Explanation of droneSettings.cs Code

Overview

The `droneSettings.cs` script serves as a utility class for configuring and stabilizing a drone in a Unity simulation. It defines constants, PID parameters, and utility functions used to control the drone's physics and stabilization behavior. Below is a detailed breakdown of the script.

1. Class Structure

The `droneSettings` class is not a `MonoBehaviour`. It is a static utility class, meaning all its members are accessible globally without the need for an instance. This design is efficient for defining shared constants and reusable functions.

2. Constants for Drone Physics

These constants ensure the drone operates within realistic physical limits.

Saturation Values

The `saturationValues` class defines boundaries for various physical parameters:

- **Rotation Speed:** Limits the rotation speed between `minRotationSpeed = 0.2f` and `maxRotationSpeed = 3f`.
- **Torque:** Constrained to `[0.2f, 3f]`.
- **Vertical Motion:** Velocity is limited to `[-8f, 15f]`, and acceleration is capped between `[-6f, 3f]`.
- **Horizontal Velocity:** The maximum speed is `5f`.

These limits are crucial for preventing instability in the drone's motion.

3. PID Constants for Stabilization

The script employs PID (Proportional-Integral-Derivative) controllers for stabilizing the drone. Different PID constants are used for various axes of stabilization.

Vertical Stabilization

- **Ideal Constants:** `constVerticalIdealVelocity = 0.385772f`, `constVerticalIdealAcceler = 0.6716582f`.
- **PID Constants:**
  - Proportional: `verticalPID_P = 0.7635331f`.
  - Integral: `verticalPID_I = 0.001476288f`.
  - Derivative: `verticalPID_D = 0.0001088255f`.
  - Utility Factor: `verticalPID_U = 0.1f`.

Axis Stabilization (X and Z Axes)

- **Ideal Constants:** `constAxisIdealVelocity = 0.482393f`, `constAxisIdealAcceler = 0.9510251f`.
- **PID Constants:**
  - Proportional: `axisPID_P = 0.2242663f`.
  - Integral: `axisPID_I = 6.129676E-05f`.
  - Derivative: `axisPID_D = 0.007565225f`.
  - Utility Factor: `axisPID_U = 0.5f`.

Yaw Stabilization (Rotation Around the Vertical Axis)

- **Ideal Constants:** `constYawIdealVelocity = 0.5534329f`.
- **PID Constants:**
  - Proportional: `yawPID_P = 0.07649516f`.
  - Integral: `yawPID_I = 2.469936E-05f`.
  - Derivative: `yawPID_D = 0.002099928f`.
  - Utility Factor: `yawPID_U = 0.2f`.

Horizontal Stabilization (Roll and Pitch)

- **Ideal Constants:** `constHorizontalIdealVelocity = 0.9380789f`, `constHorizontalIdealAcceler = 0.9398623f`.
- **PID Constants:**
  - Proportional: `orizPID_P = 0.05998019f`.
  - Integral: `orizPID_I = 5.116195E-05f`.
  - Derivative: `orizPID_D = 0.002372454f`.
  - Utility Factor: `orizPID_U = 0.3f`.

4. Utility Functions

Normalization and Clamping

- **normalizeBetween:** Maps a value to a specific range.
- **keepOnRange:** Ensures a value remains within the range $[lbound, ubound]$.
- **keepOnAbsRange:** Similar to `keepOnRange`, but for symmetric ranges $[-bound, bound]$.

Range Checks and Zeroing

- **isInsideRange:** Checks if a value lies within a range.
- **setZeroIflessThan:** Sets a value to zero if it's below a threshold.
- **setZeroIflessThan (Vector3):** Applies zeroing to each component of a `Vector3`.

5. Design and Usage

- **Realism:** The constants and saturation values ensure the drone behaves realistically.
- **Modularity:** The PID constants and utility functions can be reused across different stabilization scripts.
- **Scalability:** Adding new stabilization features or parameters is straightforward.