

**Graphical user interface for a quadrocopter**

**Abstract**

This project paper discusses the process of creating a GUI (Graphical User Interface) for a quadrotor Tello Edu that can be controlled remotely. The GUI for this quadrotor project was created using Matlab software and Support package for Ryze Tello Drone. The drone is equipped with a sensor that measures the quadrotor's orientation, speed, height and then is processed by a microprocessor to identify the quadrotor's present information. The collected data is then shown on the designed graphical user interface (GUI). This document also discusses the collected data displayed on GUI. The graphical interface has another features such as showing a 3D real-time orientation of the drone and camera streaming in real time, which will be discussed further.

**Introduction**

In the modern day quadrocopter plays important role in many fields of human life. It has been used ranging from surveillance, photography, agriculture to entertainment such as racing. The quadrocopter has an advantage compared to other type of drone, which is hovering ability. Furthermore, due to excellent maneuvering capability, quadrotors are able to do a mission with any kind of environment. During the flight it can automatically stabilizes the attitude of the vehicle and prevents the vehicle from flopping over in the absence of pilot and also maintains the desired horizontal orientation. Development and design of GUI for monitoring the UAV is mandatory because a system which is manually monitored by human is not accurate and sometimes not reliable at long distance and high altitude. A GUI system helps human to measure the orientation, position and monitor the system and environment.

**The movement of Quadrocopter and orientation of drone**

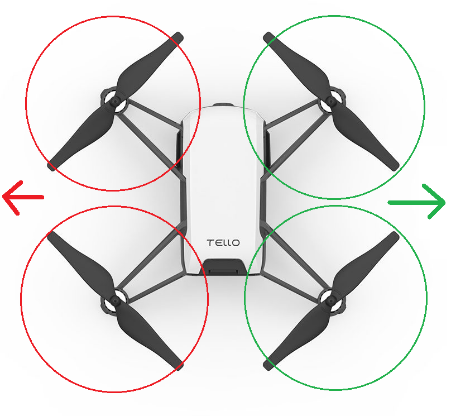
|  |  |  |  |
| --- | --- | --- | --- |
|  | Symbols |  | Meaning |
|  | x  y  z  θ  φ  ψ |  | Forward velocity (m/s)  Side velocity (m/s)  Vertical velocity (m/s)  Pitch angle (deg)  Roll angle (deg)  Yaw angle (deg) |

Table 1. List of symbols

**During the documentation the drone´s orientation is using Euler ZYX Convention**



Figure 1. Drone´s Orientataion

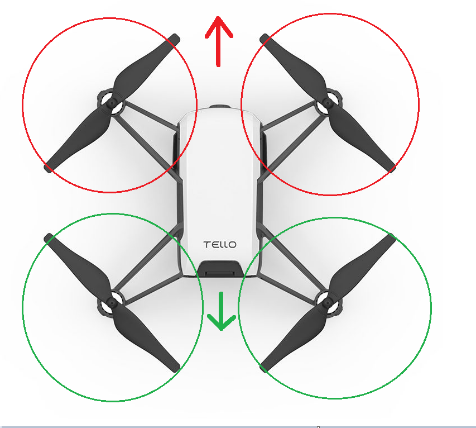
**Roll – The drone will move respect to the x-axis (red color)**

To Roll to the Left, the lift is increased on the motors on the Right. The drone must also decrease the lift on the motors on the Left. When drone moves to right side the roll angle (φ) will varies from 0 to 180 degree.

If you wish to roll right, you do the exact opposite. The lift is increased on the motors on the left. The drone must also decrease the lift on the motors on the right and the roll angle (φ) will varies from 0 to -180 degree.

Figure 2. Roll´s direction

**Pitch – The drone will move respect to the y-axis (green)**

 To move the drone to pitch forward you. The power that is applied to the rear motors is increased. This causes the Drone's nose to tilt downward due to a forward net push. To maintain angular momentum, you must also reduce the power delivered to the two front motors. During the moving forward the (θ) can be seen from 0 to -180 degree

And to move backward you need to increase power to front motors and reduce power to rear motors. The pitch angle moving backward can be seen from 0 to 180 degree

Figure 3. Pitch´s Direction

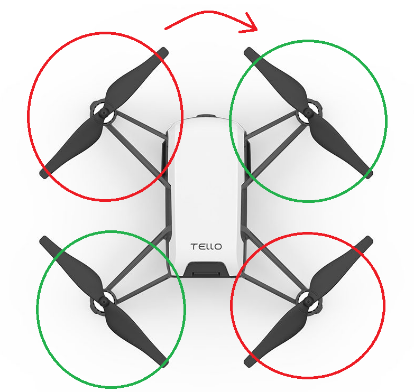
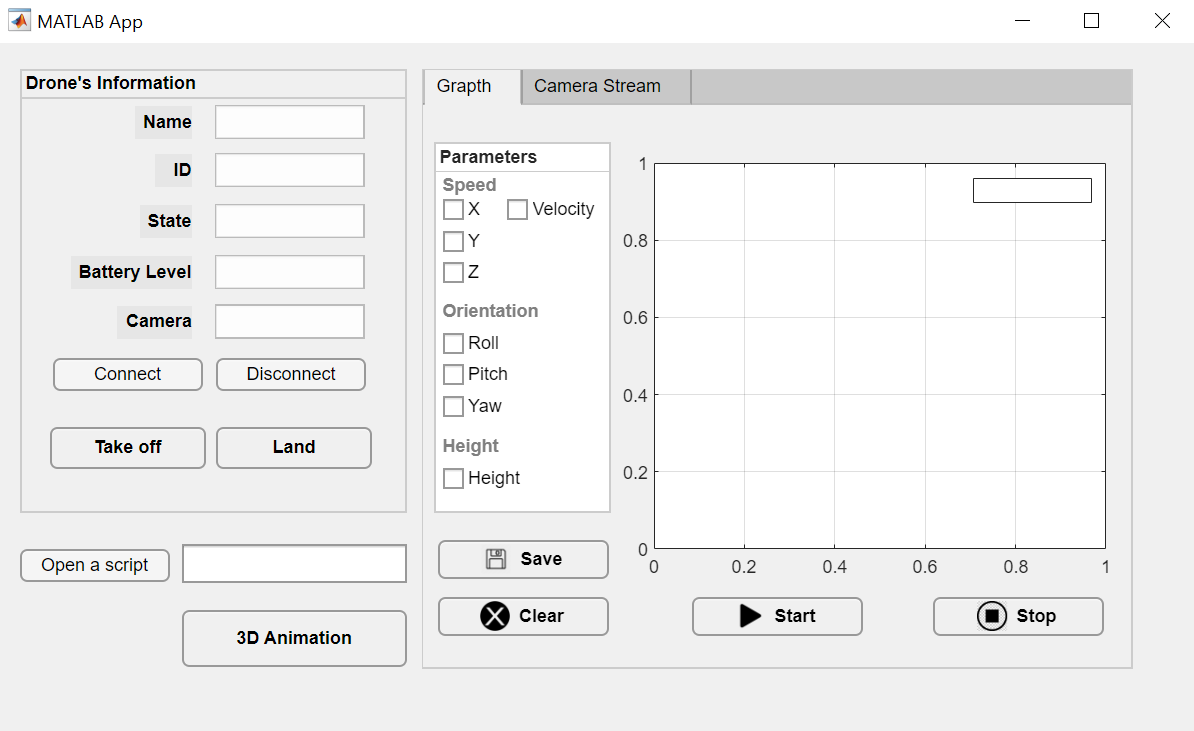
**Yaw - The drone will move respect to the z-axis (blue)**

Figure 4. Yaw´s Direction

To Yaw (rotate) the drone in a clockwise direction. The lift on the clockwise rotating motors must be increased. On anti-clockwise rotating motors, you must additionally reduce the lift. This is done in order to keep both the upward and downward net forces at zero. There is also an anti-clockwise torque as a result. To conserve Angular Momentum, the drone turns clockwise.

**Design of Graphical User Interface**

**Figure 6. Overview of Graphical User Interface**

Every functions used during my work was from Support Package for Tello Drone from Matlab.

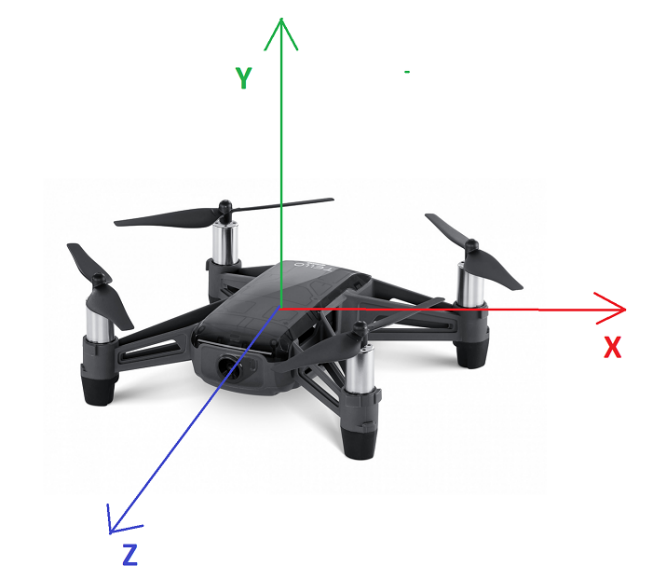
The following sections describe the different parts of the graphical user interface.

* The drone’s information panel: The panel shows the general state of the system and the main control commands. This panel provides a summary of information considered critical that needs to be permanently present on the screen.
* Parameters box: In order to measure the values of drone, the tick boxes with different values are introduced. The selected data will be displayed on the plot in real time.
* Camera Stream: In Camera tab is a option to see a view from FPV camera from drone during the flight.
* Open a script button: When user wants to use a Matlab script to control the drone and measure the data.
* 3D Animation: The GUI includes a viewer that shows in animated 3D views of drone’s body orientation.
* Start, Stop, Clear button: To interact with drone to plot data to the figure, stop plotting or clear plot.
* Save button: To save a flight data as png file

**Orientation Measurement based on body frame North, East and down**

By using *readOrientation(droneObj)* function we can determine body orientation. The function will return array of Euler angles represented as [Yaw, Pitch, Roll]. The data was in radian and was converted to degree.

The axis is along ZYX axes. This represents the rotation of the drone from the NED frame to the estimated body frame.

**Linear Movement**

For speed measurement the function *readSpeed(droneObj)* was used, which returns the current speed of the Ryze drone along the x-,y-, and z- axes with respect to the inertial NED frame. The NED frame is calculated at drone startup

**Height´s Measurement**

The quadcopter drone's rotors function similarly to wings. They generate lift by spinning quickly, pulling the air downwards and propelling the quadcopter drone into the air. If the lift cancels out with gravity's acting force, net force becomes zero, hence the quad hovers in mid-air. Then there is thrust, which makes the quadcopter to move in that direction. The altitude of drone is controlled by decreasing lift to change its height.

*ReadHeight(droneObj)* function measures the current height relative to the takeoff surface in meters along with the system time. The current height of the drone relative to the takeoff surface, specified in meters.

**References**

1. <https://www.mathworks.com/help/supportpkg/ryzeio/referencelist.html?type=function&listtype=cat&category=index&blocktype=all&capability=&s_tid=CRUX_lftnav>
2. Fathoni, M.F.; Lee, S.;Kim, Y.; Kim, K.-I.; Kim, K.H.Development of Multi-QuadrotorSimulator Based on Real-TimeHypervisor Systems. Drones 2021, 5,59. https://doi.org/ 10.3390/drones5030059
3. Natural user interfaces for human-drone multi-modal interaction, Journal: 2016 International Conference on Unmanned Aircraft Systems (ICUAS), : 2016, ISBN: 9781467393348