Python Code Explanation: Visualizing Drone IMU Data

This Python script visualizes and interacts with data from a drone's Inertial Measurement Unit (IMU) in real time. It has two main parts: data acquisition from the drone and dynamic visualization of rotor motion based on roll, pitch, and yaw angles.

Key Features of the Code:

1. Libraries and Setup:

- Uses Matplotlib and mpl_toolkits.mplot3d for 3D visualization.

- Employs the MAVSDK library to interact with the drone and retrieve telemetry data.

- Uses asynchronous programming (asyncio) for real-time data acquisition and processing.

2. 3D Visualization:

- A 3D figure is initialized using Matplotlib's interactive mode (plt.ion()).

- Two rotor types are visualized:

- Single Rotor: Rotors and their arms are represented by lines extending from the base stand.

- Twin Rotors: Two rotors are drawn as circles at specified positions, simulating a drone design with dual rotors.

3. Dynamic Rotation:

- Rotors are rotated in real-time using roll, pitch, and yaw angles.

- The script calculates the corresponding rotation matrices for each angle:

- R_roll, R_pitch, R_yaw are combined into a single matrix R.

- The rotation matrix transforms rotor positions to reflect the drone's orientation dynamically.

4. IMU Data Acquisition:

- Connects to a drone using the MAVSDK system, through a specified UDP address.

- Fetches real-time roll, pitch, and yaw data from the drone's IMU.

- Updates the 3D visualization with these values.

5. Event Loop:

- The main loop (get_imu_data) handles both IMU data acquisition and updating the plot continuously.

Functions Overview:

Visualization Functions:

1. draw_rotor(ax) / draw_twin_rotor(ax):

- Draws the initial setup of the stand and rotors for single and twin rotor configurations, respectively.

2. rotate_rotor(ax, roll, pitch, yaw) / rotate_twin_rotor(ax, roll, pitch, yaw):

- Applies rotation matrices to simulate the orientation of rotors based on IMU angles.

Drone Interaction:

1. get_imu_data():

- Establishes connection to the drone and retrieves roll, pitch, and yaw telemetry data in real-time.

Code Flow:

1. Start the script.

2. Establish a drone connection through MAVSDK.

3. Continuously fetch IMU data and update the 3D visualization in real-time.

Visualization:

- The visualization updates to reflect the drone's orientation, providing a clear, dynamic view of how the drone behaves in 3D space based on its IMU telemetry.