



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE ENGENHARIA ELÉTRICA

## **Sistemas Embarcados II - Trabalho 1**

Miguel Ravagnani de Carvalho - 12011EAU019

Uberlândia  
5 de março de 2022

# Sumário

1	Introdução	2
2	Simulação	3
3	Controle	4
4	Desenvolvimento	5
5	Conclusão	6

# 1 Introdução

O objetivo deste trabalho é implementar uma simulação 2D do que seria um controle de posição de um drone. Para alcançar esse objetivo, foi desenvolvido um jogo.

## 2 Simulação

Para realizar a simulação com fidelidade ao comportamento esperado de um drone, o primeiro passo foi a implementação de um modelo cinemático simples. Abaixo, as equações que descrevem a dinâmica do drone:

$$\dot{\bar{W}} = \frac{1}{\tau}(-W + \bar{W}) \quad (1)$$

$$\dot{\bar{r}} = v \quad (2)$$

$$\dot{\bar{v}} = \frac{1}{m}(D^{\frac{R}{B}}(\phi)F_C + P) \quad (3)$$

$$\dot{\bar{\phi}} = \omega \quad (4)$$

$$\dot{\bar{\omega}} = \frac{1}{I_Z}T_C \quad (5)$$

Para a implementação de um modelo de espaço de estados, um vetor de estados é definido:

$$x = [W^T \quad r^T \quad v^T \quad \phi \quad \omega]^T \quad (6)$$

De tal forma que:

$$\bar{W} = [W_1 \quad W_2]^T \quad (7)$$

$$\bar{r} = [x_r \quad y_r]^T \quad (8)$$

$$\bar{v} = [v_x \quad v_y]^T \quad (9)$$

$$F_C = [0 \quad F_1 z \quad + F_2]^T \quad (10)$$

Nas relações descritas acima,  $W$  é a velocidade de rotação dos rotores,  $r$  é a posição do drone,  $v$  é a velocidade linear do drone,  $\phi$  é a atitude,  $\omega$  é a velocidade angular,  $F_C$  é a força de controle,  $T_C$  é o torque de controle, e  $D^{\frac{R}{B}}$  é a matriz de rotação.

O modelo foi implementado com o auxílio do método de Runge Kutta, de 4 ordem.

### 3 Controle

Para simular o controle, parâmetros de simulação foram definidos:

- $m = 0,25$  [kg] - Massa do drone
- $I_Z = 2 * 10^{-4}$  [kg\*m<sup>2</sup>] - Momento de inércia de rotação
- $g = 9,81$  [m/s<sup>2</sup>] - Constante de aceleração da gravidade
- $l = 0,1$  [m] - Distância entre o centro de massa e ponto de atuação da força dos rotores
- $W_{max} = 15000$  [rpm] - Velocidade máxima de rotação dos rotores
- $k_f = 1,744 * 10^{-8}$  - Constante de força dos rotores
- $\tau = 0,005$  - Constante de tempo

Então, o seguinte modelo de controle foi implementado:

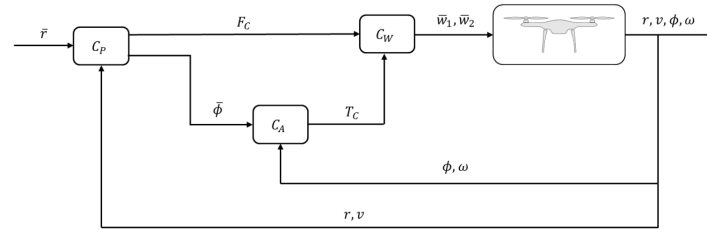


Figura 1: Sistema de controle

## 4 Desenvolvimento

A simulação foi implementada em um software escrito em C++, com padrões de programação orientada à objeto e *design patterns*. Para simulação gráfica, a API OpenGL foi utilizada, e através desta, alguns modos de renderização de *sprites* e tratamento de *input* do teclado. Todas as *sprites* (drone, botoões do menu e fundo) foram desenhadas manualmente no photoshop.

O loop do jogo consiste em uma tela de menu, uma tela para controle manual, e uma tela com a atuação do controle.

Todo o desenvolvimento foi registrado em um repositório do Github ([https://github.com/MiguelRavagnani/OpenGL\\_Drone\\_Navigation](https://github.com/MiguelRavagnani/OpenGL_Drone_Navigation)).



Figura 2: Tela de menu

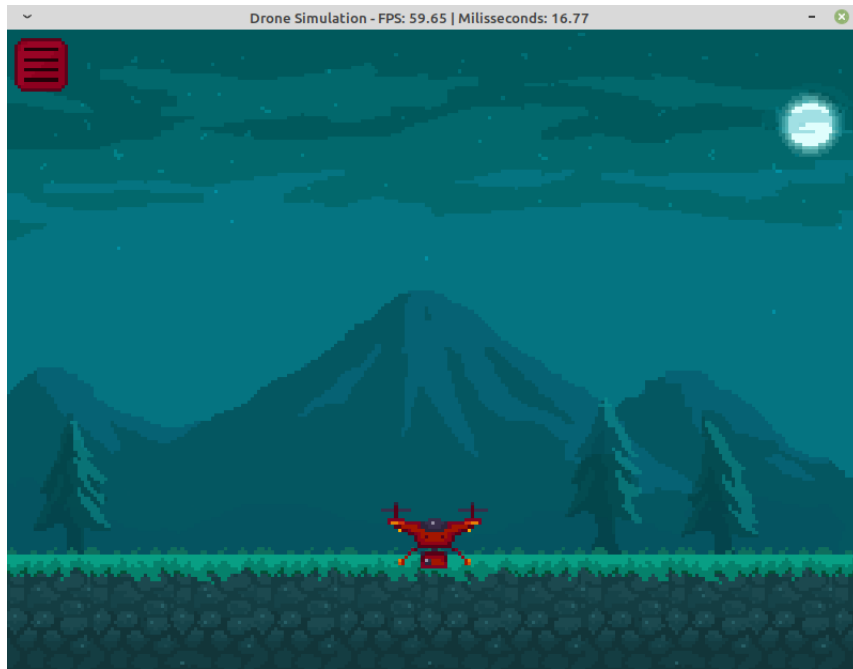


Figura 3: Tela de controle manual

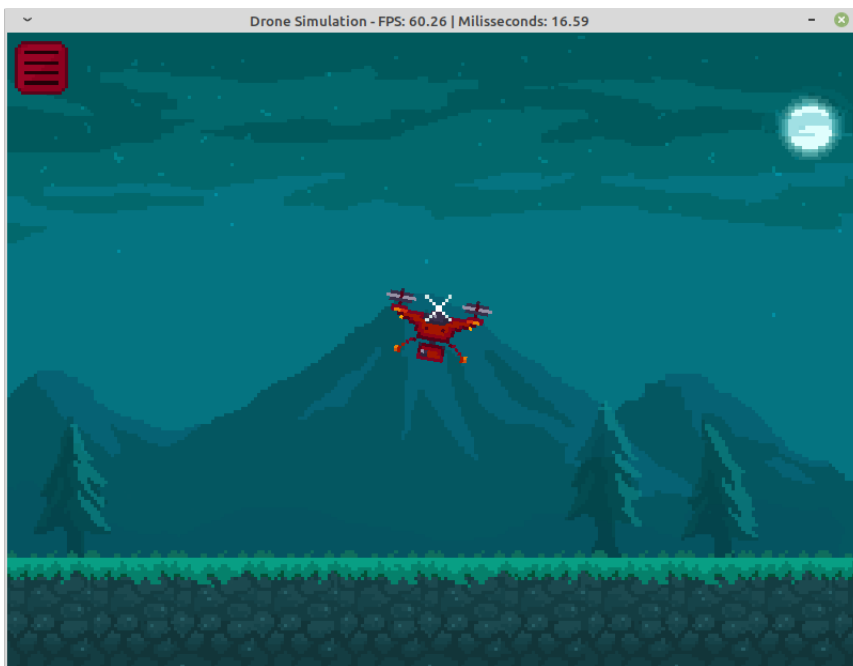


Figura 4: Tela de controle automático

## 5 Conclusão

Este trabalho permitiu que conceitos de modelagem de sistemas e controle fossem revisitados. O desenvolvimento proporcionou diversos desafios de implementação de design de software.