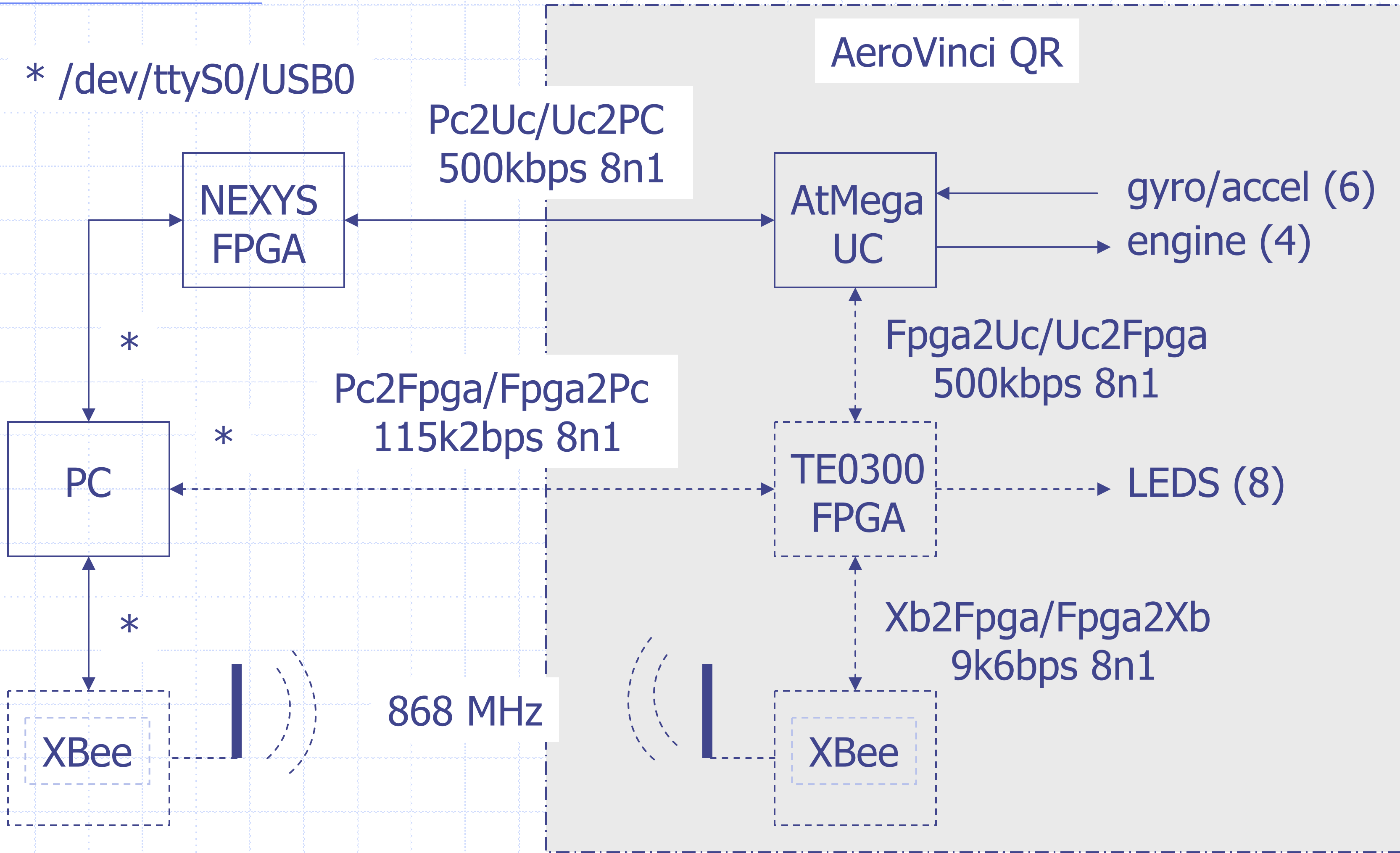
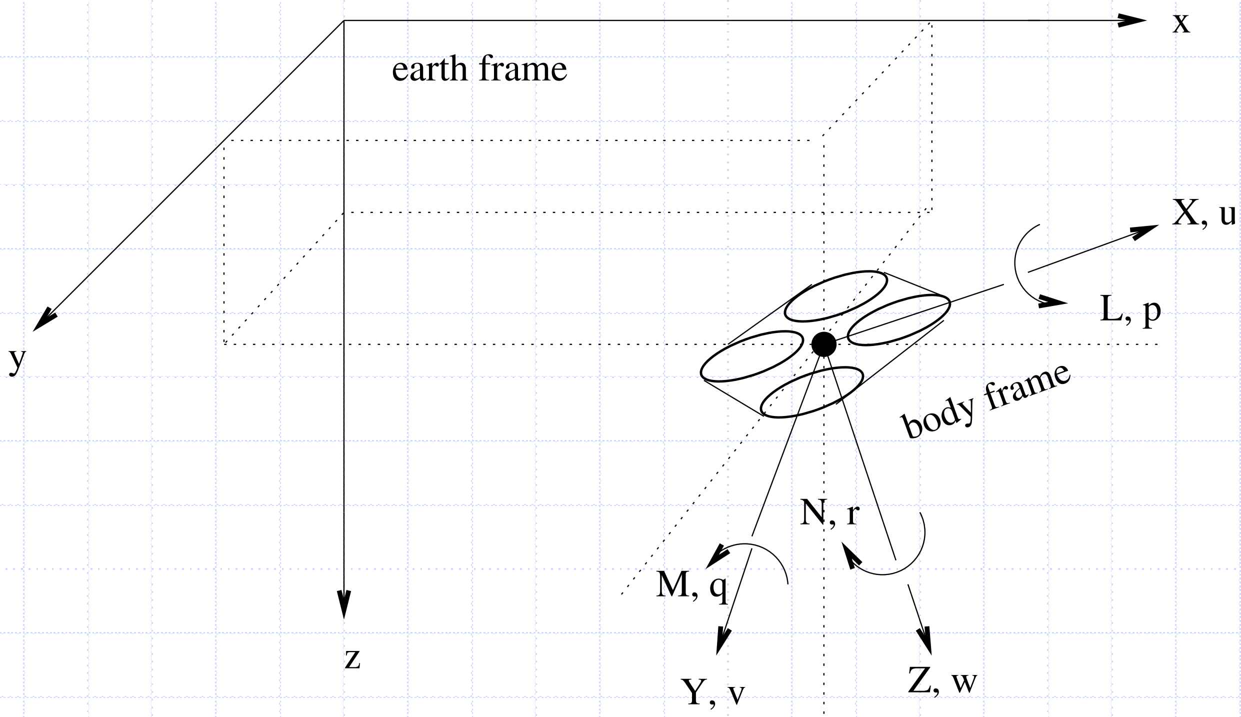
In4073
Embedded Real-Time Systems

Electrical Model Quad Rotor UAV

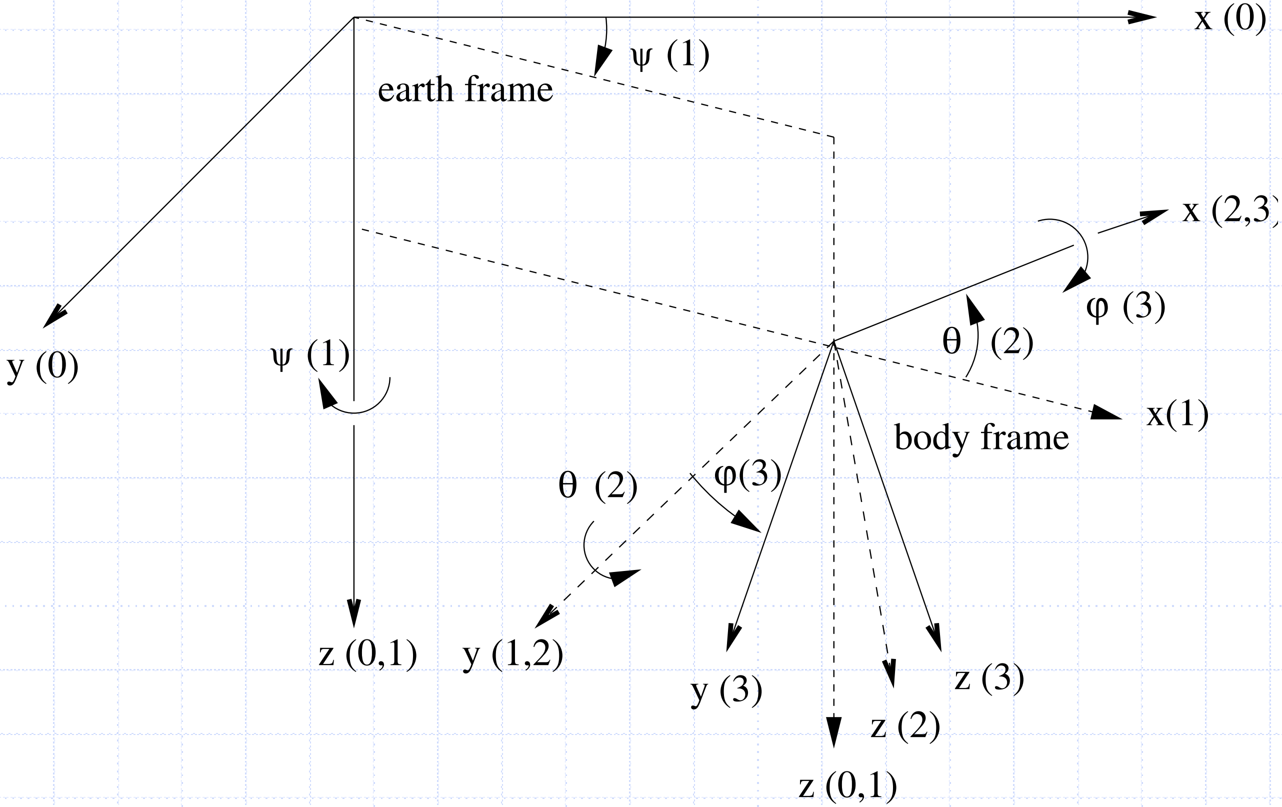
System HW view



QR: Frames & Main Variables



QR Variables: Euler Angles

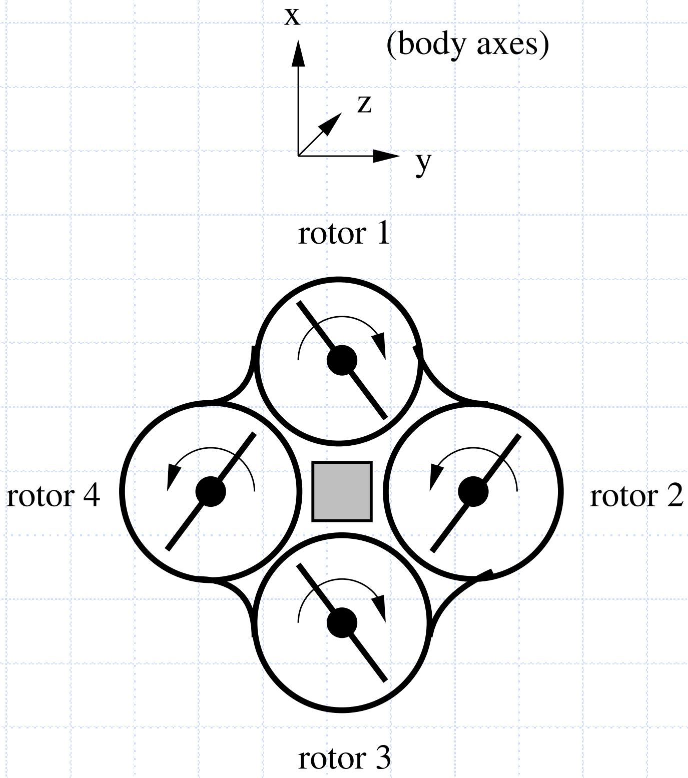


QR: Actuators

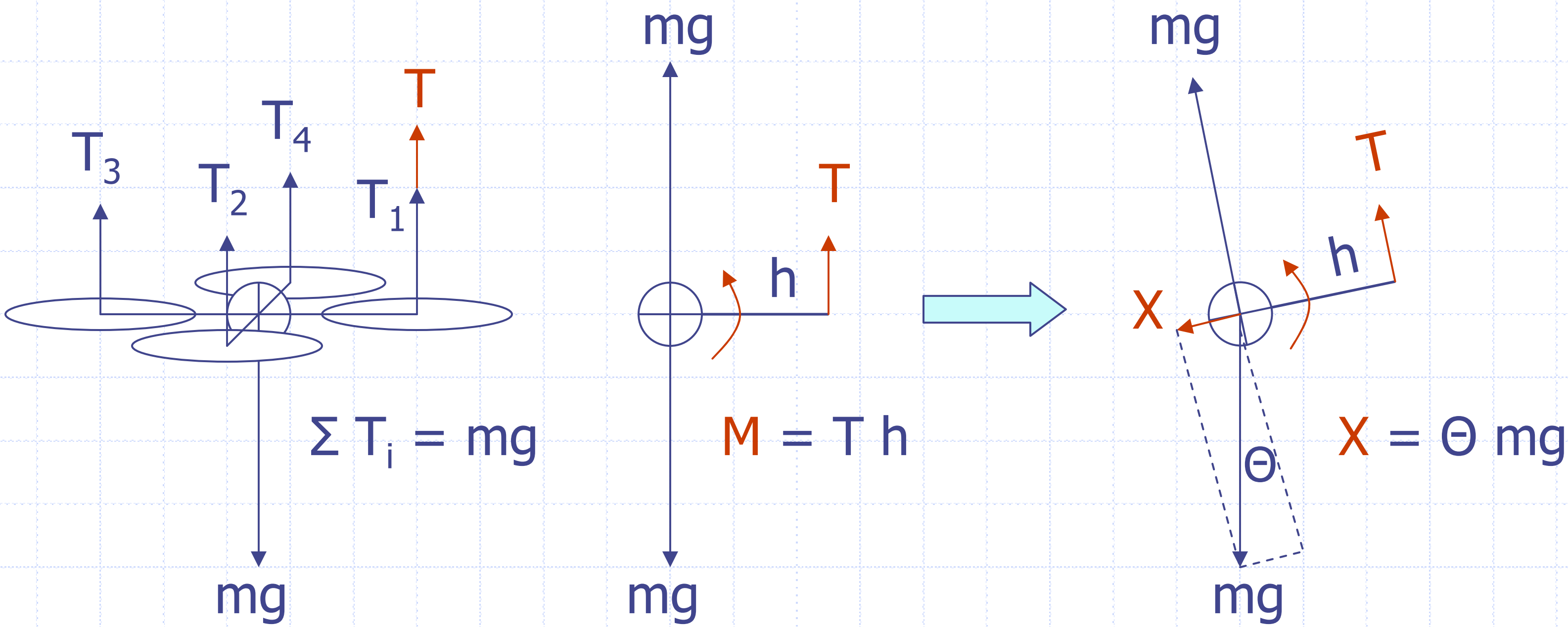
rotor 1 – rotor 4
through RPM, denoted by Ω

driven by ES signals ae1 – ae4

ae = 0x0000 -> Ω = 0
ae = 0x0FFF -> Ω = max



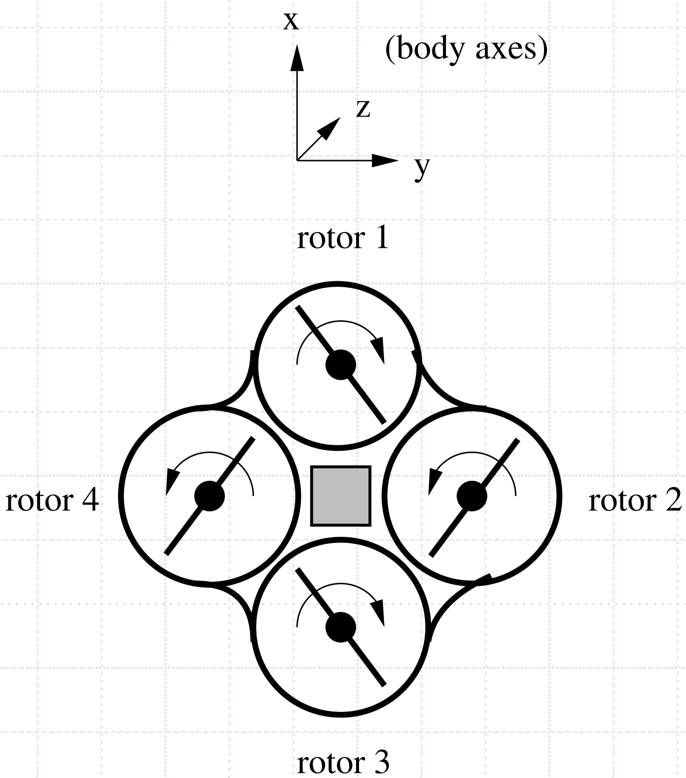
QR Dynamics (in hover)



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!

QR: Rotor Actuators



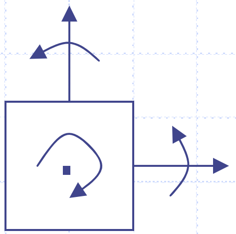
In general

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

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

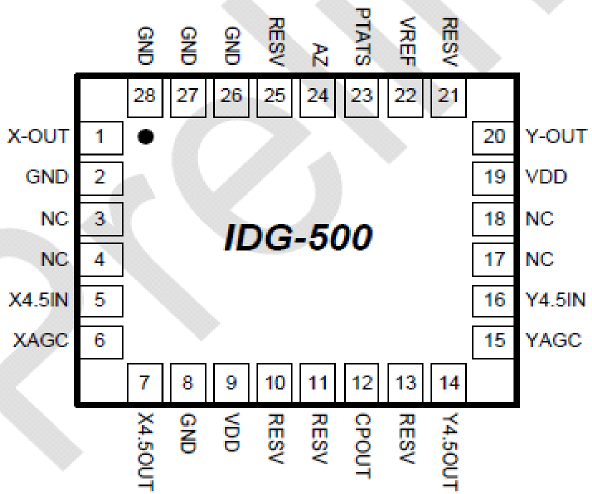
QR: Gyro Sensor

Invensense IDG500



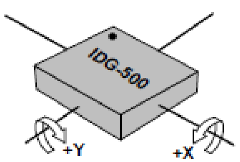
p, q, r

Top View



28-pin, 4mm x 5mm x 1.2mm
QFN Package

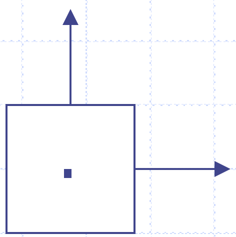
This is a dual-axis rotational-rate sensing device. It produces a positive output voltage for rotation about the X- or Y-axis, as shown in the figure below.



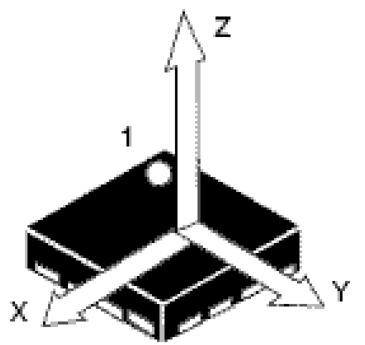
Orientation of Axes of Sensitivity and Polarity of Rotation

QR: Accelerometer Sensor

STMicroelectronics LIS344AL

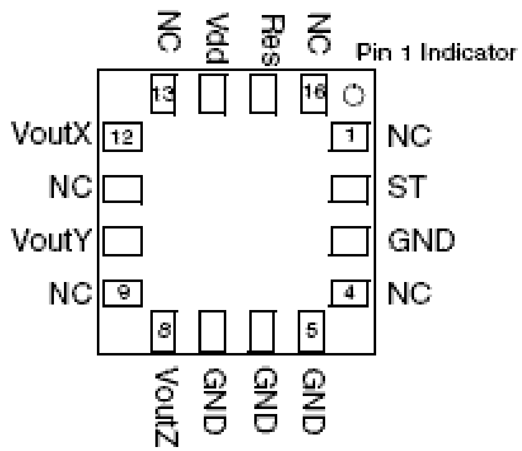


a_x, a_y, a_z

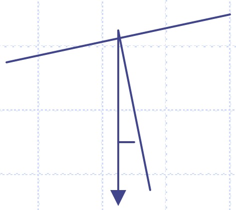


(TOP VIEW)

DIRECTIONS OF THE DETECTABLE ACCELERATIONS

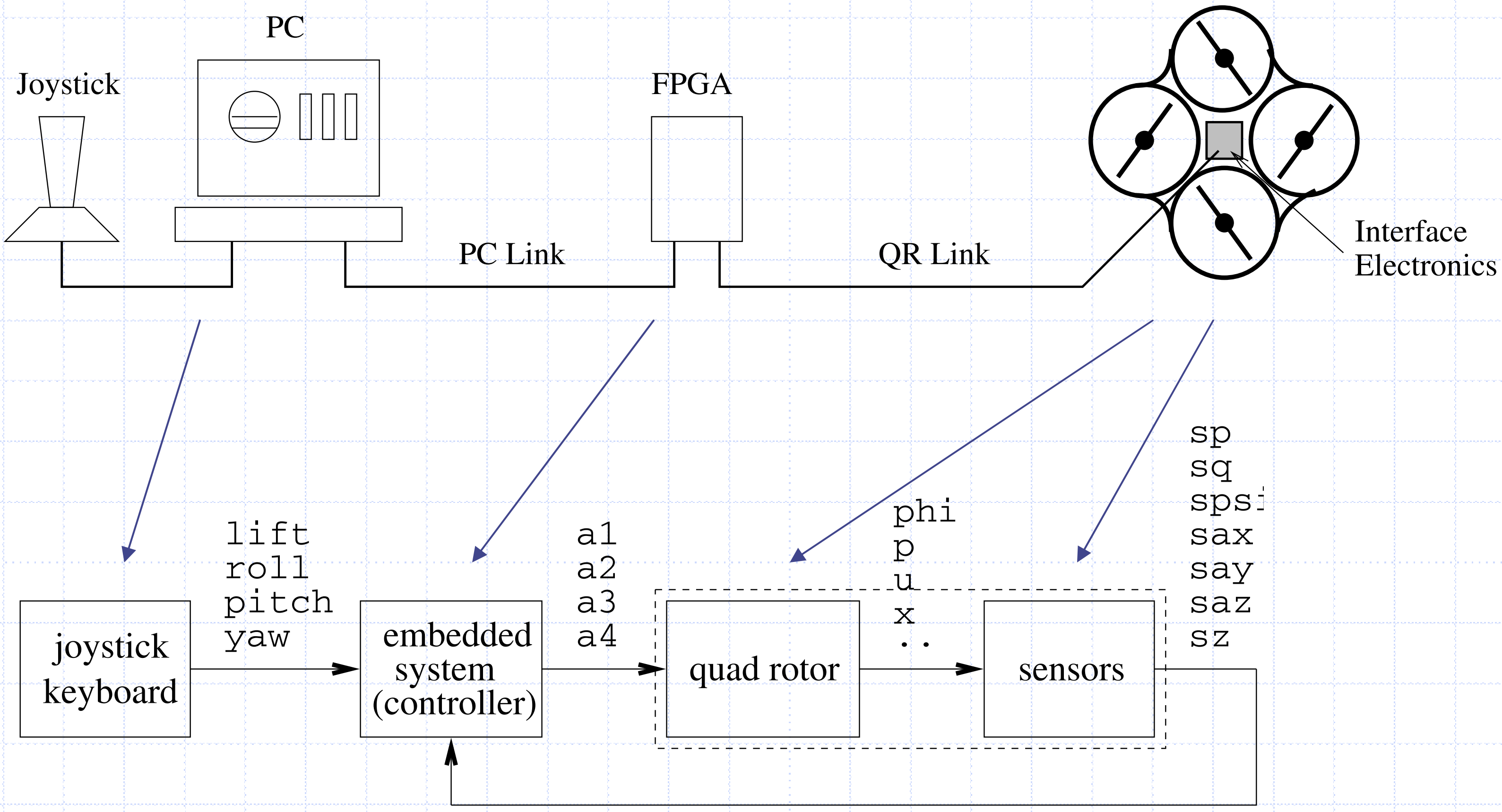


(BOTTOM VIEW)

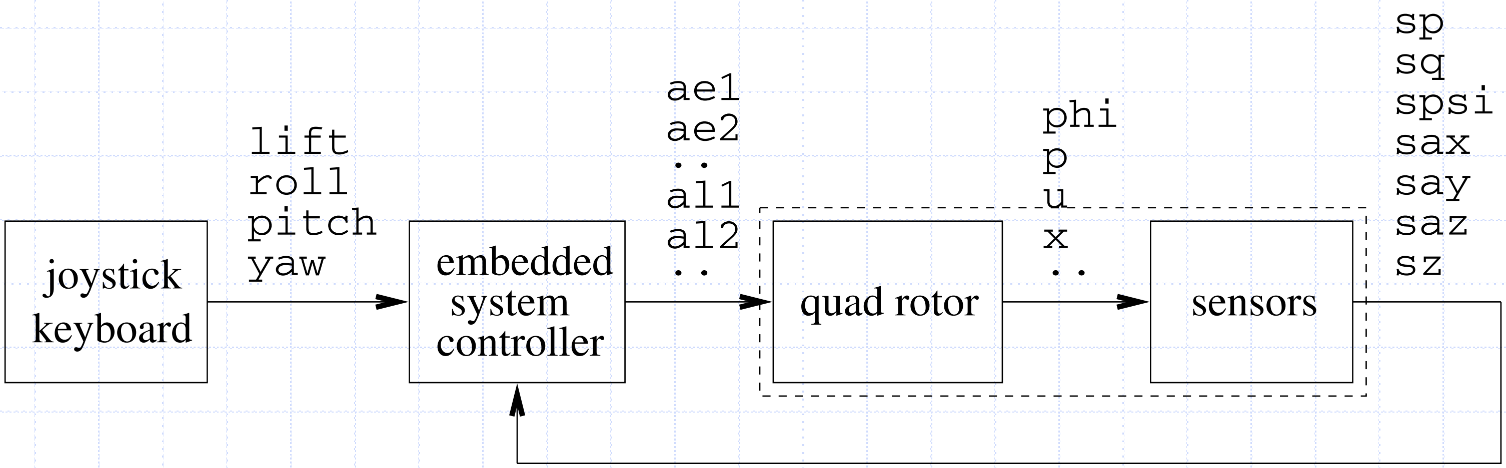


$ax = \sin\Theta\ g \sim \Theta\ g$

System SW view



QR Control Circuit

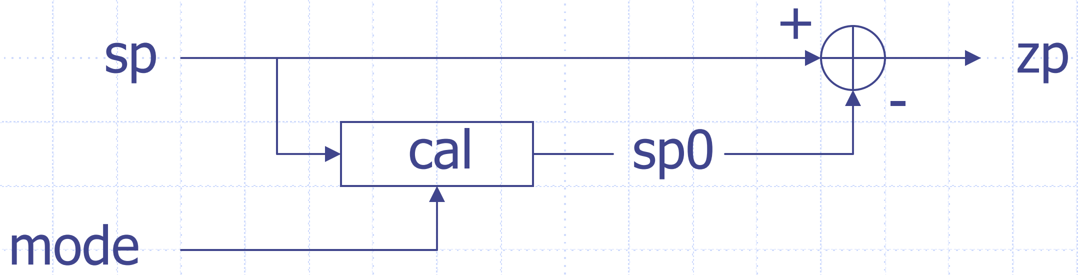


control loop example (roll rate):

```
eps = roll - sp;                    // measure deviation
L_needed = PID(eps);                // compute ctl action
ae1 .. ae4 = f(L_needed);           // actuate, see slide 8
```

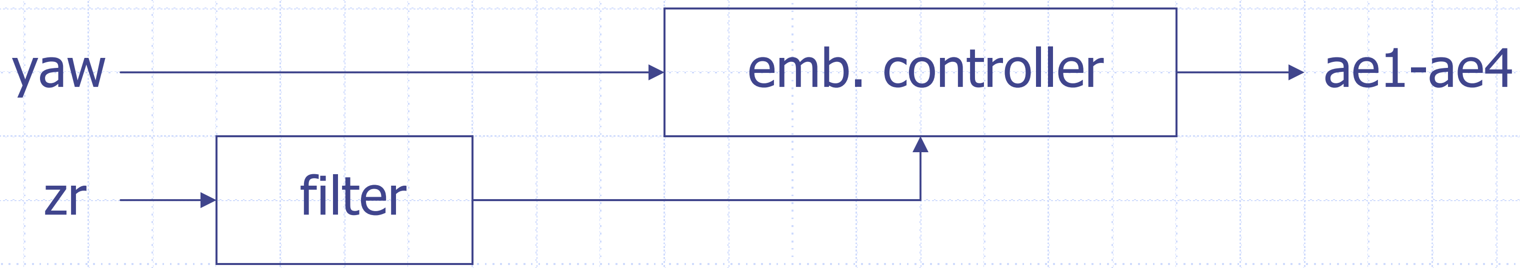
Calibration

- 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 sp0 be sensor output at rest
  - real estimate of p are given by (z for zeroed)
    zp = sp – sp0



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)