



Universidade Federal de Uberlândia FEELT
Faculdade de Engenharia Elétrica
Engenharia de Controle e Automação



Sistemas Embarcados II

Relatório Descritivo

Trabalho Final 1 - Aeropêndulo

Ana Carolina Coelho Robl
André Felipe da Cunha Garcia

12011EAU021
11611EAU006

25 de julho de 2022

Introdução

A modelagem dinâmica, tem o intuito de identificar e modelar os aspectos de um sistema no qual podem ocorrer eventos que alterem sua trajetória ou modo funcionamento. Unindo os conceitos das disciplinas de controle, tem-se a possibilidade de modelar um sistema no qual um suposto pêndulo, acoplado de uma hélice, estabiliza-se em um ponto definido através da força gerada pela rotação da mesma.

Através da linguagem de programação python, para modelagem e as bibliotecas *pygame* e *math* para representação gráfica e simulação dos fenômenos físicos, foi possível construir uma aplicação que mostra o movimento do aeropêndulo tão como sua estabilização ao até atingir o ponto definido.

Módulos/Bibliotecas

A biblioteca *Pygame* é comumente utilizada na criação de aplicações multimídia, assim como jogos. Ela é uma camada de abstração construída sobre a biblioteca SDL (Simple DirectMedia Layer), e assim como a SDL ela tem alta portabilidade e funciona nos mais diversos sistemas operacionais e plataformas.

Math, é uma biblioteca simplificada que normalmente é utilizada para trabalhar com algumas funções matemáticas, valores escalares e outros tipos de trabalhos computacionais, tal qual foi utilizada a fim de realizar operações trigonométricas, entre outras.

No código referente a modelagem do aeropêndulo, foi utilizada a biblioteca *Numpy*, visto que ela tem uma melhor abordagem quando se trata de trabalhar com matrizes, vetores etc.

Desenvolvimento

A princípio, iniciamos a modelagem visual do aeropêndulo através da biblioteca *pygame*, a fim de obter a representação de um aeropêndulo propriamente dito e em seguida começar a construir a simulação que representasse o movimento realizado por ele. Em sequência, foram implementadas as funções, parâmetros e variáveis a serem utilizadas na aplicação.

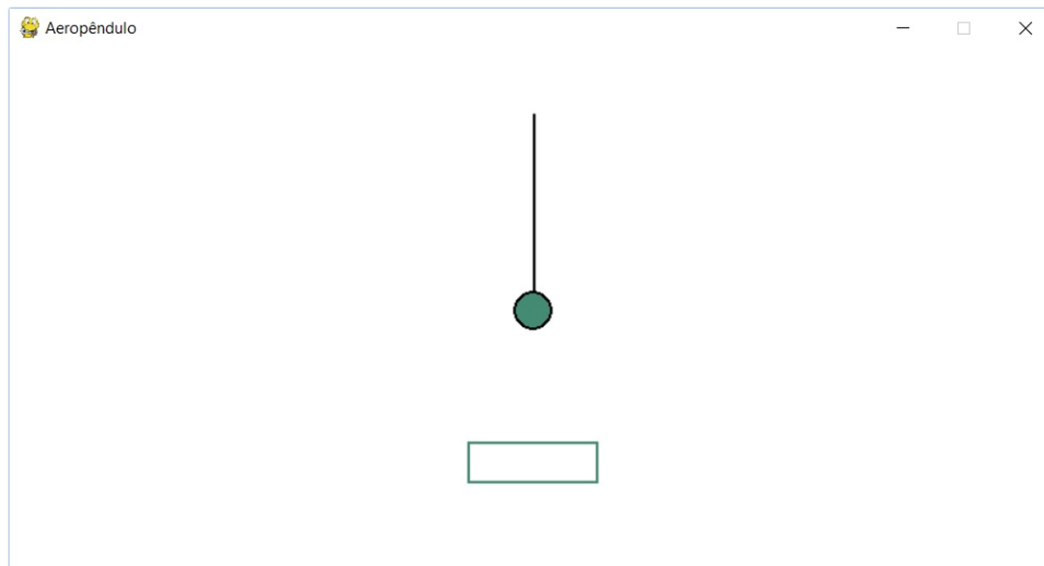


Figura 1 - Primeira Versão do Aeropêndulo

A fim de obter uma aplicação melhorada, tanto na UI quanto na UX, foi feita uma melhor definição do layout, aprimoramento das tratativas em relação ao input de dados, condições para melhorar a usabilidade do programa (apagar caracteres, aceitar valores negativos e decimais, receber apenas valores numéricos etc.) e, por último, mas não menos importante, a parte onde deixamos o aeropêndulo com um visual mais aproximado do real e todo o esquema de rotação da imagem ao longo de sua trajetória somado ao efeito animado da hélice em movimento.

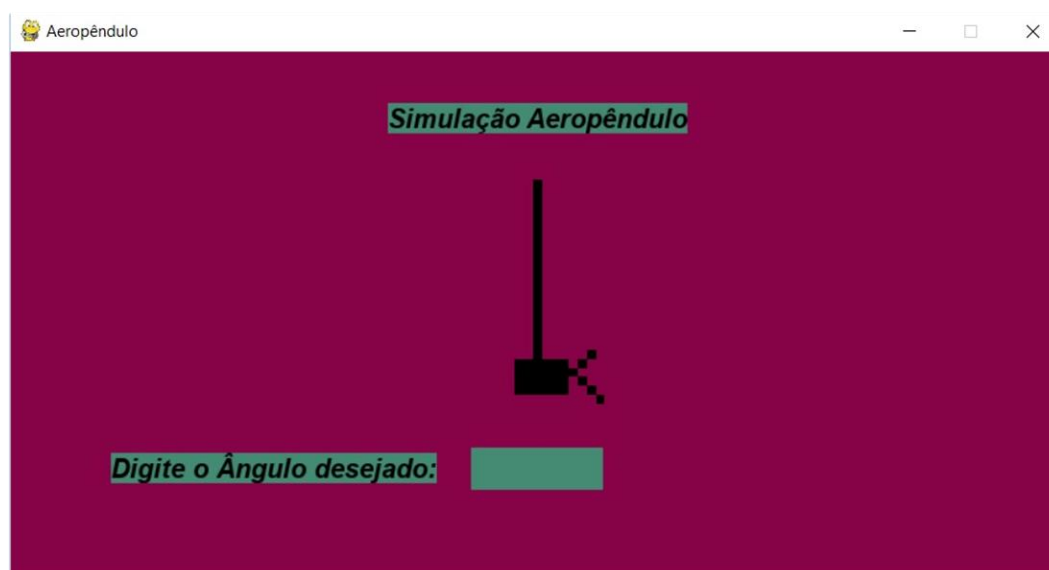


Figura 2 - Versão Final do Aeropêndulo

Funcionamento

A integração do sistema de modelagem do aeropêndulo utilizando o método Runge Kutta de quarta ordem e do código utilizando a biblioteca pygame, se deu através do envio do valor imputado pelo usuário na caixa de texto disponível na tela para o sistema de modelagem que, ao receber o ângulo, realiza todos os processos para que o aeropêndulo atinja a posição informada.

Em outras palavras, o sistema inicia-se no ângulo previamente definido de valor zero, em seguida, o usuário digita o ângulo que deseja que o aeropêndulo atinja. Essa posição angular inserida, juntamente com a posição anterior, são enviadas ao sistema de modelagem onde todos os cálculos são feitos e ocorre o retorno da matriz de valores para o código principal. Esses valores correspondem ao trajeto do aeropêndulo num período definido. Desta forma, conseguimos exibir visualmente a trajetória do aeropêndulo.

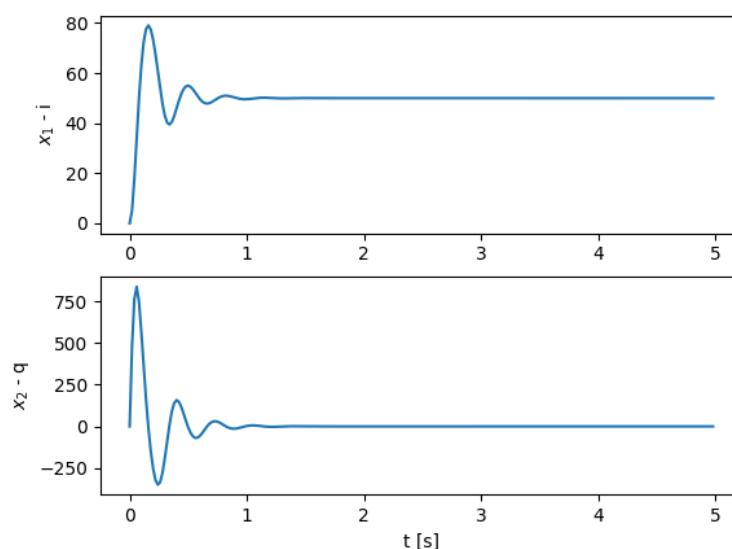


Figura 3 - Resposta do Sistema ao Ângulo de 50°

Através da *Figura 3*, é possível ver claramente o comportamento do aeropêndulo ao longo do tempo ao realizar a trajetória de ir de 0° até 50°.

Aplicação

Através da biblioteca pygame, foi possível ir além de uma simulação simples como a da *Figura 3*, para uma representação visual da trajetória do aeropêndulo mais próxima do que realmente enxergaríamos se estivessemos olhando para um aeropêndulo físico.

A princípio, como foi citado acima, o vetor de ângulos recebido é utilizado para realizar uma atualização constante da posição do aeropêndulo, ou seja, esse vetor é varrido e as posições são atualizadas na janela gerada pela aplicação. O código que modela o sistema, possui um intervalo de amostragem de 1/50 durante o período de 5 segundos já a aplicação,

foi definido como 50Hz sua taxa de atualização, sendo este valor, suficiente para representar o movimento com certo grau de fluidez, sendo possível até observar o overshoot que ocorre durante o processo de estabilização do aeropêndulo.

Para observar o processo citado acima, basta digitar o valor do ângulo desejado na caixa de texto indicada na *Figura 2* e pressionar a tecla “ENTER” ou “RETURN”. São permitidos valores angulares decimais, tanto negativos quanto positivos.

Conceitos Extras

Piskel – Editor online gratuito para animação de sprites e arte “pixelada” no geral. Foi utilizado a fim de obter uma melhor representação do aeropêndulo e uma animação dele.

Apêndice

Códigos Utilizados:

Main.py

pend_sim.py