In ArduPilot, quadcopter tuning is structured in three layers: the *inner loop* (rate control), the *middle loop* (attitude control), and the *outer loop* (position/velocity control). Each layer corresponds to a set of parameters that govern stability and responsiveness.

🌀 Inner Loop — Rate Control

- **Purpose**: Directly controls angular rates (how fast the copter rotates about roll, pitch, yaw).
- **Parameters**:
  - `ATC_RAT_RLL_P`, `ATC_RAT_RLL_I`, `ATC_RAT_RLL_D` → Roll rate PID gains
  - `ATC_RAT_PIT_P`, `ATC_RAT_PIT_I`, `ATC_RAT_PIT_D` → Pitch rate PID gains
  - `ATC_RAT_YAW_P`, `ATC_RAT_YAW_I`, `ATC_RAT_YAW_D` → Yaw rate PID gains
- **Analogy**: Think of this as the "muscle reflexes" of the quad — it ensures the motors respond instantly to stabilize angular rates.

🎯 Middle Loop — Attitude Control

- **Purpose**: Controls the aircraft's orientation (roll/pitch angles, yaw heading).
- **Parameters**:
  - `ATC_ANG_RLL_P` → Roll angle proportional gain
  - `ATC_ANG_PIT_P` → Pitch angle proportional gain
  - `ATC_ANG_YAW_P` → Yaw angle proportional gain
- **How it works**: The attitude controller sends commands to the inner loop (rate controller). For example, if you command a 10° pitch, the attitude loop calculates the rate needed and passes it down.
- **Analogy**: This is the "balance sense" — keeping the quad at the angles you command.

🌍 Outer Loop — Position & Velocity Control

- **Purpose**: Controls navigation in space — holding position, altitude, or following waypoints.
- **Parameters**:
  - `PSC_POSXY_P`, `PSC_POSZ_P` → Position controller gains (horizontal and vertical)
  - `PSC_VELXY_P`, `PSC_VELXY_I`, `PSC_VELXY_D` → Velocity controller gains (horizontal)
  - `PSC_VELZ_P`, `PSC_VELZ_I`, `PSC_VELZ_D` → Velocity controller gains (vertical)
- **How it works**: The position controller calculates desired attitude and velocity, then passes those targets down to the middle loop.
- **Analogy**: This is the "navigation brain" — deciding where the quad should be, while the inner/middle loops handle how to get there.

🧠 Putting It All Together

- **Inner loop (rate)**: Fastest, stabilizes raw angular rates.
- **Middle loop (attitude)**: Ensures the quad holds commanded angles.
- **Outer loop (position/velocity)**: Ensures the quad moves or holds position in 3D space.
- Each layer feeds into the next: *outer → middle → inner*.
- Tuning usually starts with the **inner loop first** (rate PIDs), then attitude, then position. If the inner loop isn't solid, the outer loops will never behave well.

✅ So when you're tuning ArduPilot:

- Start with **rate PIDs (`ATC_RAT_*`)**.
- Then refine **attitude PIDs (`ATC_ANG_*`)**.
- Finally, adjust **position/velocity PIDs (`PSC_*`)** if you're flying missions or GPS-based modes.
