

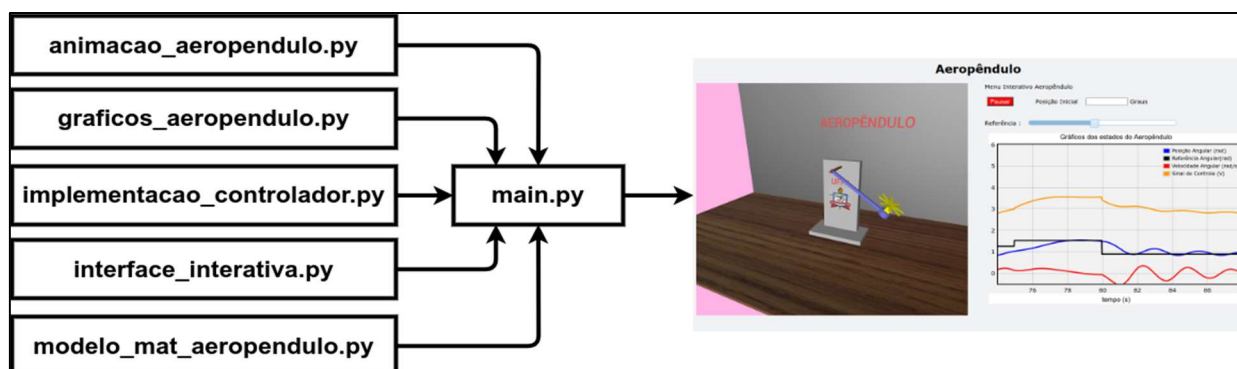


Fonte: Os Autores

4.2 Módulo 2 - Aero Pêndulo

Da mesma forma que o MAGLEV, o programa para simular o Aero Pêndulo foi modularizado para tornar o código mais organizado e legível. A Figura 7 apresenta a estrutura geral do projeto. Diferentes classes foram criadas em arquivos .py distintos e importadas para um programa principal que instancia os objetos de forma adequada.

Figura 7 – Estrutura dos arquivos Python.



Fonte: Os Autores

O módulo “animacao_aeropendulo” usa a biblioteca VPython para construir a estrutura 3D do simulador. A geração dos gráficos para ilustrar os sinais dos estados do sistema é realizada pelo módulo “graficos_aeropendulo”. Os controladores e o sinal de controle são criados a partir do módulo “implementacao_controlador”. O desenho da interface gráfica que permite a interação do usuário com o simulador advém do módulo “interface_interativa”. O módulo “modelo_mat_aeropendulo” implementa as equações de estados e retorna à solução, que é usada para guiar as animações e plotar os gráficos dos

sinais. Finalmente, o módulo “*main*” importa todos os outros módulos e realiza a interface entre as partes numérica e gráfica do simulador.

A tela do simulador foi mostrada na Figura 4, onde é possível visualizar a animação 3D e os gráficos da simulação. Todos os sinais mostrados estão em malha fechada com o controlador PI projetado conforme a Seção 2.2. Observa-se que o sistema se comporta de maneira coerente com a teoria, com erro de regime nulo. Os controles interativos do simulador permitem modificar a posição inicial do pêndulo além da amplitude de uma referência degrau.

5 CONCLUSÃO

O desenvolvimento das simulações se mostrou um projeto bastante recompensador aos discentes envolvidos. Ao engajar os alunos em desenvolver duas simulações animadas de controle em malha fechada, o MAGLEV e o Aero Pêndulo, foi possível iniciar uma metodologia que tende a tornar o ensino do assunto menos tedioso e mais visualmente atrativo. O trabalho motivou os discentes a aperfeiçoar o conhecimento sobre sistemas de controle, além de adquirir novas habilidades de programação, trabalho em equipe e desenvolvimento de software com controle de versão. Isso demonstra que o laboratório virtual é uma forma promissora de reduzir os obstáculos no ensino de sistemas de controle em cursos de Engenharia.

Apesar desta primeira iteração ter sido bem sucedida, há muitos desafios a serem vencidos. Um deles é a distribuição do pacote de programas. Para distribuir os códigos, é possível utilizar o sistema de distribuição PyPI, uma plataforma que permite compartilhar e instalar pacotes de software escritos em Python (PYPI, 2023). Os pacotes podem ser instalados usando o gerenciador de pacotes padrão da linguagem, tornando a instalação e execução das simulações mais fácil e rápida. Apesar disso, o formato requer a instalação local de um interpretador Python e de todos os pacotes necessários, incluindo o VPython. Isto pode ser inconveniente em alguns cenários, uma vez que há uma forte tendência atualmente em utilizar plataformas de *scripting* que rodam no próprio navegador da Internet, por exemplo, os largamente utilizados *Jupyter notebooks*. Infelizmente, estes se mostram inviáveis, de utilizar com o VPython, ao menos em um primeiro momento sem maior intervenção pelo programador, devido às dificuldades de chamar as rotinas gráficas remotamente.

Neste contexto, é curioso observar que o próprio distribuidor do VPython oferece um interpretador online, que permite criar animações 3D em qualquer navegador, independente do sistema operacional. Porém, o sistema não se mostrou oportuno a este projeto pois ele não permite a importação de pacotes adicionais importantes, tais como o módulo Python-Control. Vale frisar que os discentes do projeto procuraram encontrar soluções para ambos os problemas, mas os esforços se mostraram fora do escopo da proposta, haja vista que as ações adentram em aspectos mais profundos de linguagens de programação. Neste contexto, este trabalho pode ser uma oportunidade futura interessante de projeto multidisciplinar, integrando as áreas de Engenharia de Sistemas de Controle e Engenharia/Desenvolvimento de Software.

Finalmente, vale frisar que esta experiência serve de estímulo para atrair outros discentes e fazer o laboratório crescer, principalmente em variedade. Deste ponto em diante, esforços serão enviados para engajar outros discentes a apresentar suas próprias versões (desejavelmente melhoradas) dos sistemas trabalhados, ou de outros sistemas de controle clássicos, tais como o pêndulo invertido, braços robóticos, etc., tornando o

laboratório virtual uma ferramenta cada vez mais atrativa e motivadora aos estudantes de Sistemas de Controle. Os módulos desenvolvidos estão disponíveis em um repositório denominado LabVirtual na plataforma GitHub (TEIXEIRA, 2023).

REFERÊNCIAS

AHMAD, Aftab; RAFIUDDIN, Nidal; KHAN, Yusuf Uzzaman. Comparative Analysis of ANN and PID Controller of Aero-pendulum on Simscape. In: **2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)**. IEEE, 2021. p. 334-338.

COSTA, Jefferson S.; SILVA, Cleison D.; TEIXEIRA, Raphael B. Modelagem de sistema de levitação magnética por abordagem analítica baseada no conceito de circuitos magnéticos. In: XL Congresso Brasileiro de Educação em Engenharia, 2012, Belém. **Anais**. Belém.

FRANKLIN, Gene F.; POWELL, J. David; EMAMI-NAEINI, Abbas. **Sistemas de controle para engenharia**. 6. ed. Porto Alegre: Bookman Editora, 2013.

GLOWSCRIPT. **Documentação do VPython**, 2023. Página de documentação. Disponível em: <https://www.glowscript.org/docs/VPythonDocs/index.html>. Acesso em: 05 mai. 2023.

MENEZES, Nilo Ney Coutinho. **Introdução a programação com Python**. 1. ed. São Paulo: Novatec, 2010.

MILHOMEM, Rômulo L. **Construção de uma Bancada Didática de Processo de Levitação Magnética-MAGLEV**. 2010. 110 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Faculdade de Engenharia Elétrica, Universidade Federal do Pará, Tucuruí.

MOHAMMADBAGHERI, Amin; YAGHOUBI, Mahdi. A new approach to control a driven pendulum with PID method. In: **2011 UKSim 13th International Conference on Computer Modelling and Simulation**. IEEE, 2011. p. 207-211.

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

NUMPY. **NumPy Documentation**, 2023. Página de documentação. Disponível em: <https://numpy.org/doc/stable/>. Acesso em: 05 mai. 2023.

OGATA, Katsuhiko et al. **Modern control engineering**. 5. ed. New Jersey: Prentice-Hall, 2010.

PYPI. **LabVirtual**, 2023. Página de documentação do projeto Disponível em: <https://pypi.org/project/labvirtual/>. Acesso em: 12 mai. 2023.

SCIPY. **SciPy Documentation**, 2023. Página de documentação. Disponível em: <https://docs.scipy.org/doc/scipy/reference/>. Acesso em: 05 mai. 2023.

SYMPY. **SymPy Documentation**, 2023. Página de documentação. Disponível em:
<https://www.sympy.org/en/index.html>. Acesso em: 06 mai. 2023.

SILVA, Y. L. M. **Projeto, construção e controle de um aeropêndulo**. 2018. 43 f.
Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Centro de
Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Campina
Grande. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/xmlui/handle/riufcg/18954>.
Acesso em: 01 jun. 2023.

TEIXEIRA, Raphael. **LabVirtual**. Disponível em:
<https://github.com/raphateixeira/LabVirtual>. Acesso em: 01 mai. 2023.

EXPLORING THE POTENTIAL OF VIRTUAL LABORATORY FOR SYSTEM CONTROL USING VPYTHON IN ENGINEERING EDUCATION: ANALYSIS AND FUTURE PERSPECTIVES

Abstract: *This article describes the development of system control simulations using the VPython library and the PyPi platform. The simulation results demonstrate the effectiveness of using these tools in the engineering education process, providing a practical and interactive experience for students. The MAGLEV and Aero Pendulum simulations were developed and their results were presented and discussed in detail. Based on the results obtained, it was concluded that simulations are effective for teaching system control, as they allow students to visualize and understand theoretical concepts learned in the classroom in a practical way. In the future, it is expected to expand the use of these tools to other areas of knowledge, as well as to implement new simulations to further enrich the teaching and learning process.*

Keywords: *system control, simulation, VPython, PyPi, engineering education.*