ROTARY-WING UAVs FOR MICROGRAVITY EXPERIMENTS

USER MANUAL FOR GITHUB REPOSITORY

Table of Contents

1. Repository Description

All the files available in the GitHub Repository were created using version *2021.3.10f1* on Unity. To work with the files from this project they must be downloaded from the repository and imported into a Unity project by selecting "Import New Asset" after right-clicking in the Project Tab.

Materials – This folder contains all the materials and physics materials used for the project.

Prefabs – This folder contains the Quadcopter prefab with the scripts, rigidbody elements, mesh colliders, etc. attached to it and can be used directly after importing without needing to make any edits. The quadcopter CAD model has been made available as a .obj file in the repository.

Scenery – This folder contains all the scenery objects added in the background and is a free asset downloaded from the Unity Asset Store. All the documentation on how to use this is available within this folder itself.

Scenes – This folder contains the main scene for this project.

Scripts – This folder contains the scripts used in this project.

MATLAB scripts – This folder contains three main scripts used for post-processing of the data saved as a .csv file.

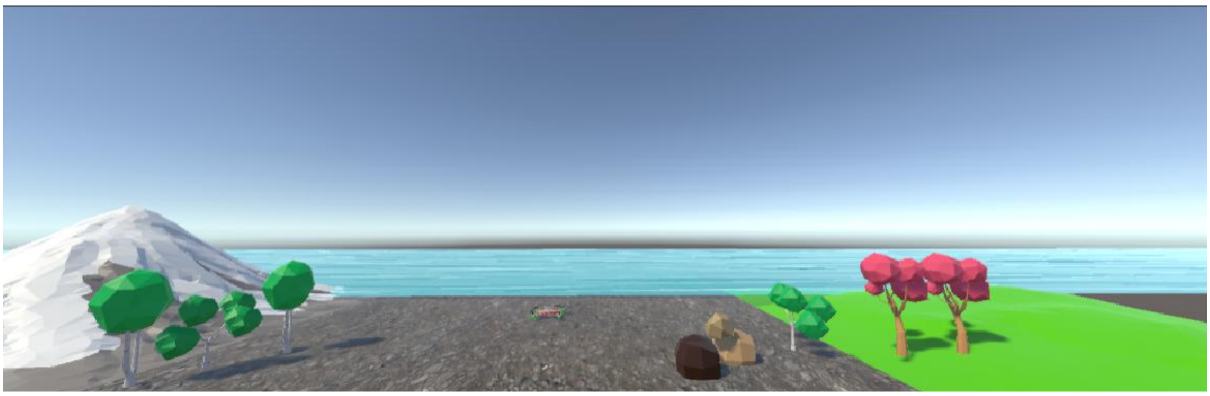

Figure 1: Sample arrangement of scenery elements with UAV resting on the ground

2. Scripts

2.1 Unity C# Scripts

Script Name	Function
AutomateMovement5	User inputs are controlled with this script, and the PID controller and Drag Implementation is also done with this script. Any changes to drag parameters, PID gains, etc. are controlled through this script.
DragTesting	This script contains the drag implementation by itself and was used for testing. It is not attached to the Quadcopter Prefab but can be added in a separate scene for drag tests.
Gusts	The user inputs and implementation of discrete gusts is controlled with this script.
PlotMovement	This script records and saves position and velocity of the UAV at each time step into a .csv file with specified file path.
PlotPayloadMovement	This script performs the same function as PlotMovement script for the payload on-board the UAV. This file should be attached to any payload objects.
Testing	This script applies a simple downward force to the UAV rigidbody and is used for one of the verification tasks. This script can be attached to the quadcopter in any test scenes, but is otherwise of no use at the final stage of the project.

Table 1: Description of the functions of various scripts in the "Scripts" folder of the Repository

2.2 Stored data as a csv file

Time	PositionX	PositionY	PositionZ	VelocityX	VelocityY	VelocityZ
0	1.19E-06	-2.18E-06	3.95E-07	0	0	0
0.01	1.18E-06	-0.0005	5.18E-07	-4.87E-13	-0.05008	-2.02E-09
0.02	1.18E-06	-0.00097	5.18E-07	-4.87E-13	-0.04716	-2.02E-09
0.03	1.18E-06	-0.00186	5.18E-07	5.82E-10	-0.08825	-7.00E-11
0.04	1.18E-06	-0.00268	5.18E-07	-4.43E-10	-0.08253	1.13E-10
0.05	1.18E-06	-0.00384	5.18E-07	-8.12E-11	-0.11548	9.69E-12
0.06	1.18E-06	-0.00491	5.18E-07	1.23E-11	-0.10706	-1.21E-11
0.07	1.18E-06	-0.00623	5.18E-07	1.22E-11	-0.13262	-6.83E-12
0.08	1.18E-06	-0.00745	5.18E-07	7.72E-12	-0.12163	-5.99E-12

Table 2: Sample of data stored as a .csv file by the PlotMovement script

2.3 MATLAB Scripts for Post-processing

MATLAB is used for post-processing of the raw data displayed in table 2 and to obtain analytical values for comparison. Script 'PlotResultsMovAvg.mlx' is used to calculate and average the acceleration, g Levels, Power, Thrust and Total Energy Consumption data. This script requires the data in table 2 to be imported as column variables with the same names as shown above (column headers are set in 'PlotMovement.cs' script in Unity). 'GustsPlot.mlx' calculates the acceleration and g-Levels for the same for the Gust tests for all cases – gust in x direction, y direction, z direction and no gusts. 'PlottingAccVelDisp.mlx' provides the theoretical results for all user inputs to compare the simulation results with.

Defining Input Values

```
%Input values chosen by user
t1 = 0:0.01:3.0; %accelerated climb phase duration
t2 = 3.0:0.01:6; %microgravity phase duration
t3 = 6:0.01:10; %recovery phase duration
a1 = 4.9; %accelerated climb phase acceleration
a2 = -9.8; %microgravity phase acceleration
a3 = 1.1; %recovery phase acceleration
v_initial = 0; %initial velocity
h_final = 10; %final altitude
h_initial = 0; % initial altitude
```

Figure 2: User inputs needed to be defined in PlottingAccVelDisp.mlx script that can provide altitude, velocity in y-direction and acceleration in y-direction

2.4 Calculated and Averaged acceleration data from MATLAB

Acc	gLevels	Power	Thrust
-0.14773	0.984926	2.802662	19.32266
0.212576	1.021691	2.702662	20.04664
0.263291	1.026866	2.488038	20.14825
0.588579	1.060059	2.647438	20.7984
0.640713	1.065379	2.359389	20.90272
0.934513	1.095358	2.418877	21.48991
0.987286	1.100743	2.056258	21.59541
1.252758	1.127832	2.0174	22.12601
1.305533	1.133218	1.58025	22.23147
1.545505	1.157705	1.445538	22.71117
1.597767	1.163037	0.935175	22.81563
1.814773	1.185181	0.707801	23.24955
1.866107	1.190419	0.126459	23.35222

Table 3: Sample of averaged acceleration, g-Level, Power and Thrust data as stored in MATLAB variables