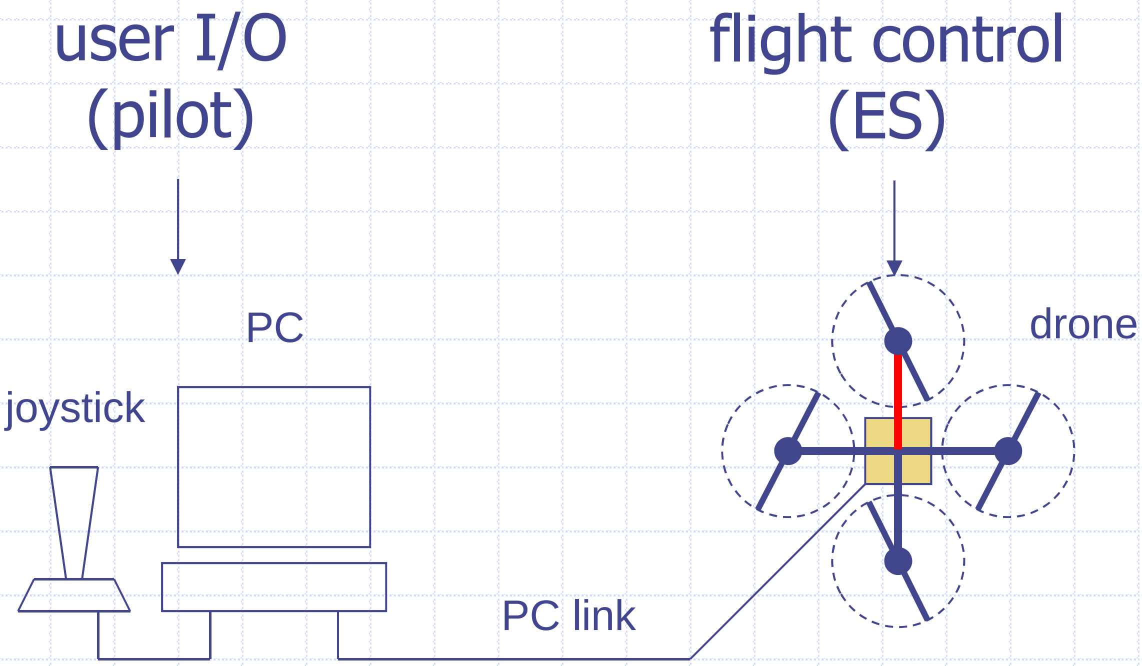
In4073
Embedded Real-Time Systems

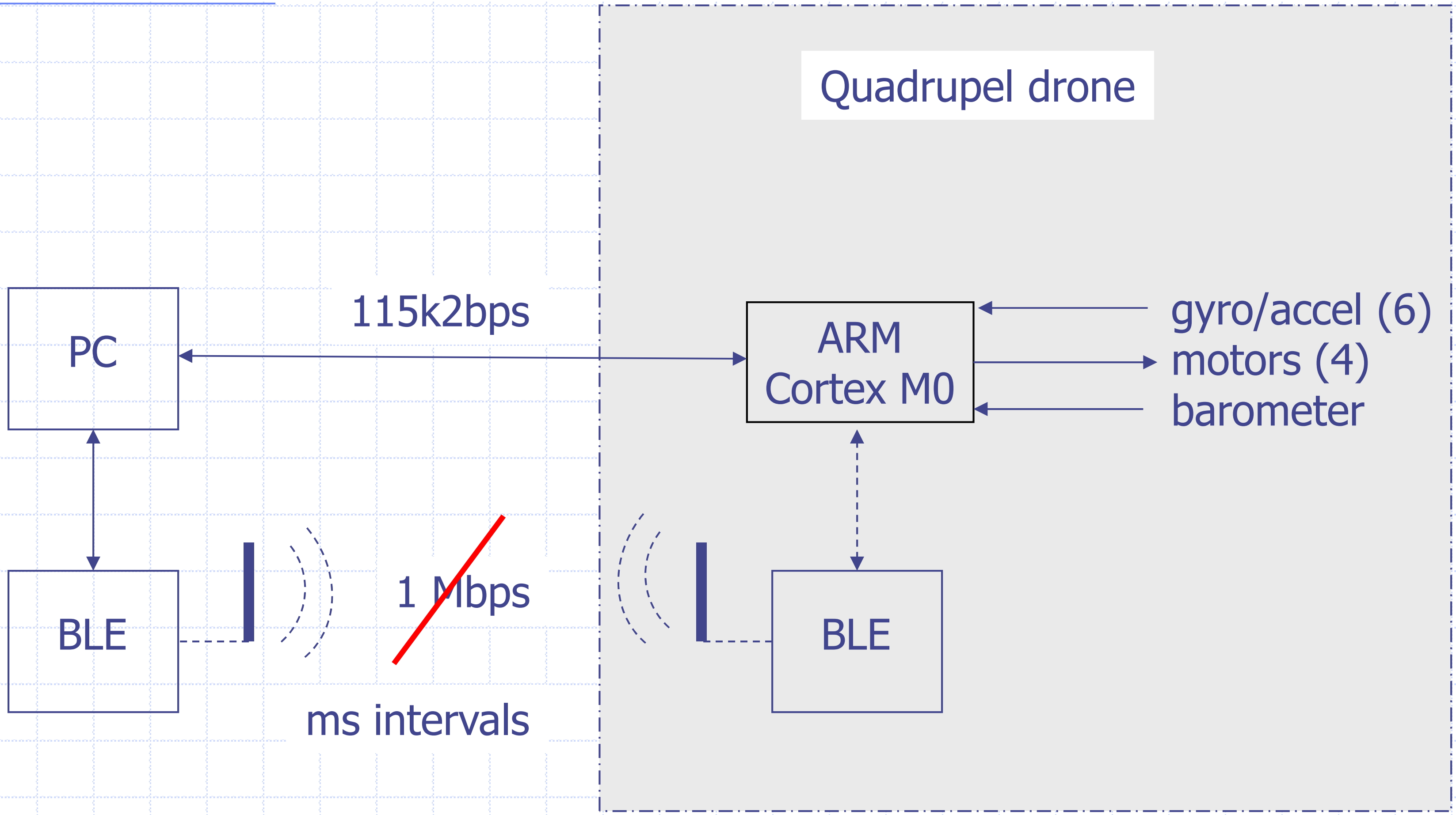
Electrical Model Quad Rotor UAV

System Setup



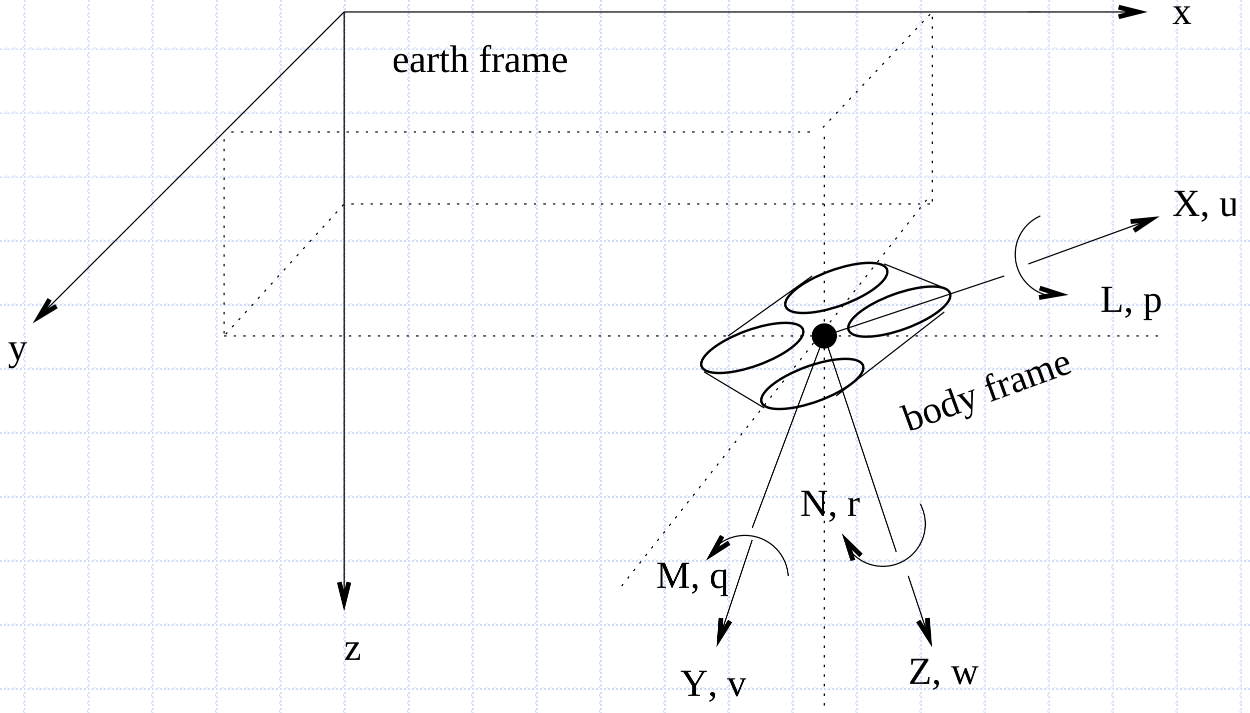
(source: assignment.pdf)

HW view

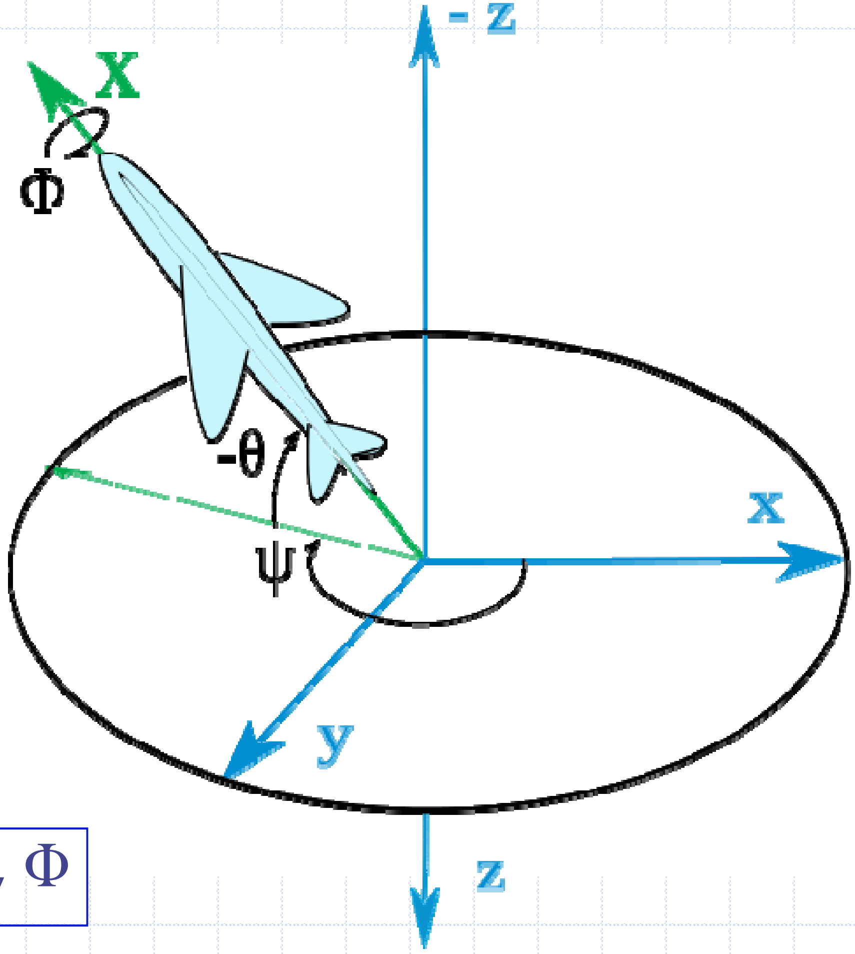


Quadrupel drone
115k2bps
PC
ARM
Cortex M0
gyro/accel (6)
motors (4)
barometer
BLE
1 Mbps
BLE
ms intervals

Drone: Frames & Main Variables

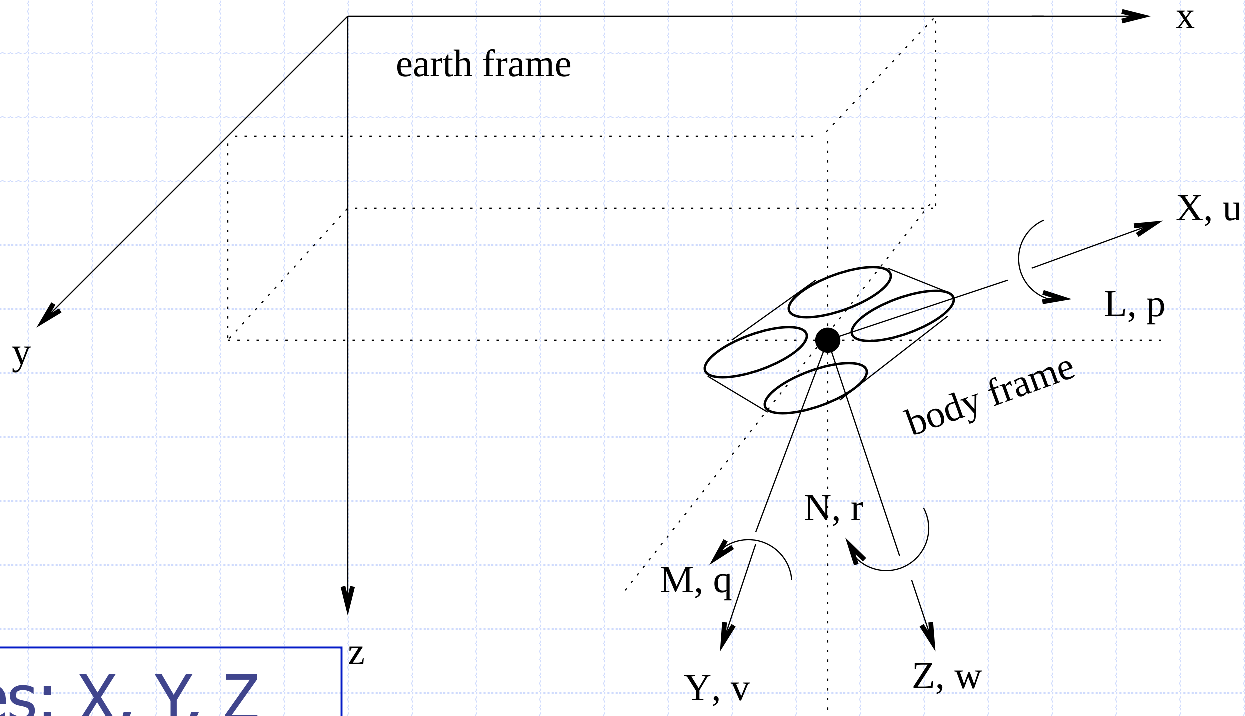


Drone attitude: Euler Angles



rotation order ψ, θ, Φ

Drone: Forces



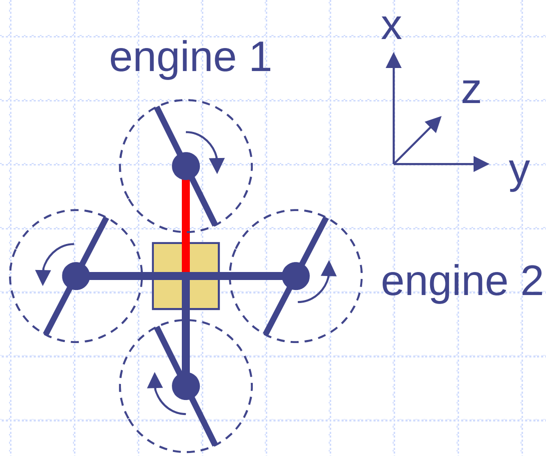
Forces: X, Y, Z
Moments: L,M,N

Drone: Actuators

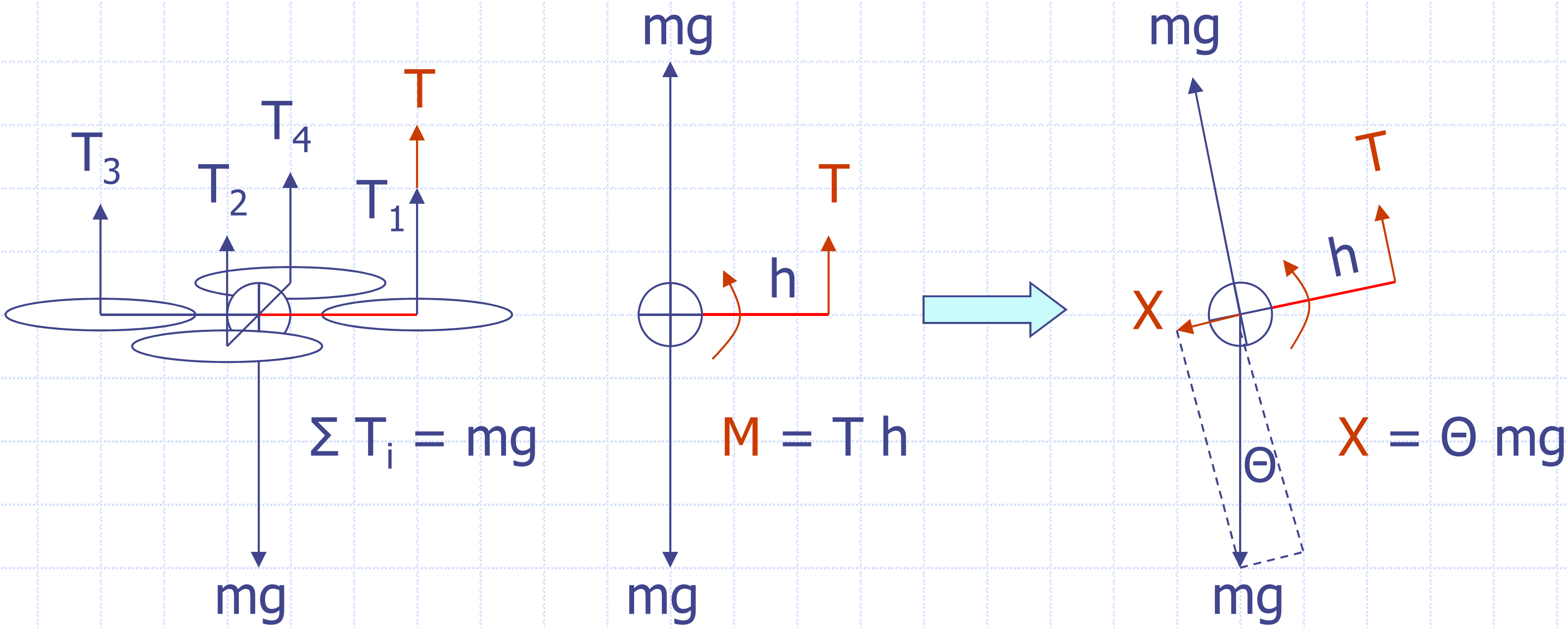
rotor 1 – rotor 4
through RPM,
denoted by Ω

driven by ES signals
ae1 – ae4

ae = `0` -> Ω = 0
ae = `1000` -> Ω = max



Drone: Dynamics (in hover)



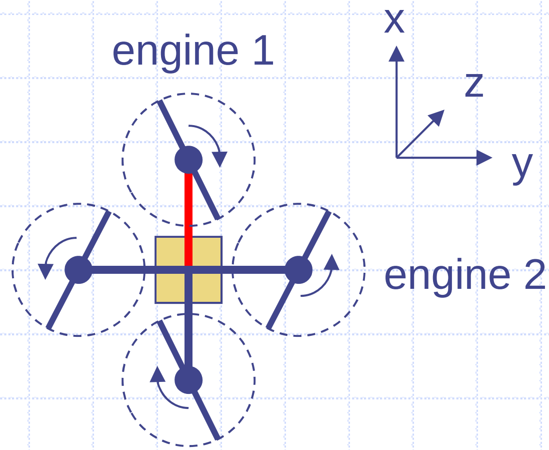
T_i = rotor thrust = $f(\Omega_i)$
mg = gravity
h = rotor distance ref. center of gravity
I_Y = heli rotation inertia in Y-axis

$dq/dt = M / I_Y$

$du/dt = X / m$

accelerated rotation & xlation!

Drone: Rotor Actuators

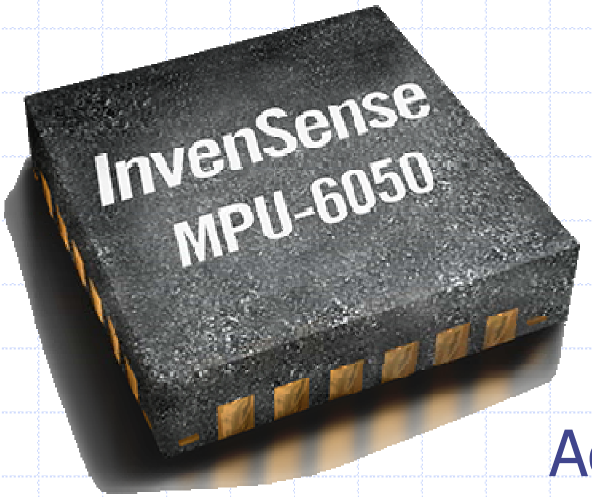


In general

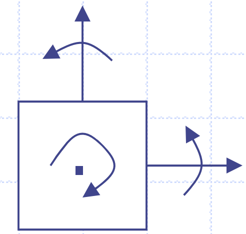
$Z = -b(\Omega_1^2+\Omega_2^2+\Omega_3^2+\Omega_4^2)$
$L = b(\Omega_4^2-\Omega_2^2)$
$M = b(\Omega_1^2- \Omega_3^2)$
$N = d(\Omega_2^2+\Omega_4^2-\Omega_1^2-\Omega_3^2)$

So compute Ω_i (i.e., ae_i) from desired lift (Z), roll rate (L), pitch rate (M), and yaw rate (N)

Drone: Sensors (angles)

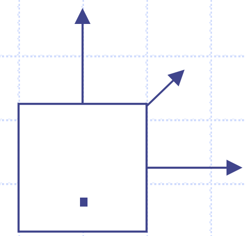


Gyro:

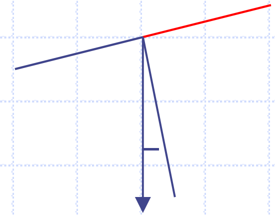


p, q, r

Accelerometer:

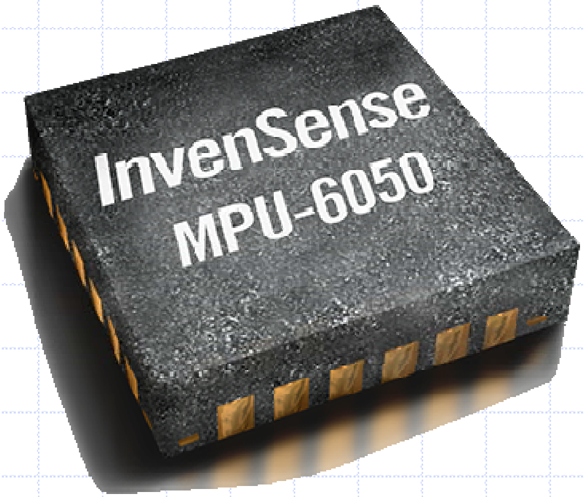


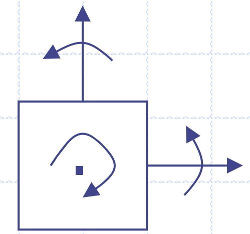
a_x, a_y, a_z

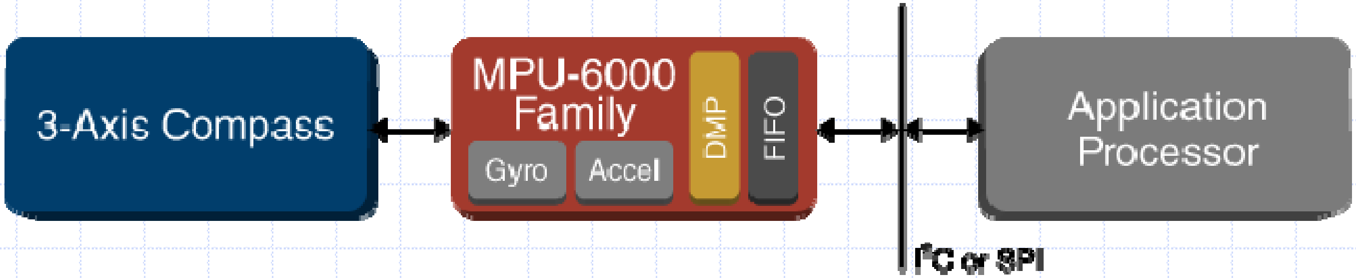


$ax = \sin\Theta \, g \sim \Theta \, g$

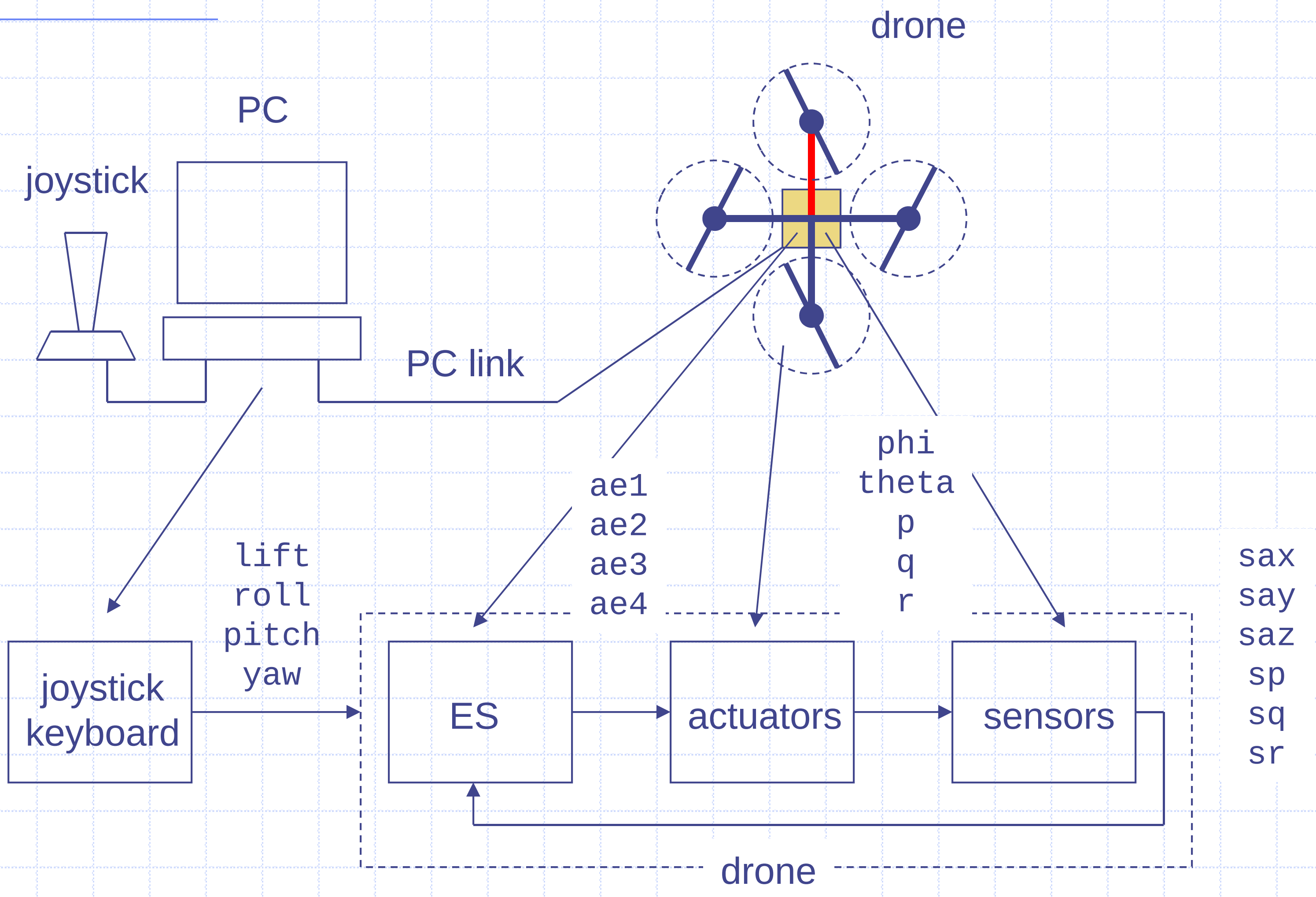
Drone: Sensors (angles)



DMP:  Φ, θ, ψ



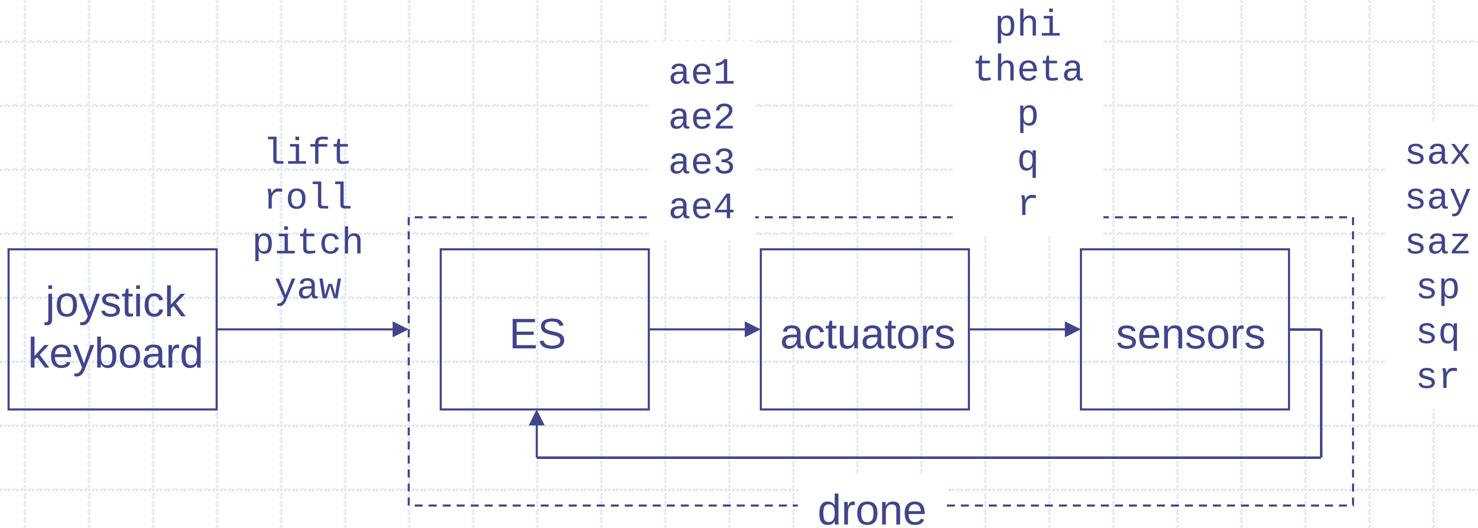
SW view


drone
PC
joystick
PC link
ae1
ae2
ae3
ae4
phi
theta
p
q
r
lift
roll
pitch
yaw
sax
say
saz
sp
sq
sr
joystick
keyboard
ES
actuators
sensors
drone

Communication protocol (lab 1)

- PC -> Drone (send)
  - periodic: pilot control
  - ad hoc: mode changing, param tuning
- Drone -> PC (receive)
  - periodic: telemetry (for visualization)
  - ad hoc: logging (for post-mortem analysis)
- Dependable, robust to data loss
  - header synch

Drone: Control Circuit



control loop example (yaw rate):

eps = yaw - sr; // measure deviation

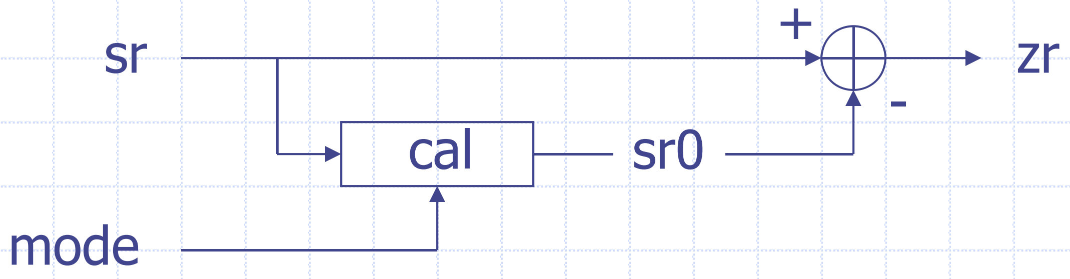
N_needed = P * eps; // compute ctl action

ae1 .. ae4 = f(N_needed); // actuate, see slide 9

Calibration

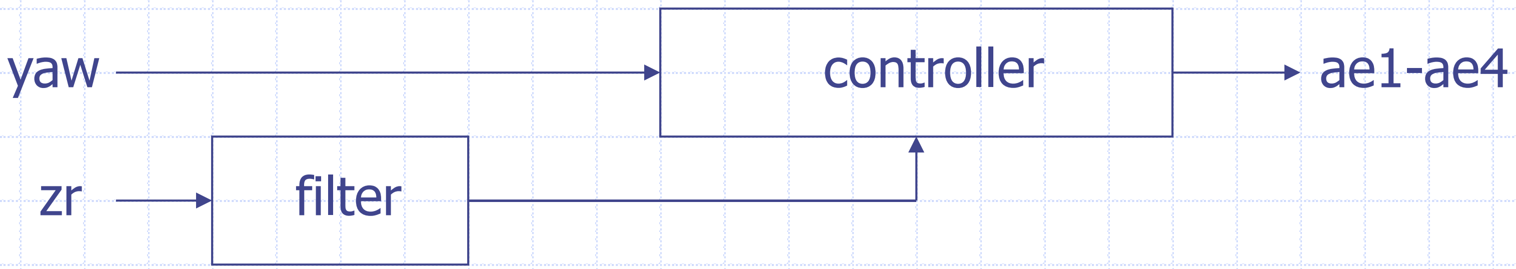
- real p, q, r, .. are sensed in terms of sp, sq, sr, ..
- sp, sq, .. have a (voltage) bias (are not zero at rest)
- so need to calibrate all 6 sensors at rest:
  - let sr0 be sensor output at rest
  - real estimate of r are given by (z for zeroed)

    zr = sr – sr0



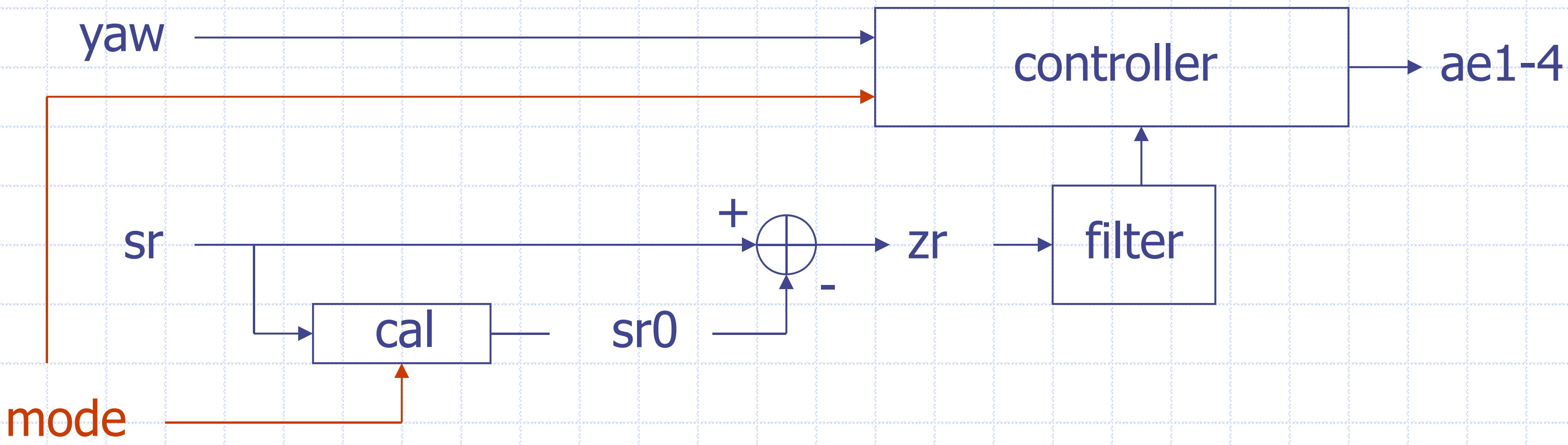
Filtering

- signals also need to be *filtered* to remove noise
- filtered signal input to embedded controller



Controller Modes

- controller mode: manual
- controller model: calibrate
- controller mode: control (yaw, pitch, roll)



Before you go

Safety first:
- goggles
- common sense