Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica
Sistemas Embarcados II

Sistemas Embarcados II

Trabalho Prático 1

Aluno: Rafael Dias Pereira **nº de matrícula:** 11911EAU003
Professor: Éder Alves de Moura

21 de julho de 2022

Conteúdo

Objetivos

Este trabalho visa realizar o controle de um aeropêndulo e simular computacionalmente o seu comportamento com base na entrada sugerida pelo usuário. Para isso, deve-se também obter e aprimorar conhecimentos acerca de bibliotecas que realizam uma simulação gráfica do projeto, como a pygame, obedecendo os princípios físicos e a função de transferência do sistema.

Desenvolvimento

Controle

Para que o sistema funcione, é necessário que o pêndulo atinja o ângulo de entrada definido pelo usuário. Desta forma, a sua saída deve ser modelada de forma que a entrada passe pela função de transferência do sistema, em conjunto com seu controlador, para que a saída seja o valor desejado.

Tendo em vista que o sistema foi modelado previamente através método de Runge-Kutta de 4^a ordem em Python, o restante do processo de controle foi a integração dos cálculos do sistema com a aplicação em que o usuário verá a simulação.

Tal integração funcionou da seguinte forma: o usuário entra com um ângulo (em graus) e esse dado é armazenado na variável *input angle*. Essa variável, juntamente com uma outra que diz a posição angular atual do sistema (*initial pos*), são enviadas para o sistema de controle para que os cálculos sejam feitos.

Após a realização dos cálculos, os resultados que foram armazenados em uma matriz são retornados para a aplicação. Esses resultados são a própria resposta do sistema, em ângulos, em um período de simulação, como exemplificado a seguir (Fig. 1).

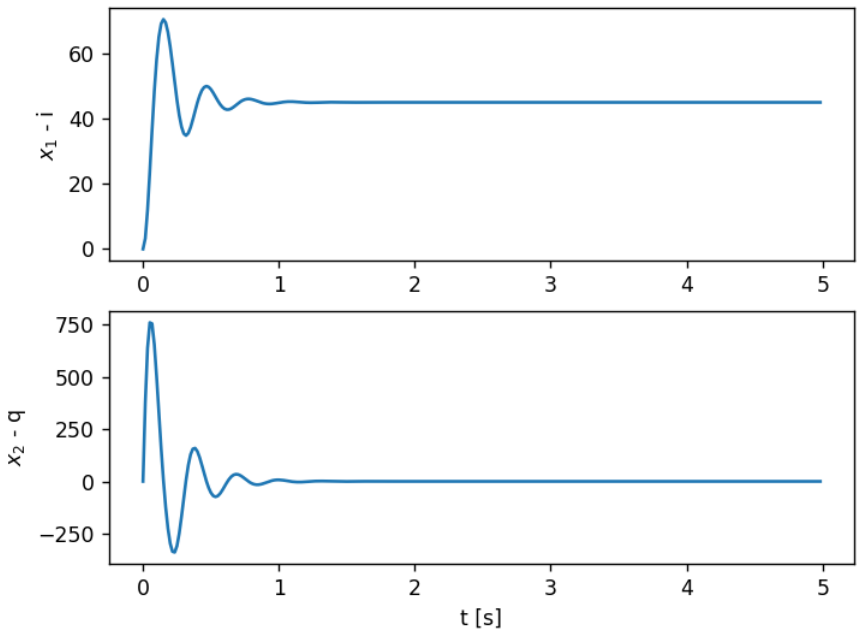

Figura 1 – Simulação do sistema.

Como podemos observar, para uma entrada de 45 graus, temos a resposta angular representada pelo gráfico acima. O primeiro gráfico representada a posição angular, enquanto que o segundo representa a sua velocidade. O que nos interessa é a posição, então retornaremos apenas o vetor dos ângulos para a aplicação.

Aplicação

A aplicação é a área responsável pela simulação do sistema. Isso foi possível utilizando a biblioteca pygame, do Python. A lógica da simulação se baseou em atualizar continuamente o ângulo do pêndulo conforme percorria-se o vetor de posições do sistema, obtido pelo controlador. A Imagem 2 a seguir mostra a interface geral e a apresentação do pêndulo em questão.

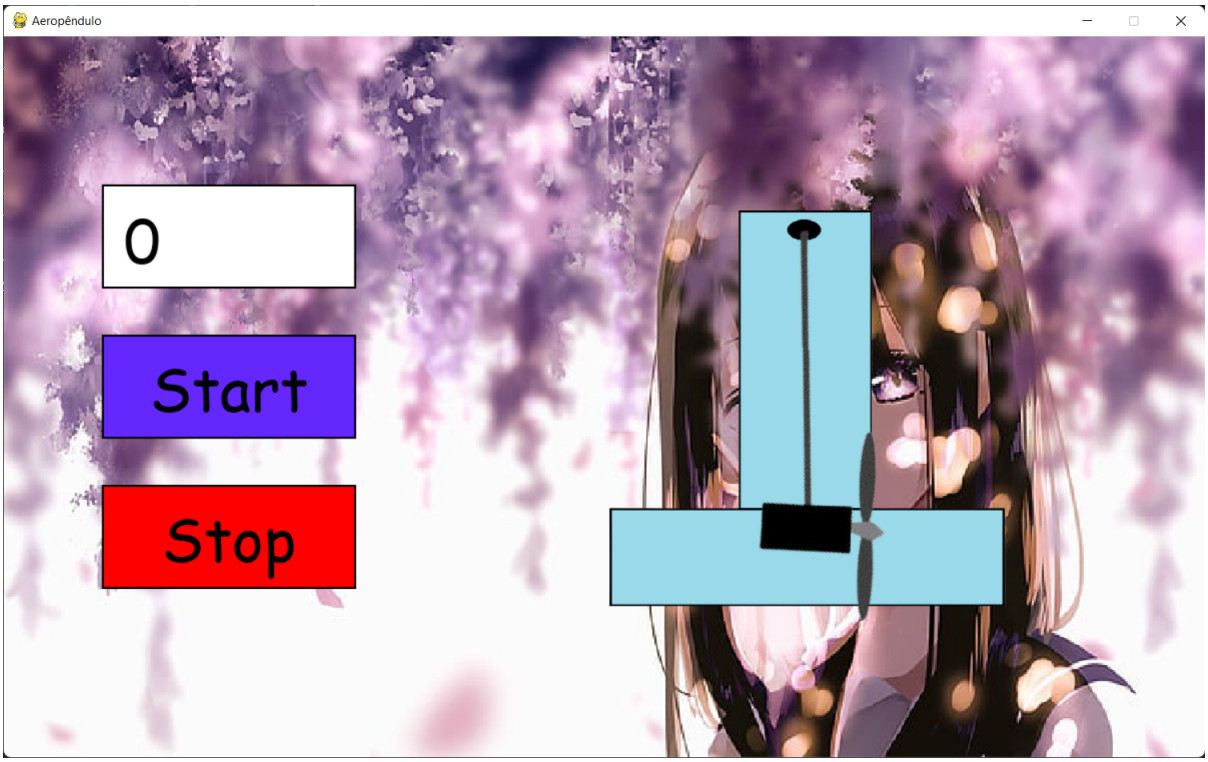

Figura 2 – Aplicação.

Podemos analisar que o lado esquerdo é a região de entrada, onde o usuário irá definir qual ângulo (em graus) deseja-se que o pêndulo atinja. Dessa forma, ao digitar um valor numérico e clicar no botão *Start*, o pêndulo irá alterar seu ângulo de forma a respeitar a sua função de transferência.

Além disso, uma importante medida foi tomada para sincronizar a taxa de atualização do ângulo do pêndulo com o tempo real. Foi definido um clock de 60 Hz para a aplicação. Ou seja, ela irá ler 60 posições por segundo. Para que essa leitura por segundo fosse igual ao tempo real, o período de amostragem no cálculo do sistema foi definido como $\frac{1}{60}$. Então, com a aplicação

rodando 60 posições por segundo, e o vetor de posições tendo 60 valores entre um segundo e outro, 1s na simulação equivale a 1s na vida real.