



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**  
**FACULDADE DE ENGENHARIA MECÂNICA**  
**Curso de Graduação em Engenharia Mecatrônica**



**RELATÓRIO DO TRABALHO 1 DA DISCIPLINA**  
**SISTEMAS DIGITAIS PARA MECATRÔNICA (FEELT49081)**

**Simulação Drone 2D**

Prof. Éder Alves de Moura

Hericles Felipe Ferraz - 11811EMT022

Vinicius Marques Machado Rodrigues - 11621EMT023

José Divino Ferreira Júnior - 11621EMT010

Gabriel Augusto de Moraes Batista - 11421EMT007

Breno Batista Moura - 11711EMT010

**Uberlândia, março de 2022**

## 1. INTRODUÇÃO

Na engenharia, a simulação é um processo fundamental para avaliar um comportamento de um sistema, e com o rápido aprimoramento dos computadores o processo de simulação teve uma grande evolução, juntamente do avanço tecnológico na fabricação e uso de drones. Dessa forma, o objetivo deste trabalho é realizar a simulação de um sistema de controle de posição aplicado em um drone 2D, utilizando a modelagem cinemática e dinâmica do mesmo.

## 2. MÓDULOS EXTERNOS

Para a execução da simulação ser possível, é importante que se tenha o Python na versão 3.8 ou superior instalado e também os módulos:

- Numpy (<https://numpy.org/install/>)

É um módulo que fornece um grande conjunto de funções e operações de biblioteca que ajudam os programadores e engenheiros a executar facilmente cálculos numéricos, relacionados a operações em arrays multidimensionais,.

- Pygame (<https://www.pygame.org/install/>)

Pygame é uma biblioteca de jogos multiplataforma feita para ser utilizada em conjunta com a linguagem de programação Python baseada em SDL. É voltada para o desenvolvimento de games e interfaces gráficas. Fornece acesso a áudios, teclados, controles, mouses e hardwares gráficos via OpenGL e Direct3D.

## 3. IMPLEMENTAÇÃO

A simulação será realizada utilizando a linguagem Python para implementação da dinâmica do drone e do sistema de controle, e para representação visual será usada a biblioteca Pygame.



Figura 1: Simulação no Pygame

### 3.1 Inicialização

O código *Move* é o principal. É onde é aberta a tela de simulação e onde são definidas as suas propriedades. Em relação ao movimento do drone, é onde foi declarado um vetor de waypoints que servem de referência para o sistema de controle, e também é onde é feita a leitura da entrada do teclado para direcionar o drone.

### 3.2 Sistema de controle

O código *Simulation* contém uma classe que é responsável pela implementação do sistema de controle de posição, onde foi implementado um controlador proporcional derivativo (PD) para gerar a ação de controle, onde a referência é definida pelas setas direcionais do teclado do usuário. Além disso, foi utilizado o método de Runge-Kutta para resolução de EDO's, conhecido como uma otimização do método de Euler. Esse método consiste em comparar um polinômio de Taylor apropriado para eliminar o cálculo das derivadas, fazendo-se várias avaliações da função a cada passo.

### 3.3 Interface de gerência de aplicação

O código *menu* é responsável pela parte onde se realiza a gerência da aplicação, por meio das funções *bg\_menu* e *drone\_menu*. A função *bg\_menu* é utilizada para a escolha de um plano de fundo entre as cinco opções disponíveis, que será utilizado na aplicação, trabalhando de maneira a carregar o plano de fundo selecionado pelo usuário por meio do clique do mouse e a função *drone\_menu*, que é a responsável pela seleção do modelo do drone que será utilizado para realização da simulação, funciona de maneira semelhante, permitindo ao usuário selecionar um entre os dois drones disponíveis para a execução da aplicação.

## 4. BIBLIOGRAFIA

[1] PYGAME ORG. Pygame.org, 2022. Disponível em: <<http://pygame.org/>>.

Acesso em: 07 de fevereiro de 2022.

[2] PyGame: A Primer on Game Programming in Python.

<<https://realpython.com/pygame-a-primer/>>.

Acesso em: 14 de fevereiro de 2022.

[3] Ogata, K. - Engenharia de Controle Moderno, Prentice-Hall, 4ª. ed., 2004. Kuo, B.C.

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