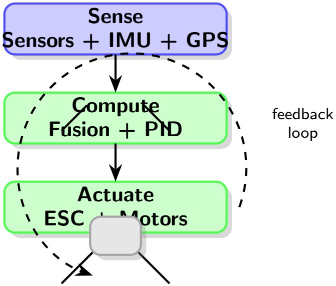
Flight Controller and Sensors
Unit 3

Drone Technology (DI05000081)

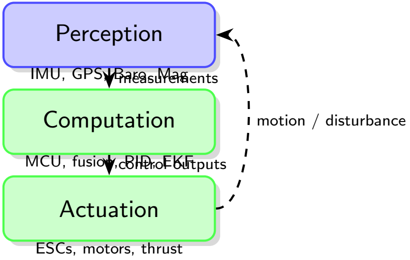
Gujarat Technological University



Unit 3 Roadmap: Perception → Computation → Actuation

Topics Covered

1. Flight controller architecture and microcontroller role
2. Sensor interfaces: I2C, SPI, UART
3. Accelerometer, gyroscope, magnetometer, barometer, GPS
4. IMU construction and placement
5. Sensor fusion: complementary, Mahony, Madgwick, Kalman
6. Stabilisation, feedback loops, PID control and tuning
7. Optical flow, rangefinders, and full EKF architecture



Learning Goal

By the end of Unit 3, you should be able to explain how raw sensor data becomes motor commands that keep a drone balanced, navigated, and autonomous.

3.1 Flight Controller: Definition and Importance

What is a Flight Controller?

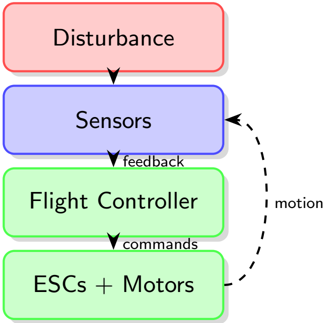
The flight controller (FC) is the **brain of the drone**. It continuously reads sensors, estimates attitude, compares actual motion with pilot or mission commands, and sends corrective outputs to the ESCs.

Human Body Analogy

- **Sensors** ≈ eyes, ears, inner ear
- **Flight controller** ≈ brain + spinal cord
- **Motors/propellers** ≈ muscles

Why 400–8000 Hz?

A quadrotor is naturally unstable. Fast loops are needed so the FC can react within milliseconds to gusts, motor mismatch, or pilot stick changes before the drone tips over.



If FC fails: immediate loss of stabilisation, flyaway, crash, or uncontrolled shutdown.

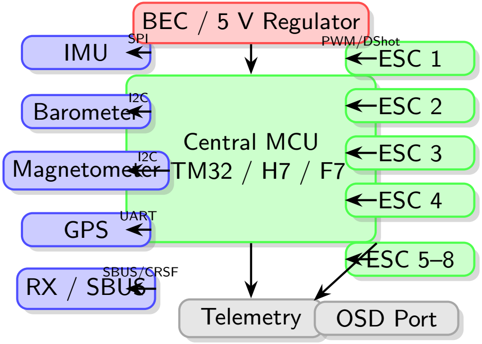
Detailed Flight Controller Block Diagram

Functional View

- **MCU**: executes sensor reading, estimation, mixing, failsafe
- **Inputs**: RC receiver via SBUS/CRSF/UART
- **Sensors**: IMU, GPS, barometer, magnetometer
- **Outputs**: PWM or DShot to 4–8 ESCs
- **Utility ports**: telemetry, OSD, USB, buzzer
- **Power**: regulated 5 V from BEC / power module

Closed-Loop Role

FC software performs state estimation, compares setpoint and actual state, then mixes the corrections to each motor.



Flight Controller Examples and Comparison

How to Choose an FC

- Processor performance and memory
- Sensor quality and redundancy
- Supported protocols and firmware ecosystem
- Weight, power budget, and vibration tolerance
- Mission type: racing, mapping, hobby, industrial

Typical Segments

Professional: Pixhawk / CUAV
Racing: Betaflight F7/H7
Hobby: Kakute class boards
Consumer: tightly integrated DJI stacks

FC	MCU	Onboard sensors	Protocols	Wt.	Cost	Use-case
Pixhawk 6C	H743	Dual IMU, baro	CAN, UART, I2C, PWM	39 g	Rs. 18k	Survey / VTOL
Betaflight F7	F722	IMU, OSD	SPI, UART, DShot	8 g	Rs. 5k	Racing quad
Kakute H7	H743	IMU, baro, OSD	UART, I2C, DShot	9 g	Rs. 7k	Hobby freestyle
DJI Naza	Proprietary	IMU, GPS combo	CAN-like bus, PWM	25 g	Rs. 15k	Consumer ease
CUAV X7	H743	Triple IMU, dual baro	CAN, UART, I2C, PWM	50 g	Rs. 32k	Industrial UAV

Observation

Racing boards optimise latency and weight; industrial autopilots prioritise redundancy, CAN peripherals, and long mission reliability.

3.2 Microcontroller Role in the Flight Controller

MCU Responsibilities

Modern FCs use **STM32F4/F7/H7 ARM Cortex-M** microcontrollers running roughly **168–400 MHz**. They must:

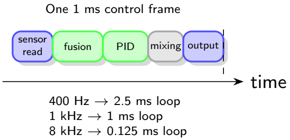
- sample sensors at deterministic intervals,
- execute fusion and PID loops,
- decode RC / mission commands,
- update motor outputs every cycle,
- log data and enforce failsafes.

Timing Insight

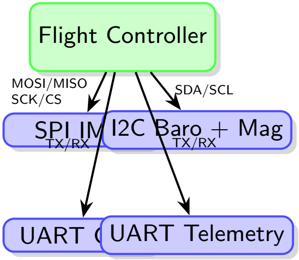
A **1 ms loop** means **1 kHz**. If the loop takes longer than its budget, output latency rises and control quality degrades.

RTOS and Firmware

Autopilots such as **PX4** and **ArduPilot** often run on an RTOS for task scheduling, while **Betaflight** is highly optimised for very fast bare-metal style control loops.



Bus	Wires	Speed	Addressing	Common drone use
I2C	SDA, SCL	up to 400 kHz (typ.)	Yes, shared bus	Barometer, magnetometer, some IMUs
SPI	MOSI, MISO, SCK, CS	up to 21 MHz or more	Chip-select based	High-speed IMU, flash memory, OSD
UART	TX, RX	board dependent	No addressing; point-to-point	GPS, telemetry radios, receivers



Key Idea

Use **SPI** when latency matters, **I2C** when multiple low-speed sensors share a bus, and **UART** for asynchronous serial modules such as GPS or telemetry.

What an Accelerometer Measures

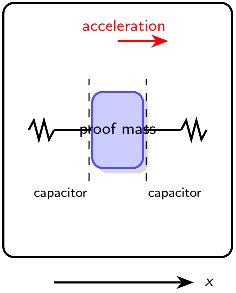
A 3-axis MEMS accelerometer measures **specific force** along the x, y, and z axes. At rest, it mainly senses gravity, so it can infer tilt.

MEMS Construction

A tiny **proof mass** is suspended by micro-springs. Acceleration displaces the mass, changing capacitance. Electronics convert this change into digital acceleration values.

Typical Specifications

- Range: $\pm 2g$ to $\pm 16g$
- Sampling rate: hundreds of Hz to several kHz
- Noise floor: small random vibration-like variation
- Output: a_x, a_y, a_z in g or m/s^2



Accelerometer Applications in Drones

Applications

- Estimate roll and pitch from gravity direction
- Detect strong vibration from damaged props or loose frames
- Trigger crash detection or free-fall logic
- Reveal motor imbalance during tuning and logging

Worked Example

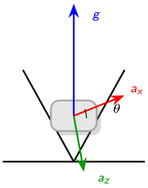
If $a_x = 0.34g$ and $a_z = 0.94g$, then

$$\theta \approx \tan^{-1}\left(\frac{a_x}{a_z}\right) = \tan^{-1}\left(\frac{0.34}{0.94}\right) \approx 19.9^\circ$$

So the drone is tilted by about 20°.

Caution

During aggressive manoeuvres, the accelerometer also senses translational acceleration, so angle estimates become noisy unless fused with the gyro.



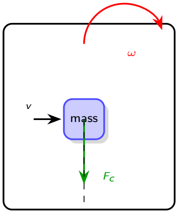
Gyroscope: Coriolis Principle and Drift

Measurement Principle

MEMS gyroscopes contain a vibrating mass. When the sensor rotates, the **Coriolis force**

$$\mathbf{F}_c = 2m(\mathbf{v} \times \boldsymbol{\omega})$$

deflects the mass in a perpendicular direction. This deflection is converted to angular-rate output.



Output and Limits

- Output unit: degrees/s or rad/s
- Measures rate, not absolute angle
- Small bias errors integrate into **drift**
- Therefore gyro-only attitude estimate becomes wrong over time

Why Gyro Alone is Insufficient

Integration of even a tiny bias produces a large angle error after many seconds of flight.

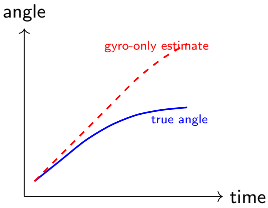
Gyroscope Applications in Drone Stabilisation

Why the Gyro is Critical

- Inner stabilisation loop uses gyro rate directly
- Rate mode / acro mode is almost entirely gyro-driven
- Angle estimate follows

$$\theta(t) = \int \omega(t)\,dt$$

- High update rate gives fast disturbance rejection



Drift Example

A bias of only 0.5°/s creates about 10° of angle error after 20 s:

$$0.5 \times 20 = 10^\circ$$

Complementary Roles

Gyro is excellent for short-term motion; accelerometer and magnetometer correct long-term drift.

Magnetometer: Electronic Compass for Yaw

Principle

A magnetometer measures the Earth's magnetic field along three axes using Hall-effect or magneto-resistive sensing. It acts as a **digital compass**.

Heading Calculation

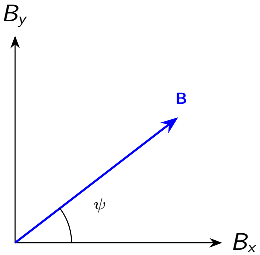
Yaw can be estimated from horizontal magnetic components using

$$\psi = \tan^{-1}\left(\frac{B_y}{B_x}\right) \quad \text{or more robustly} \quad \psi = \text{atan2}(B_y, B_x)$$

followed by local **magnetic declination** correction.

Major Challenge

Motors, ESC wires, batteries, and steel parts distort the magnetic field and can cause large yaw errors if the compass is badly placed.



Units: Gauss or Tesla
$1\ \text{T} = 10^4\ \text{G}$

Why Calibration is Necessary

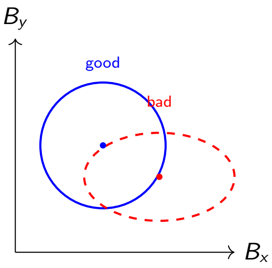
- **Hard iron distortion**: constant offset from permanent magnetic sources
- **Soft iron distortion**: field stretching/rotation due to nearby metals
- Without calibration, heading is biased and navigation modes wander

Procedure

Figure-8 or full 3D rotation calibration collects many samples while rotating the drone around all axes. Software then fits an ellipsoid and corrects it back toward a sphere centred at zero.

Good vs Bad Calibration

Good data forms a near-spherical cloud; bad data forms a shifted ellipsoid and produces unreliable yaw.



Barometer: Pressure-Based Altitude Estimation

Pressure–Altitude Relationship

Atmospheric pressure falls with altitude approximately as

$$P = P_0\, e^{-h/H}, \qquad H \approx 8400\,\mathrm{m}$$

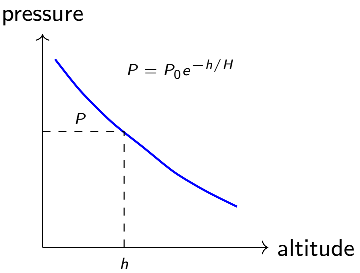
where P_0 is sea-level pressure and h is altitude.

Drone Sensors and Performance

- Common parts: **BMP280**, **MS5611**
- Relative altitude accuracy often about ± 1 to ± 2 m
- Temperature compensation is required because pressure sensors drift with temperature

Use in Flight

Barometers are essential for altitude hold, climb-rate estimation, and as a medium-altitude reference when close-range sensors are out of range.



Constellations

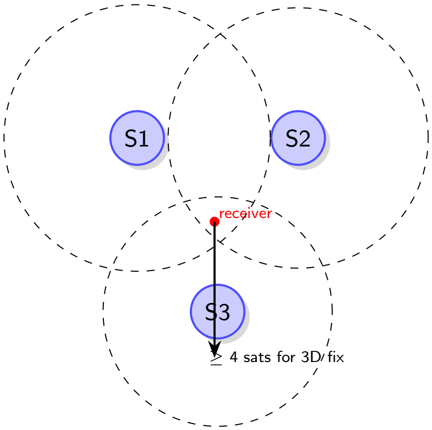
Drones may use **NAVSTAR GPS**, **GLONASS**, **Galileo**, and **NavIC**. More visible satellites generally improve geometry and availability.

How Position is Found

Satellites transmit time-coded signals (for example C/A code). The receiver measures **time of flight** to estimate range. Intersecting multiple ranges gives position by **trilateration**. At least **4 satellites** are needed for a full 3D fix with receiver clock bias.

Accuracy

Standard GNSS gives about **3–5 m** position accuracy; RTK carrier-phase systems can reach about **1–2 cm** relative accuracy.



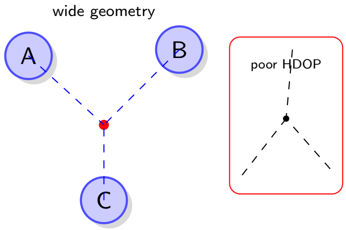
Common GPS Outputs

UART GPS modules often send NMEA strings such as `$GPGGA` (fix data) and `$GPRMC` (recommended minimum data). These include latitude, longitude, altitude, speed, UTC time, and fix quality.

Fix	Meaning	Typical accuracy
No fix	insufficient satellites	unavailable
2D	lat/long only	5–10 m
3D	lat/long/altitude	3–5 m
DGPS	differential corrected	0.5–2 m
RTK float	carrier phase, partial lock	0.1–0.3 m
RTK fixed	full carrier lock	1–2 cm

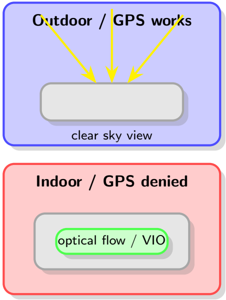
HDOP

Horizontal Dilution of Precision indicates how good the satellite geometry is. Lower HDOP means better position confidence.



Major GPS-Based Functions

- **Position hold / loiter**: resist drift in wind
- **Return-to-Home (RTH)**: recover after signal loss or low battery
- **Waypoint navigation**: autonomous path following
- **Geo-fencing**: prevent flying outside a safe region
- **Precision landing**: often improved by RTK or markers



Why Indoor GPS Fails

Satellite signals are extremely weak and are attenuated by roofs and walls. Multipath reflections also corrupt timing. Therefore GPS-denied environments need alternatives such as **optical flow**, **visual-inertial odometry (VIO)**, or UWB beacons.

3.4 IMU: Inertial Measurement Unit

Definition

An IMU combines an **accelerometer** and **gyroscope**; if a magnetometer is included, the package is often called a 9-DoF module.

6-DoF vs 9-DoF

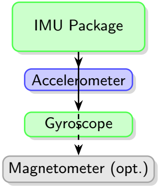
- **6-DoF**: $a_x, a_y, a_z, \omega_x, \omega_y, \omega_z$
- **9-DoF**: 6-DoF $+ B_x, B_y, B_z$

Typical Parts

Part	Strength	Common use
MPU-6050	low cost	basic learning boards
ICM-42688	low noise, fast SPI	FPV / modern FCs
BMI088	better vibration tolerance	industrial / robotics

Placement Rule

Mount the IMU close to the drone's centre of gravity and isolate it from excessive vibration.



Why Fusion is Required

No single sensor is ideal: gyro drifts, accelerometer is disturbed by linear acceleration, and magnetometer is disturbed by magnetic noise. Fusion combines their strengths.

Complementary Filter

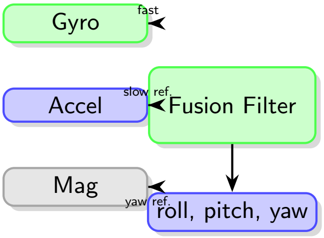
A common idea is

$$\hat{\theta} = \alpha\,\hat{\theta}_{gyro} + (1-\alpha)\,\theta_{acc/mag}$$

where α is close to 1. Gyro handles fast motion; accelerometer and magnetometer slowly pull the estimate back from drift.

Beyond the Basic Filter

Mahony and **Madgwick** filters are lightweight nonlinear fusion algorithms widely used on embedded flight controllers.



Kalman Filter Concept

State Estimation Problem

The controller needs the best estimate of states such as angle, velocity, and position, even though sensors are noisy and the dynamics are uncertain.

Predict–Update Logic

- **Predict**: use a motion model to estimate the next state
- **Update**: correct that prediction using fresh sensor data

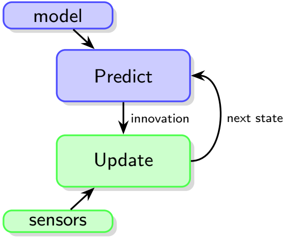
The filter balances trust between model and measurements.

Noise Matrices

Q: process-noise covariance, representing model uncertainty
R: measurement-noise covariance, representing sensor uncertainty

Why It Is the Gold Standard

Kalman filtering provides mathematically consistent estimates and naturally handles multi-sensor fusion.



Meaning of Stabilisation

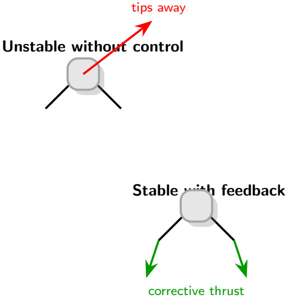
Stabilisation means maintaining the desired attitude or position despite disturbances such as gusts, asymmetrical thrust, or pilot input changes.

Levels of Stabilisation

- **Rate mode**: controls angular velocity
- **Angle / stabilise mode**: controls attitude angle
- **Position mode**: controls location using GPS or vision

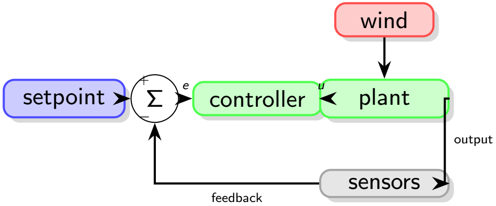
Why Active Stabilisation is Mandatory

A quadrotor is inherently unstable. Without continuous feedback, even tiny disturbances grow into a tip, spin, or crash.



Essential Blocks

- **Setpoint**: desired altitude, angle, or speed
- **Process variable**: measured actual value
- **Error**: setpoint − measurement
- **Controller**: PID or another control law
- **Plant**: motors, propellers, airframe, aerodynamics
- **Feedback path**: sensors and estimator



Altitude Hold Example

Setpoint = 20 m, measurement = baro/GPS altitude, disturbance = wind gust, controller output = throttle correction.

PID Controller Fundamentals

Mathematical Form

$$u(t) = K_p e(t) + K_i \int e(t)\,dt + K_d \frac{de(t)}{dt}$$

Meaning of the Three Terms

- **P**: reacts to present error
- **I**: reacts to accumulated past error
- **D**: reacts to rate of change of error

Physical Interpretation

Large K_p makes the controller aggressive, K_i removes long-term offset, and K_d adds damping and anticipatory action.

Term	Function	Units
K_p	output per error	u/e
K_i	output per error-s	$u/(e \cdot s)$
K_d	output per error rate	$u \cdot s/e$

Drone Example

Error may be roll angle in degrees, yaw rate in deg/s, altitude in metres, or velocity in m/s depending on the loop.

Behaviour of Proportional Gain

- Low K_p: sluggish correction, large tracking error
- Correct K_p: quick response with acceptable overshoot
- High K_p: oscillation and twitchiness

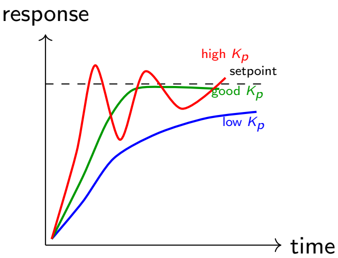
Worked Example

If desired roll is 10° and measured roll is 7°, then error $e = 3^\circ$. With $K_p = 4$ control-units/deg,

$$u_P = K_p e = 4 \times 3 = 12$$

Limitation

P-only control usually leaves **steady-state error** when constant disturbance exists, such as wind or payload bias.



I Term: Integral Control and Windup

Purpose of Integral Action

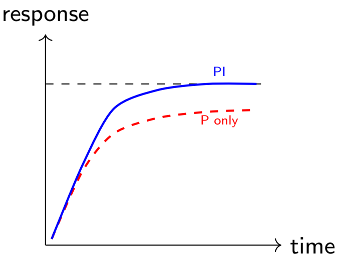
The integral term accumulates error over time, so it removes steady-state bias that P control alone cannot eliminate.

When K_i Matters

If a drone faces constant crosswind or a slightly unbalanced payload, the integral term keeps building correction until the long-term error becomes near zero.

Integrator Windup

If actuators saturate while error persists, the integral term can become too large. When the system recovers, the stored integral causes overshoot. **Anti-windup** limits or resets the integral state.



D Term: Derivative Control and Damping

What the D Term Does

Derivative action responds to how fast the error is changing, so it adds **damping**. It helps stop overshoot and oscillation by resisting rapid motion toward the setpoint.

Numerical Idea

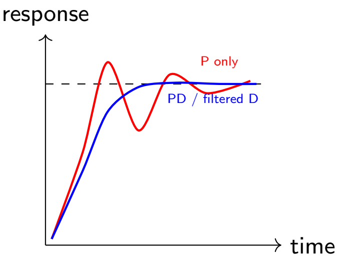
If the error slope is $de/dt = -15$ units/s and $K_d = 0.2$, then

$$u_D = K_d \frac{de}{dt} = 0.2(-15) = -3$$

The negative sign opposes the fast closing motion.

Practical Issue

Derivative action amplifies high-frequency sensor noise, so drone firmware applies low-pass filtering before or within the D term.



PID Velocity Hold Example: Maintaining 5 m/s

Problem Setup

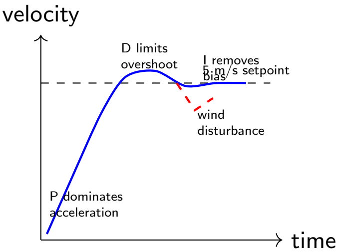
Desired forward velocity $v_{set} = 5$ m/s. Actual velocity is measured by GPS, optical flow, or another estimator.

Controller Behaviour by Phase

- **Start:** $v = 0$, error $= 5$ m/s, so P commands strong acceleration.
- **Approaching setpoint:** error decreases, D damps overshoot.
- **Steady state:** if drag causes $v = 4.7$ m/s, I slowly adds thrust to remove the 0.3 m/s bias.
- **Disturbance:** a headwind drops speed; P reacts immediately, I restores exact 5 m/s, D limits oscillation.

Error Calculation

At any instant, $e_v = v_{set} - v_{actual}$. For example, if $v_{actual} = 3.8$ m/s, then $e_v = 1.2$ m/s.



PID Tuning in Practice

Conceptual Tuning Approaches

- **Ziegler–Nichols**: raise gain to the edge of oscillation, then back-calculate PID values
- **Practical drone sequence**: tune **P first**, then **D**, then **I**
- Start with small steps and observe logs or flight behaviour

Wrong Tuning Symptoms

Over-tuned drones oscillate, under-tuned drones feel soft, and poorly tuned GPS loops may show the famous *toilet bowl* spiral in loiter.

Automation

ArduPilot and some Betaflight workflows offer **auto-tune** features, but pilots still need to understand the physics to judge whether the result is safe.

Symptom	**Likely adjustment**
Sluggish response	increase K_p
Fast oscillation	reduce K_p or increase filtered K_d
Long-term drift / offset	increase K_i
Bounce-back after flip	raise K_d slightly
Noisy motors / hot motors	reduce K_d or improve filtering
Toilet bowl in loiter	compass / GPS tuning, outer-loop gains

Cascade PID Control

Why Cascaded Loops are Used

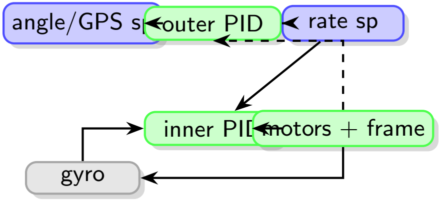
One controller is usually not enough. A drone uses nested loops so that fast inner dynamics are stabilised before slower outer objectives are enforced.

Typical Structure

- **Inner loop**: rate control from gyro data
- **Outer loop**: angle or position control
- Navigation loops may sit above both and generate velocity or angle setpoints

Bandwidth Rule

The inner loop should be roughly **5–10 times faster** than the outer loop. Example: 1 kHz rate loop, 100–200 Hz attitude loop, 10–50 Hz position loop.



Optical Flow Sensor

How It Works

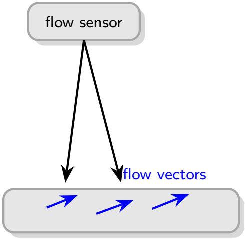
An optical flow sensor uses a downward-facing camera or flow chip to measure how surface texture moves between frames. This gives horizontal motion relative to the ground.

Why It Matters

- Supports indoor hovering where GPS is unavailable
- Helps position hold in warehouses, rooms, and tunnels
- Often paired with a rangefinder so pixel motion can be converted to real ground speed

Limitations

Sensors like **PMW3901** work poorly on glossy, dark, or textureless floors and need adequate lighting.



Rangefinders and Altitude Sensors

Near-Ground Altitude Aids

- **Ultrasonic**: simple and cheap for very low altitude
- **LiDAR / laser**: better precision and range
- **Barometer**: useful above close-range sensor limits
- **Camera depth**: useful in advanced vision stacks

Landing Assist

Autopilots often fuse barometric altitude for general flight with LiDAR or sonar near the ground to obtain smooth flare and touchdown.

Sensor	Example	Range	Typical accuracy
Ultrasonic	HC-SR04	0–4 m	2–5 cm
LiDAR	TF-Mini	0–12 m	1–5 cm
Barometric	BMP280 / MS5611	20 m+	about 1–2 m rel.
Camera depth	stereo / VIO	scene dependent	varies with texture

Design Insight

No single altitude sensor works perfectly at all heights, surfaces, and lighting conditions; fusion is the practical answer.

Complete Sensor Fusion Architecture with EKF

Typical EKF State Vector

$$[x,\ y,\ z,\ v_x,\ v_y,\ v_z,\ \phi,\ \theta,\ \psi]$$

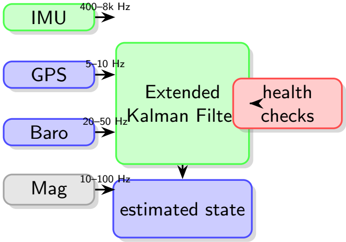
where x, y, z are position, v_x, v_y, v_z are velocity, and ϕ, θ, ψ are roll, pitch, yaw.

Sensor Update Rates

- IMU: hundreds of Hz to kHz
- Barometer: tens of Hz
- Magnetometer: about 10–100 Hz
- GPS: typically 5–10 Hz, higher on premium modules
- Optical flow / rangefinder: tens to hundreds of Hz

EKF Health Monitoring

Autopilots watch innovation residuals, vibration levels, GPS glitches, compass inconsistency, and sensor timeout flags before trusting the estimate.



Key Formulas to Remember

$$\theta \approx \tan^{-1}\left(\frac{a_x}{a_z}\right), \qquad \psi = \text{atan2}(B_y, B_x)$$

$$P = P_0 e^{-h/H}, \qquad u(t) = K_p e + K_i \int e\,dt + K_d \frac{de}{dt}$$

$$\mathbf{F}_c = 2m(\mathbf{v} \times \boldsymbol{\omega}), \qquad \hat{\theta} = \alpha\hat{\theta}_{gyro} + (1-\alpha)\theta_{acc}$$

Sensor	Best at	Weakness
Accel	tilt ref.	disturbed by translation
Gyro	fast rate	drift
Mag	yaw ref.	magnetic noise
Baro	altitude trend	weather / airflow
GPS	global position	weak indoors

Core Takeaways

- FC is the real-time brain of the drone.
- IMU is the primary stabilisation sensor set.
- GPS enables navigation, not basic balancing.
- PID and EKF are the two central algorithm families in this unit.

Review Questions

1. Define a flight controller.
2. Why are fast control loops needed?
3. Compare I2C, SPI, and UART.
4. Explain accelerometer and gyro roles.
5. Why is magnetometer calibration required?
6. How does a PID controller stabilise a drone?
7. Why are cascade loops used?
8. How does EKF combine multiple sensors?