Multirotor Control Guidance and Navigation

An introduction

Y. Briere[1]

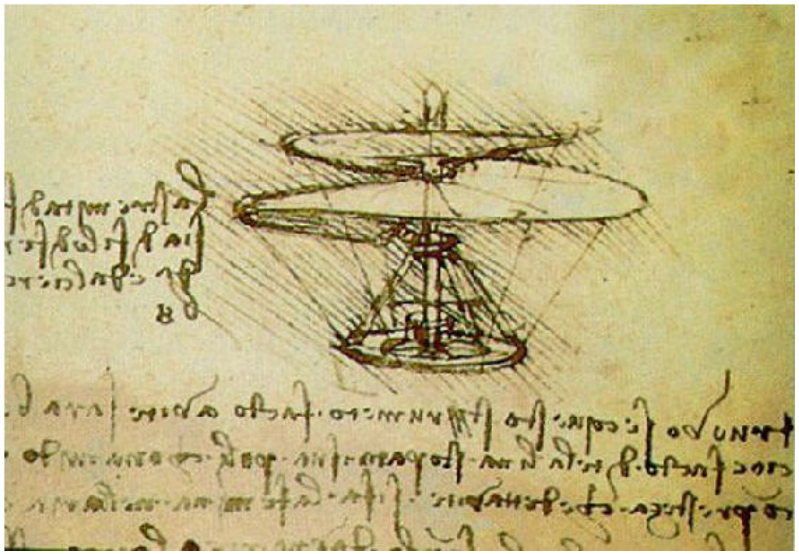
[1]ISAE DCAS

January 2018

Table of Contents

Table of Contents

Super Short Hystory of Quadrotors



Early dreams (Da Vinci 1493)

Modern toys (Parrot 2018)

In 2005 the first conference dedicated to micro UAV ("journées microdrones") was held in Toulouse.
Less than 20 years later the technology is mature.

Why quadrotors ?

Airplane vs Helicopter

Airplanes are nice (fast, hevy loads) but design and control is difficult:

- Wing profile
- Wing and mobile surface surface and shape
- Wing load
- Position of CoG
- Need to actuate mobile surface (ailerons, rudder, elevator)

Helicopter are ice (hovering) but desig, and control is difficult:

- Size and structure of the rotor
- Motion inclination control via the "swashplate" (complex mechanical device that periodically control blade angle of attack)
- Need of rotor tail

Why quadrotors

Quadrotor are simple

Compared to conventional fixed wing or helicopter design, multirotor are extremely simple:

- 4 motorts, 4 propellers, that's it
- Do not need specific mechanical design for control
- Only controlled via speed rotation of motors
- Very well adapted to low cost designs:
  - Few parts
  - Very simple mechanical design
  - Many parts can be bought for nothing (thank you Apple)

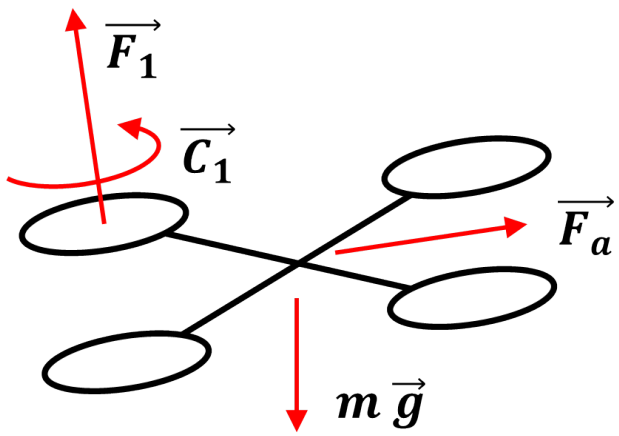
Of course several drawbacks:

- Very poor aerodynamic performance
- Short operation time compared to fixed wing designs
- Efficient lateral control is still a challenge

Objective of the course

- List the main components of a modern quadrotors and corresponding technical challenges
- Understand the main control principles
- List a few control and navigation challenges

How a quadrotor flies

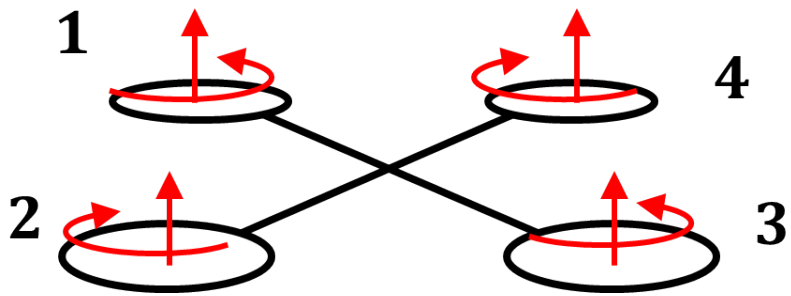


The rotation of the propellers and the motion of the quadrotor creates forces and moments:

- Gravity $m\vec{g}$
- Main effect of the propeller rotation is the thrust : $\overrightarrow{F_i}$
- Secondary effect of the propeller rotation is the counter torque due to rotation of propellers: $\overrightarrow{C_i}$
- Motion of the quadrotor induces aerodynamic forces $\overrightarrow{F_a}$

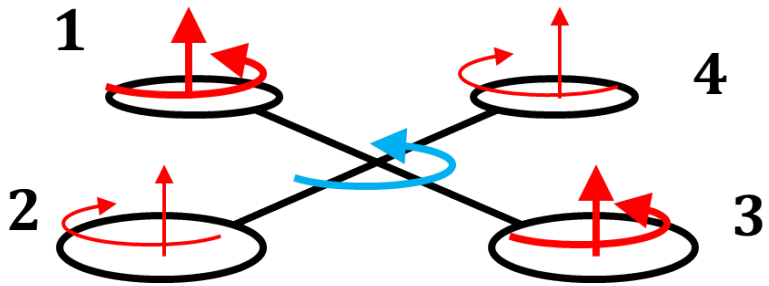
Combination of forces and moments generates a six degrees of freedom motion.

Vertical Motion



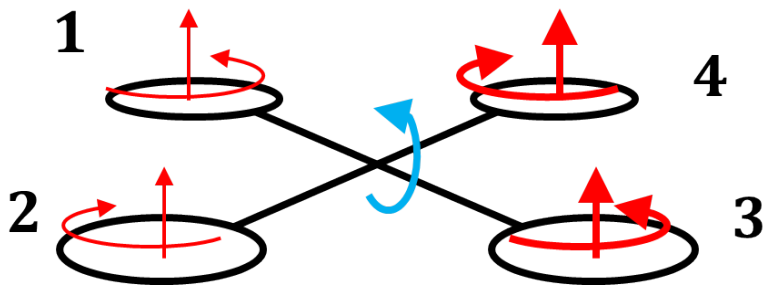
Equilibrium is obtained when $\vec{F_1} + \vec{F_2} + \vec{F_3} + \vec{F_4}$ exactly compensate for gravity and $C_1 + C_2 + C_3 + C_4 = 0$. The equilibrium of torques is obtained thanks to clockwise orientation of propeller 1 and 3 and counterclockwise rotation of 2 and 4.

Yaw motion



A rotation around vertical axis (yaw) is obtained by increasing thrust of 1 and 3 and reducing thrust of 2 and 4. The result is a constant total thrust (no vertical motion) but a positive torque due to increase of C_1 and C_3.

Pitch (and Roll) motion



A rotation around an horizontal axis is obtained by increasing thrust of 3 and 4 and decrease thrust of 1 and 2. The result is a constant torque (no yaw), a constant total force (no vertical motion) but unbalanced vertical forces. The quadrotor consequently tilts and moves horizontally.

Table of Contents

The multi-objective Optimization Problem

The design of a quadrotor is the result of a multi objective optimization process :

- Carry the payload (image sensor, communication, etc.)
  $\rightarrow$ Need enough power
- Perform the task (play, take pictures, etc.)
  $\rightarrow$ Need enough energy
- Reduce cost (including maintenance)

In order to save power and energy the main parameter to be reduced is the weight, the main property to enhance is efficiency.

The geometry

The Parrot Bebop

The Eagle octocopter

KC X8 drone

Most multi-rotor design are symmetrical:

- Most popular is the quadrotor
- Hexacopter, Octocopter can be programmed to deal with the loss of one or more motor/propeller. (Adapted to heavy payload)
- X8 : quadcopters with 4 pairs of counter-rotating blades. (Adapted to heavy payload)

The size of the design depends mainly to the payload and task

Propulsion

The force is generated by rotating propellers. The pair motor/blade is chosen together.

- Most multi-rotor uav use brushless motors. The differ with inner or outer mounting of the magnets:
  - Inner mounting: fixed coils, magnets mounted to the armature shaft which spins into the casing.
  - Outer mounting : magnets mounted on the rotating outer casing, the coils are fixed in the center.
- Most propellers have two blades. Main parameters are:
  - Diameter
  - Angle of attack, clockwise or counter-clockwise.

  → Example: a 5030 propeller will have a 50mm diameter and a 30mm (ideal) translation by turn
  - Efficiency.

Power conversion

An Electronic Speed Controller

The ESC converts the control signal into a suitable sequence of currents to the motor.

- Inputs:
  - Power (directly from the battery)
  - Control signal from the computer
- Outputs:
  - Three phase current to the motor

The ESC contains the power conversion and a micro-controller dedicated to motor control, sometimes programmable.

Batteries

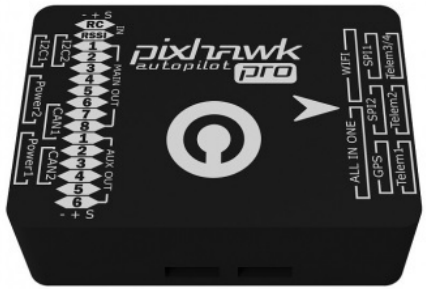
Most batteries are (in 2018) Lithium Polymer:

- Good ratio power/mass
- Good Transient performance (peak current)
- Good value

Example a Lipo 3S 25C 11.1V 2000mAH means:

- Lithium Polymer techno
- 3S for three cells (each 3,7V) in series...
- ... result is 11.1V
- 25C is the capacity: means the battery can deliver 25 times its capacity in constant current (here 50A)
- 2000mAh is the capacity (2A during 1 hr)

Flight Controller



The Pixhawk controller

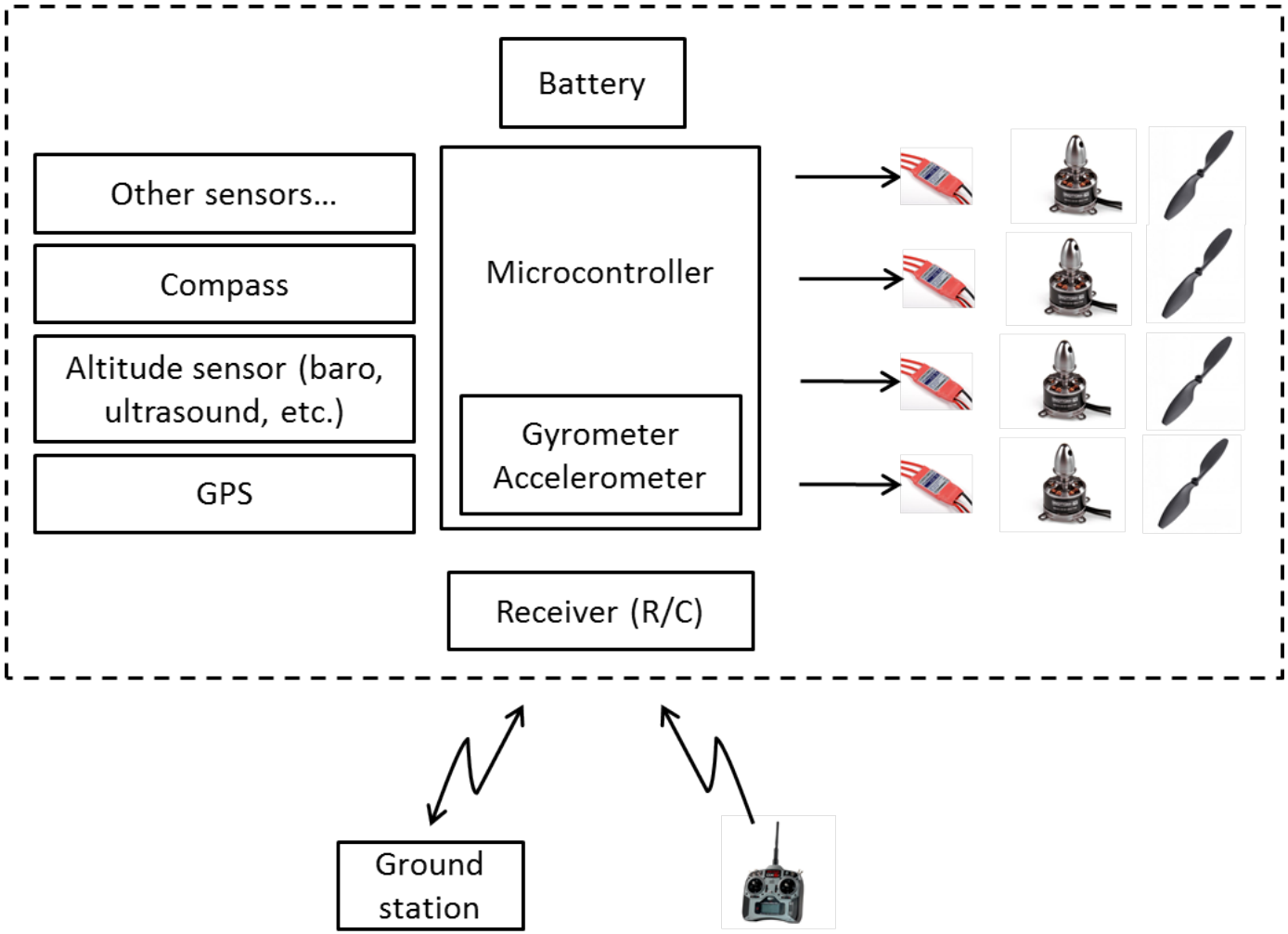
The Flight controllers is in charge of low level functions:

- stabilization of the uav,
- altitude, velocity and position control,
- main security and safety functions,
- communication with an operator or a ground station.

Many commercial or opensource hardware/software systems (Ardupilot, Paparazzi, Pixhawk, etc.)

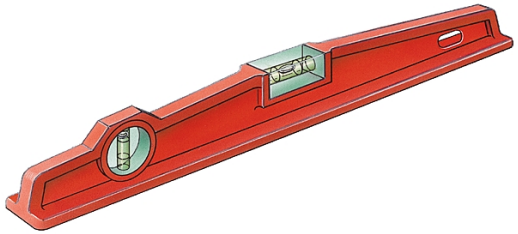
Higher level functions (navigation, obstacle avoidance, decision, payload control, etc.) can be done by the same controller or a dedicated one.

Minimal design of a quadrotor

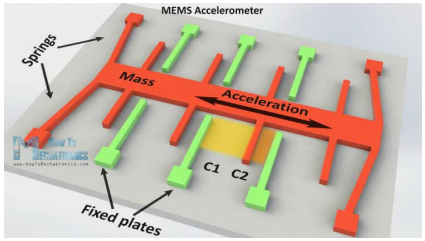


Sensors

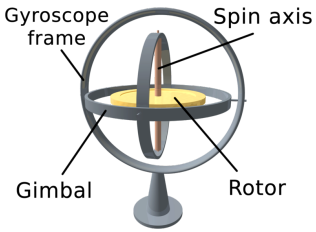
Attitude sensor



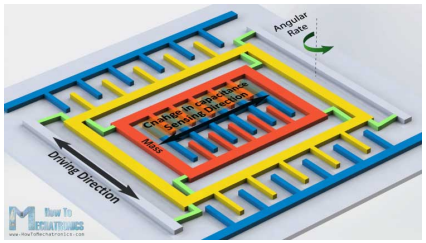
A basic tilt sensor



MEMs accelerometer

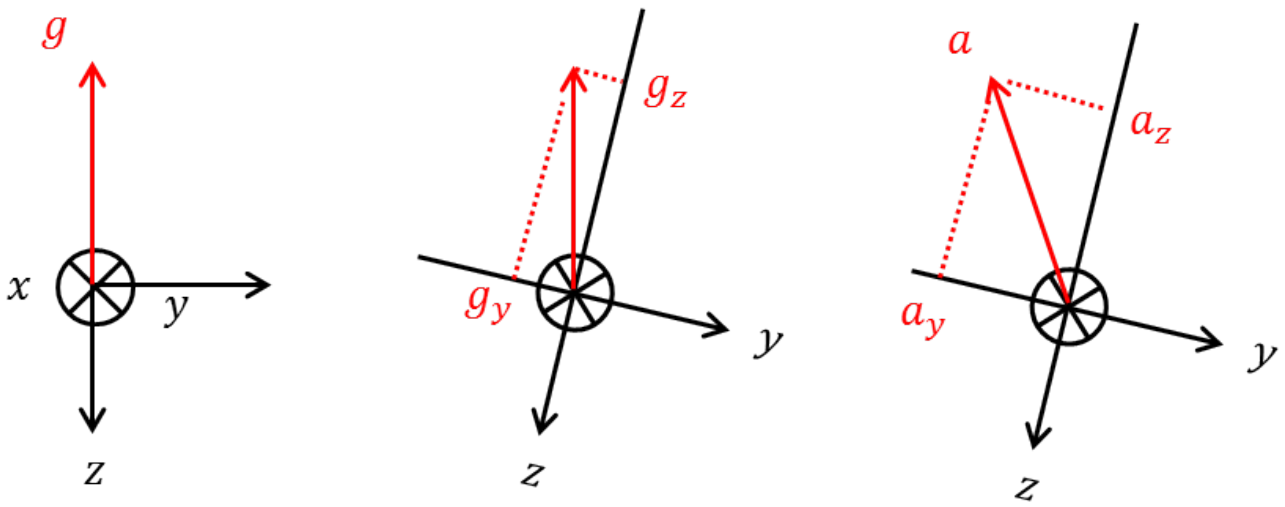


The well known Gyroscope



MEMs gyrometer

Combining accelerometer and gyrometer reading allows to estimate precisely the attitude.



The accelerometer reads the full acceleration: gravity (up) and inertial acceleration due to navigation.
When the accelerometer is just tilted with an angle θ_x it reads:

$$acc = \begin{bmatrix} 0 & -g\sin\theta_x & -g\cos\theta_x \end{bmatrix}^T \tag{1}$$

If the sensor is just tilted the accelerometer allows to estimate the angle:

$$\theta_{acc} = \arctan(\frac{acc(2)}{acc(3)}) \tag{2}$$

Main drawback: it is ok only if there is no inertial acceleration

The gyrometer simply reads the rotation rate along its axis. The integration allows to estimate the angle:

$$\begin{aligned} gyr_x &= \dot{\theta}_x \\ \theta_{gyr} &= \int_0^t gyr_x(\tau) d\tau \end{aligned} \tag{3}$$

Main drawback: a small error in the sensor reading results in a drift after integration:

$$\begin{aligned} gyr_x &= \dot{\theta}_x + \epsilon \\ \theta_{gyr} &= \int_0^t gyr_x(\tau) d\tau + \epsilon\, t \end{aligned}$$

Historical solution was to add filtered signals:

- Low pass component of (2) is ok if there is no constant acceleration (such as constant coordinate turn)
- (3) is ok if the mean value is removed by high pass filter

$\rightarrow$ The correct estimate is $\theta = LP(\theta_{acc}) + HP(\theta_{gyr})$

The result is a pretty nice estimate but:

- Not so easy to generalize in 3D
- Beware of gimbal lock!
- And if we want to use another sensor input? (Compass)

"Modern" solutions are based on Kalman filer (1960, Apollo program):

- Elegant and mathematically proven solution
- Easy to understand and program
- Very versatile (easy to add sensor readings)

Kalman filter is one practical solution for "data fusion" (mix different sensors for estimation)

The state to be estimated is $X = \begin{bmatrix} \hat{\theta}_x & \dot{\hat{\theta}}_x \end{bmatrix}^T$. Without sensor reading one can predict the next state:

$$\begin{aligned} X_1(t + \Delta T) &= X_1(t) + \Delta T \, X_2(t) \\ X_2(t + \Delta T) &= X_2(t) \end{aligned} \tag{4}$$

Given this estimate one can predict the expected sensor reading $\hat{Y}$ and compare to the actual reading Y:

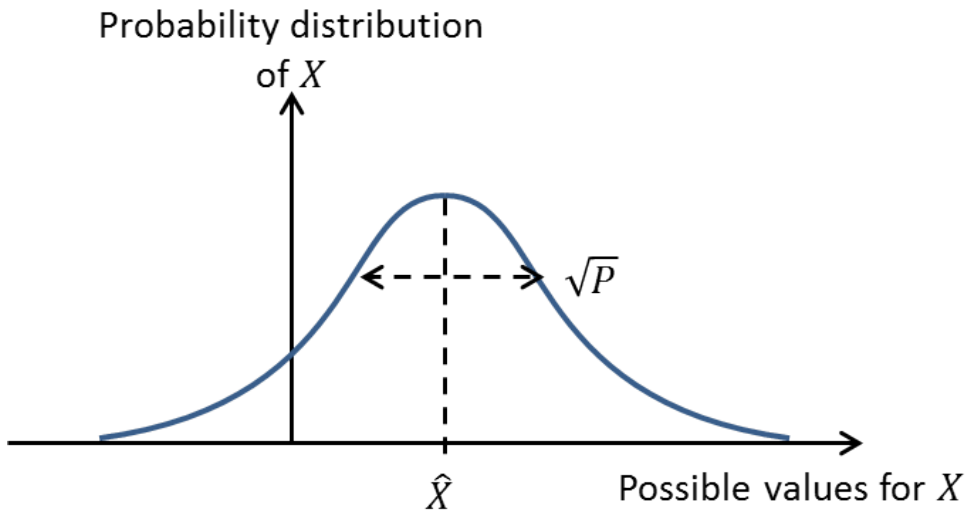
$$\hat{Y} = \begin{bmatrix} -g \sin\theta_x & -g \cos\theta_x & \dot{\theta}_x \end{bmatrix}^T = \begin{bmatrix} -g \sin X_1 & -g \cos X_1 & X_2 \end{bmatrix}^T$$

$$Y = \begin{bmatrix} acc_y & acc_z & gyr_x \end{bmatrix}^T$$

The Kalman gain (not explained here how it is obtained, a few lines of code anyway) allows to update the estimate:

$$X = X + K\,(Y - \hat{Y}) \tag{5}$$

The Kalman filter algorithm not only estimate the state X but also the covariance P: "how much" confident with the estimate we are.



The Kalman filter is based on a probabilistic Gaussian assumption: the state is most probably $\hat{X}$ with a Gaussian distribution of covariance P

The 3D case is an extension of the 2D case explained above with some particularities:

- It is natural to estimate the full attitude state as $X = \begin{bmatrix} \theta & \phi & \psi \end{bmatrix}^T$ but the estimation update (4) may result in gimbal lock for angles near $\pi/2$. The elegant (and only valid solution) is to use a quaternion, i.e. a (4×1) vector, for the attitude estimation.
- The accelerometer/gyrometer is only valid for the estimation of roll and pitch. For the full attitude estimation (including yaw) another sensor is mandatory, usually a magnetometer.

Facts about the IMU (accelerometer, gyrometer and often magnetometer):

- This kind of IMU is called "straptdown IMU" because sensors are attached to the body frame (different from the old fashioned gyrometer with moving parts)
- Many specialized microcontroller dedicated to UAV control have native IMU
- Price range: from a few € (thank you Apple) to several 10k€ and more (military)
- Only high performance IMU (aircraft navigation, missiles, submarines, etc...) can be used for velocity and (position (by integration of acceleration).

Altitude can be estimated with:

- Barometer
- Ultrasound
- Laser rangefinder
- GPS

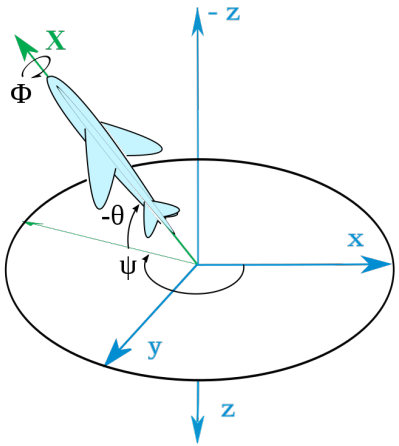
Vision can be used for control purpose:

- Vertically mounted camera and optical flow allows to estimate lateral velocity
- Wide angle camera for relative localization and navigation (SLAM)

Table of Contents

Frame references

The NED convention



https://commons.wikimedia.org/w/index.php?curid=15243475

Order of rotations (see above) : Ψ (yaw) along z, θ (pitch) along y and Φ (roll) along x.
The fixed (inertial frame) frame is attached to the earth, generally at ground level. The moving frame (body frame) is attached to the UAV.

The matrix allowing to calculate the coordinate in inertial R_I frame, given the coordinate in body R_B frame:

$$\begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix} = Z_1 Y_2 X_3 \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \tag{6}$$

With:

$$Z_1 Y_2 X_3 = R = \begin{bmatrix} c_1 c_2 & c_1 s_2 s_3 - c_3 s_1 & s_1 s_3 + c_1 c_3 s_2 \\ c_2 s_1 & c_1 c_3 + s_1 s_2 s_3 & c_3 s_1 s_2 - c_1 s_3 \\ -s_2 & c_2 s_3 & c_2 c_3 \end{bmatrix} \tag{7}$$

Rotation speed measured in the body frame (generally obtained via gyrometer) is denoted $\begin{bmatrix} p & q & r \end{bmatrix}^T$. The resulting rotation speed in inertial frame is $\begin{bmatrix} \dot{\Phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$.

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi\tan\theta & \cos\Phi\tan\theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi/\cos\theta & \cos\Phi/\cos\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

For small angles $\begin{bmatrix} p & q & r \end{bmatrix}^T \approx \begin{bmatrix} \dot{\Phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$

Kinematic Equations

The kinematic equations of the system are:

$$\dot{X} = V$$
$$m\dot{V} = m\vec{g} + R\,\Sigma_F \quad (8)$$
$$\dot{R} = R\,\omega_\times$$
$$I\,\dot{\omega} = -\omega \times I\omega + \Sigma_C \quad (9)$$

$$\omega_\times = \begin{bmatrix} 0 & p & -q \\ -p & 0 & r \\ q & -r & 0 \end{bmatrix}$$

- X is the position of the Center of Gravity (CoG) in the intertial frame
- V is the velocity
- m is the total mass of the quadrotor
- $\vec{g}$ is the gravity vector
- $R = Z_1 Y_2 X_3$ is the rotation matrix
- Σ_F is the total sum of forces in body frame (propeller thrust, aerodynamic drag, etc.)
- $\omega_\times$ is the rotation velocity matrix
- Σ_C is the total sum of torques in body frame (propeller torque, aerodynamic effect, etc.)

$$P = m\overrightarrow{g}$$

Forces and Torques

Propellers thrust and torque

Thrust:

$$F = C_F \, \rho \, D^4 \, \omega^2$$

Torque:

$$C = C_C \, \rho \, D^5 \, \omega^2$$

Where:

- F and C are the force and torque generated by the propeller
- ρ is the air density
- D is the surface of the propeller
- ω is the rotation speed of the propeller
- C_T and C_F are the thrust and torque coefficient, depending mostly on blade design

Forces and Torques

Propeller efficiency

As a secondary effect the thrust and torque efficiency depends also on relative speed of the quadrotor relatively to the air.
Vertical ascent $\rightarrow$ less efficiency
Vertical descent $\rightarrow$ more efficiency $\rightarrow$ unstability in attitude control

Forces and Torques

Rotor flapping

"Rotor flapping" is the perturbating torque caused by differential speed on forward and backward blade due to horizontal speed

Aerodynamic drag

$$F_x = -C_x \, S_x \, \frac{1}{2} \, \rho \, V_x^2$$

Where:

- F_x (or F_y or F_z) is the aerodynamic force
- S_x is the reference surface projected along x axis (depends on drone attitude)
- V_x (or V_y or V_z) is the velocity along x axis (or y or z)

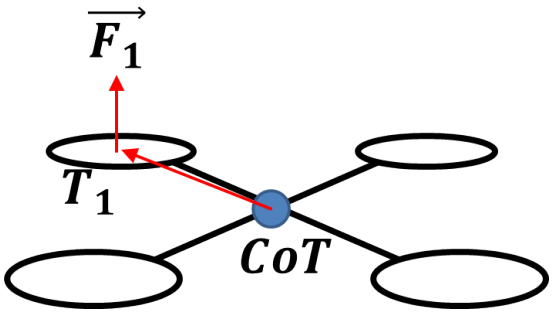
Torque

Coriolis effect

$$C_{coriolis} = -\omega \times I\omega$$

Torque

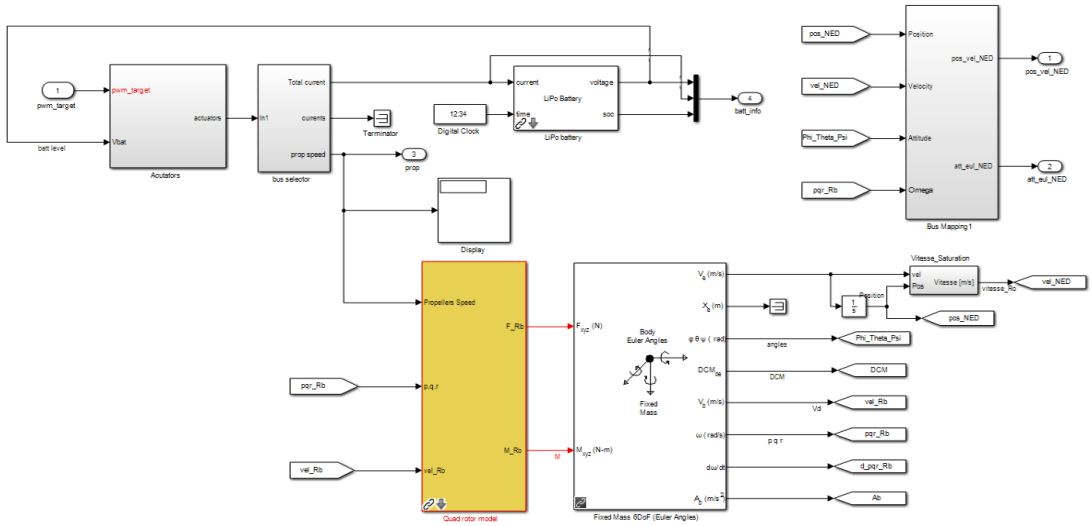
Torque due to propeller forces



$$C_p = \sum_{n=1}^{4} F_n \times \overrightarrow{CoT.T_n}$$

- C_p is the total torque due to the propellers forces
- T_n is the center of propeller n
- CoT is the centroid of $T_{1:4}$ (not necessarily coincident with the Center of Gravity CoG)

Full model



Screenshot of the ISAE/DCAS drone simulator

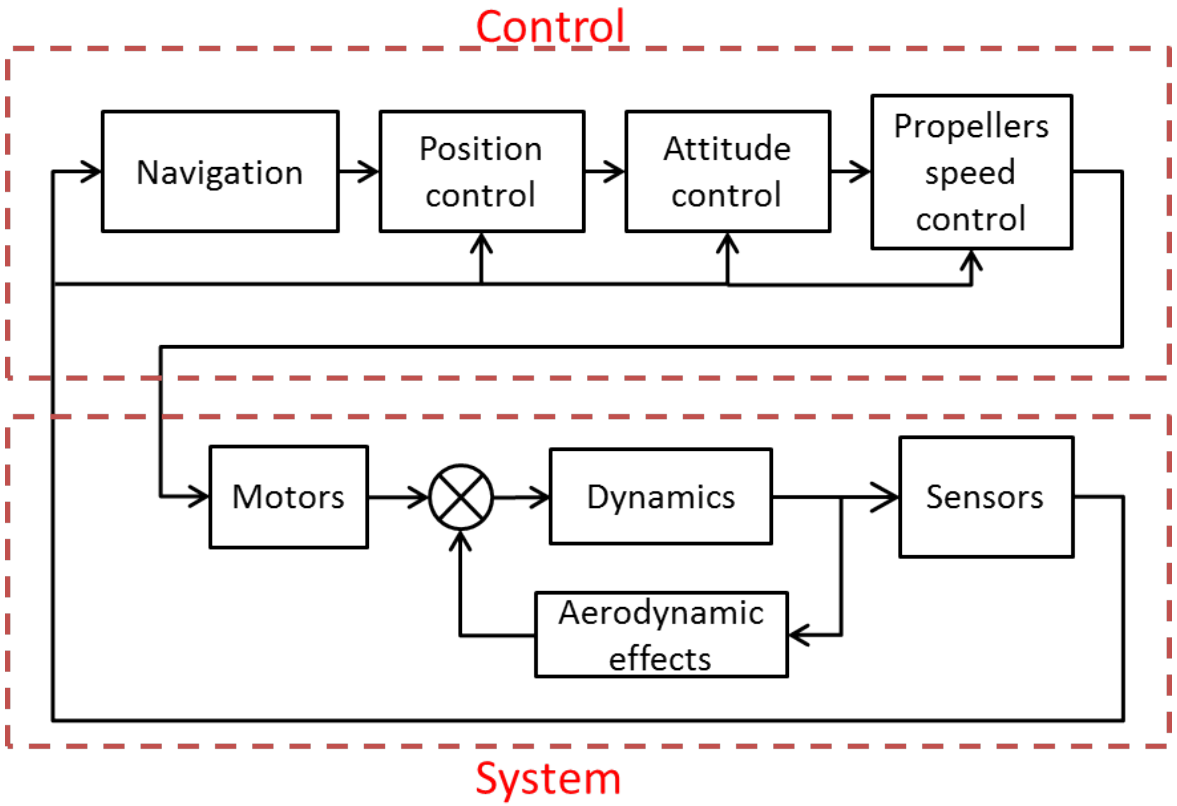
As seen in previous slides the full model will be quite complicated: many effects, reference frame transformation, non linear effects, etc.
A full simulation model may be usefull for:

- design: compare geometries, technological choice, etc.
- validate control, navigation and guidance algorithm in simulation

Table of Contents

Control Architecture

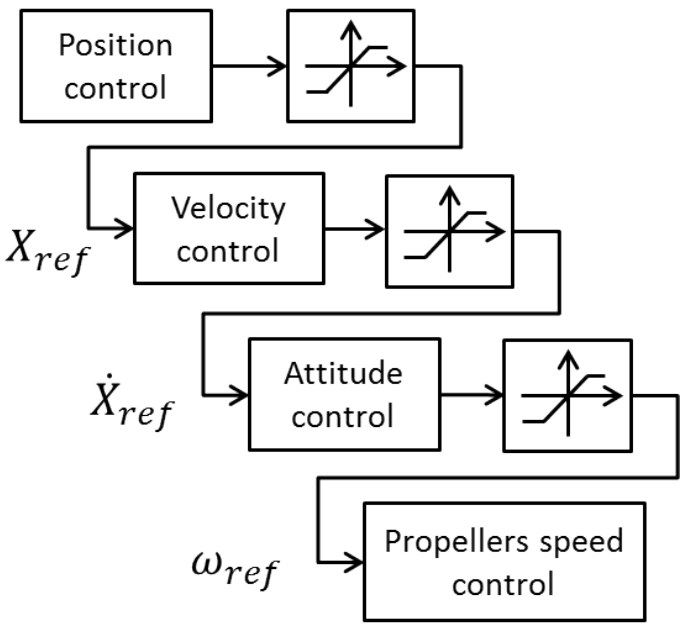
Cascade Loops principle



The embedded controller is in charge of the the low level control of the uav: stability, velocity and position. Most architectures are based on cascade loops.

Control Architecture

Cascade Loops Advanteges



Thanks to the cascade control architecture adding protections is easy:

- Maximum thrust
- Maximum attitude angles
- Maximum velocities
- Etc.

Each inner loop is significantly faster than the upper loop:

- Motor current control: 100ms
- Propeller speed control: 250ms
- Attitude control: 500ms
- Speed control: 3s
- Position: 3s

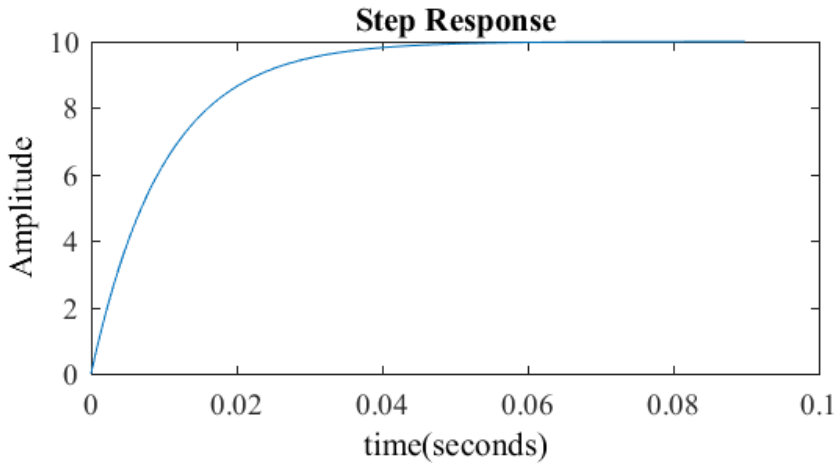
Control basics: Example of propeller speed control

Model in time domain

We assume that the propeller speed ω only depends on the input voltage u with a transfer function $H(s)$:

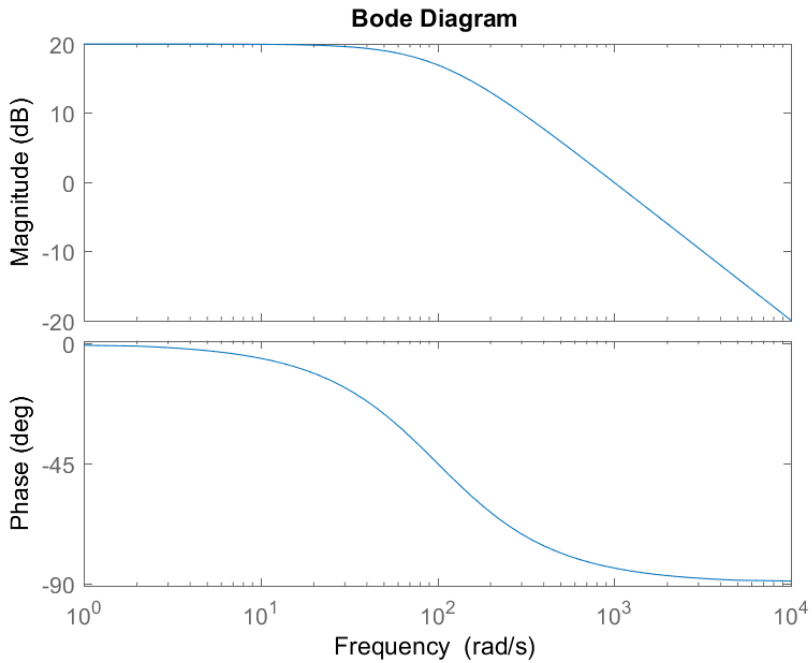
$$H(s) = \frac{\omega}{u} = \frac{A}{1 + \tau s}$$

This equation means that for an voltage input step of $1V$ it will take τ seconds (here $0.01s$) to reach a constant velocity $A\,rad/s$ (here $10\,rad/s$):



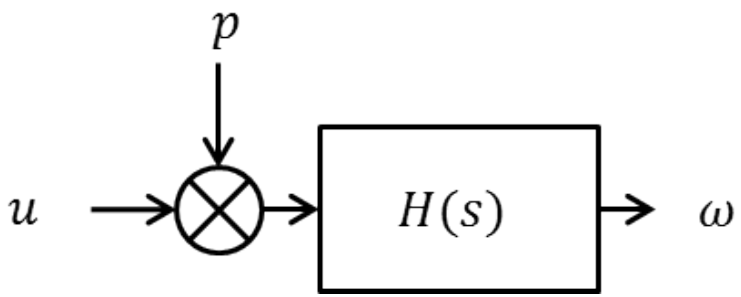
Control basics: Example of propeller speed control

Model in frequency domain



The same system can also be described in frequency domain. The Bode diagram shows how much amplified is the signal and its phase shifted for every frequency.
The Bandwidth is $BW = 1/\tau\ rad/s$ (here 100 rad/s). At a frequency higher than BW the amplification becomes negligible.

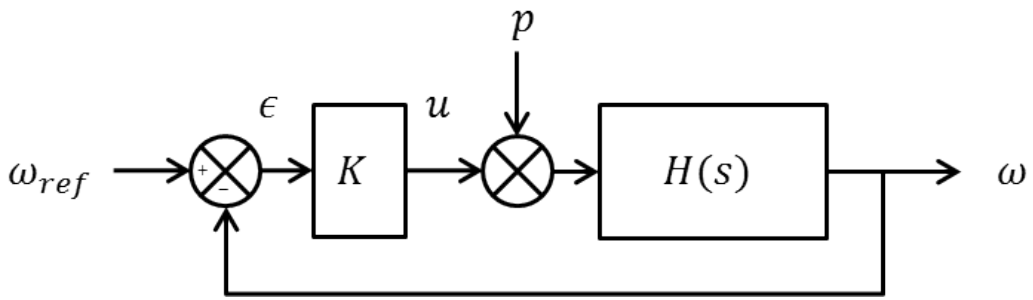
A more accurate model will take into account perturbations:



$$\omega = H(s)(u + p)$$

Control basics: Example of propeller speed control

The proportional controller



The most simple (and widely used) controller is the "proportional controller". Control input is simply proportional to the difference between reference and output.

$$u = K\,\epsilon = K\,(\omega_{ref} - \omega)$$

The Closed Loop transfer function becomes:

$$H_{CL} = \frac{\omega}{\omega_{ref}} = \frac{A'}{1 + \tau' s} \tag{10}$$

With:

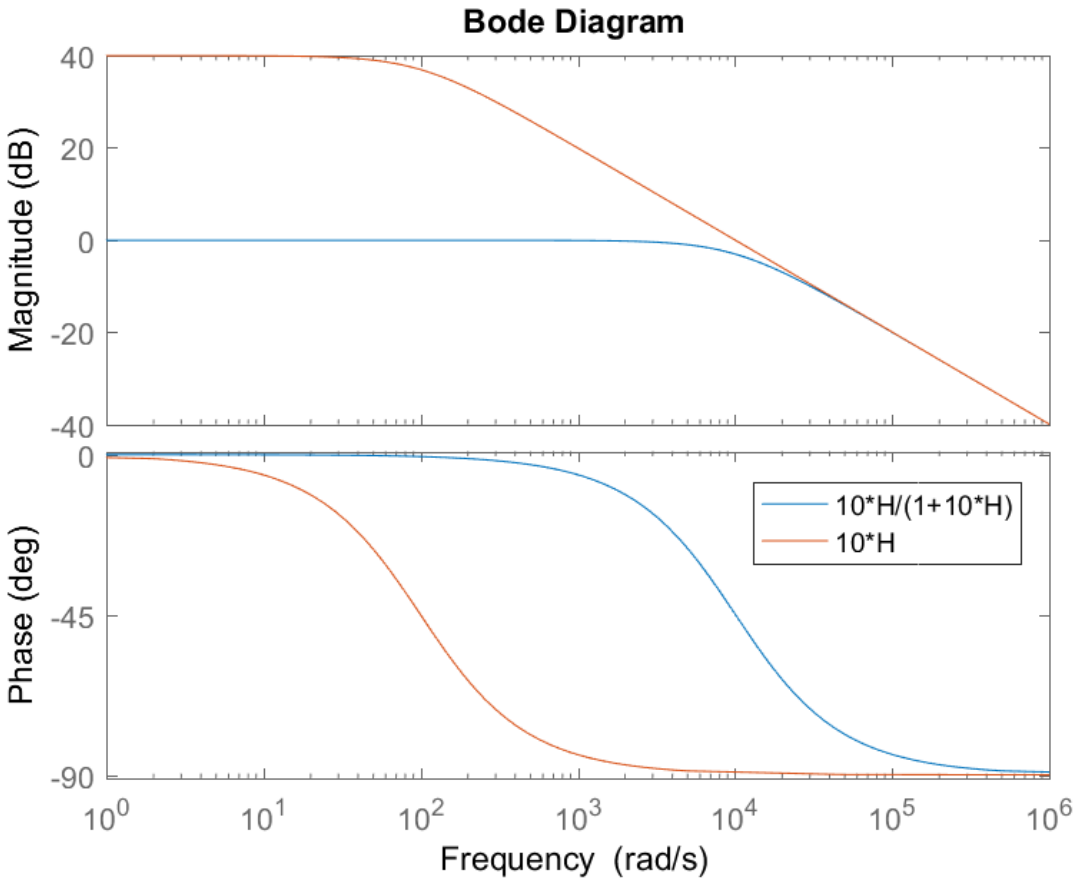
- $A' = \frac{AK}{1+AK}$: when K is big A' comes close to 1
- $\tau' = \frac{\tau}{1+AK}$: when K is big the time constant is reduced

Conclusion, increasing K means:

- Increase precision
- Reduce time response
- Reduce effect of perturbations

Control basics: Example of propeller speed control

Frequency domain interpretation of the Proportional Controller



The Bandwidth as been increased, now $10^4 rad/s$

Control basics: Example of propeller speed control

Limitations of the Proportional controller

K cannot be indefinitely increased because:

- The amplification of ϵ can lead to control (u) saturation
- At high frequency neglected dynamics must be taken into account, leading to instability

Control basics: Example of propeller speed control

Sensor

The controller of the propeller rotation speed needs a sensor:

- dedicated tachometer and feedbcack signal,
- or analyze of the back-EMF (Counter-electromotive force) by the ESC

Advanced control takes into account non-linearities (climb/descent, quadratic term)

Control Architecture

Decoupled control principle for attitude control

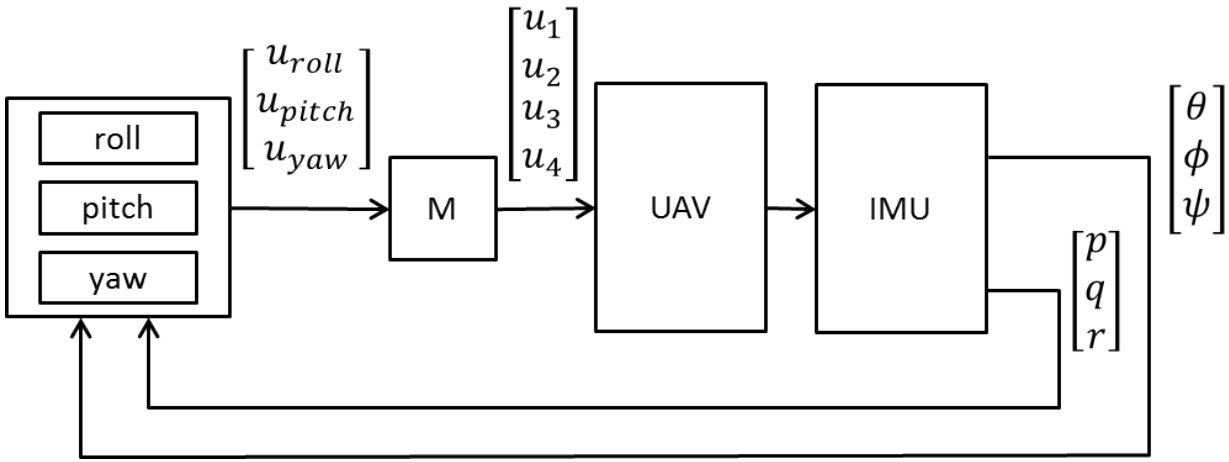
As seen in slides 9 to 11 the design of the quadrotor is made for easy decoupling. Each motor will be controlled threw a decoupling matrix:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 \end{bmatrix} \times \begin{bmatrix} u_{vert} \\ u_{roll} \\ u_{pitch} \\ u_{yaw} \end{bmatrix} = M \times \begin{bmatrix} u_{vert} \\ u_{roll} \\ u_{pitch} \\ u_{yaw} \end{bmatrix}$$

The controller generates $\begin{bmatrix} u_{vert} & u_{roll} & u_{pitch} & u_{yaw} \end{bmatrix}^T$ according to altitude, roll, pitch and yaw measurement and requirement and then multiplies by M for motors inputs.

Control Architecture

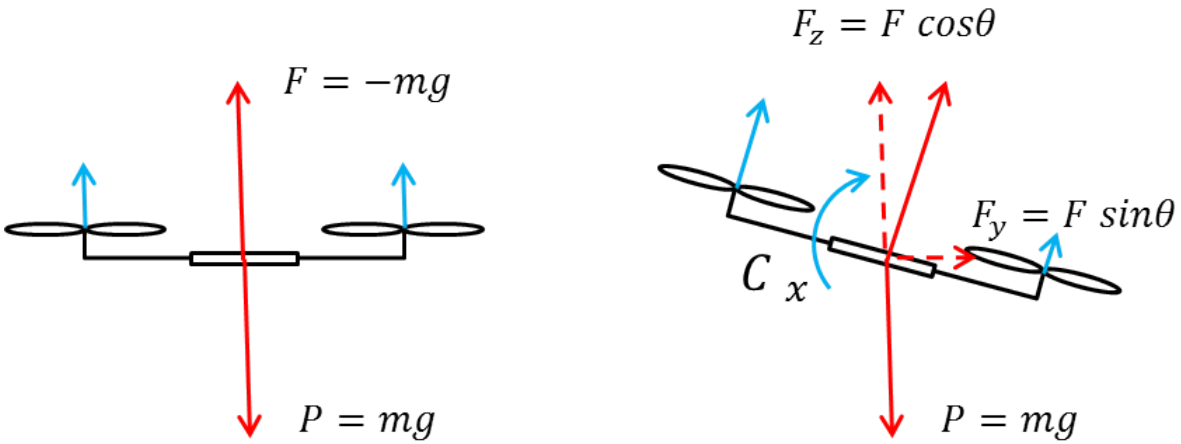
Attitude control



The attitude control is fully decoupled, made of 3 independent controllers.

Attitude control

Basic understanding of Attitude control



A disequilibrium in propellers thrust results in a tilt of the quadrotor:

$$C_x = I\,\ddot{\theta}_x \tag{11}$$

The vertical force still compensates for gravity and a significant lateral forces F_y that causes lateral translation. According to Newton's law:

$$F_y = m\,\ddot{y}$$

Model (according to equation (11)):

$$\theta = \frac{1}{I}\frac{1}{s^2} C_x = \frac{A}{s^2} u$$

Where u is the mixed input for orientation control.
The proportional controller with gain K gives:

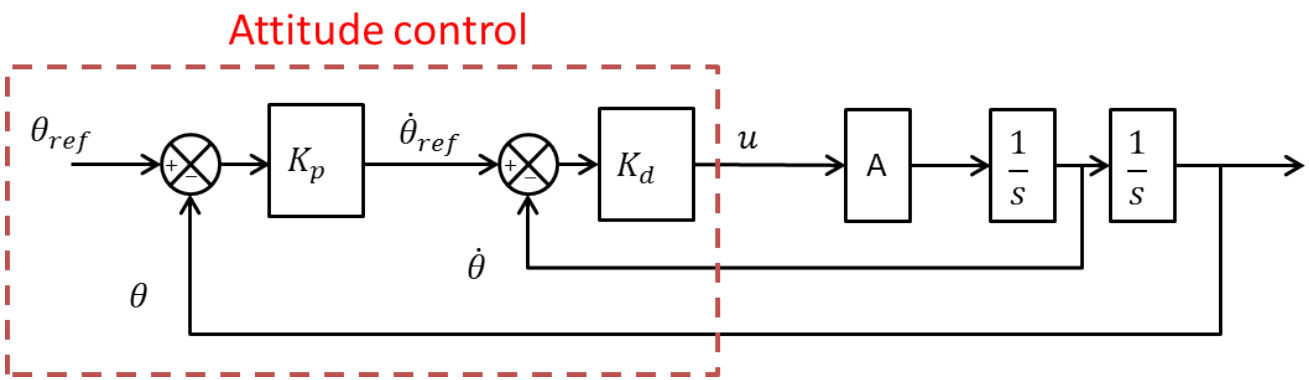
$$\frac{\theta}{\theta_{ref}} = \frac{KA}{KA + s^2}$$

The result is a purely oscillating system at frequency $\omega_0 = \sqrt{KA}$

Attitude control

The proportional derivative controller

The appropriate attitude controller is a cascade loop of rotation speed control and orientation control.



Rotation speed loop:

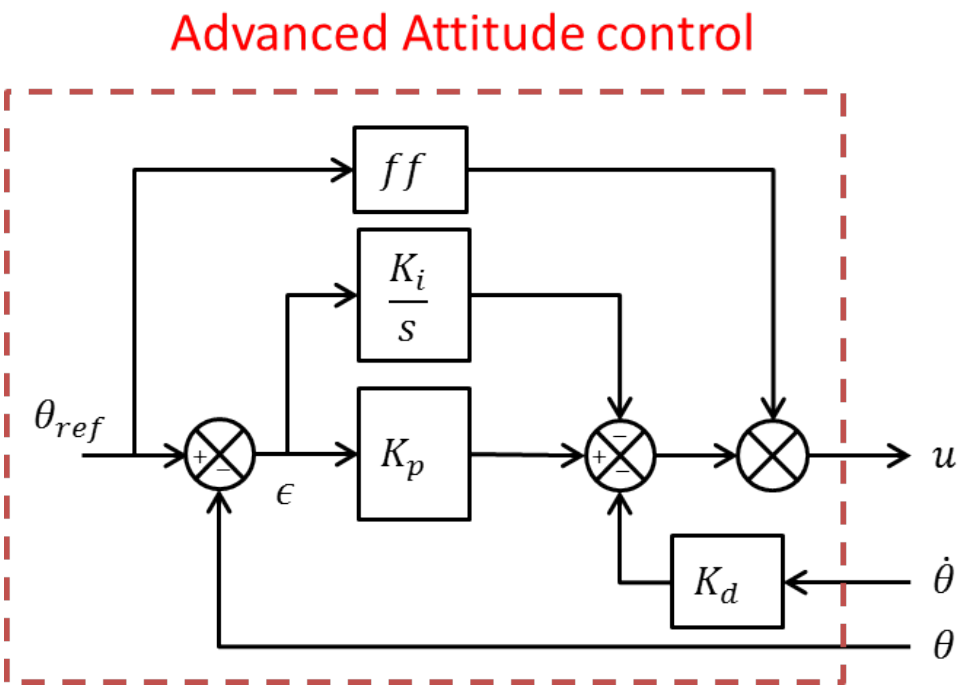
$$\frac{\dot{\theta}}{\dot{\theta}_{ref}} = \frac{k_v\, A}{k_v\, A + s}$$

Orientation loop:

$$\frac{\theta}{\theta_{ref}} = \frac{k_p\, k_v\, A}{k_p\, k_v\, A + k_v\, As + s^2} = \frac{1}{1 + 2\sigma/\omega_0 s + s^2/\omega_0^2}$$

Attitude control

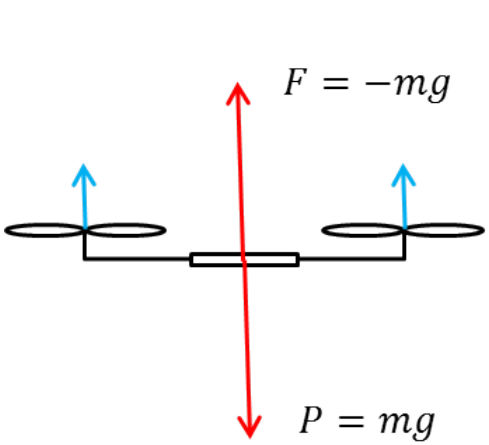
Advanced attitude control

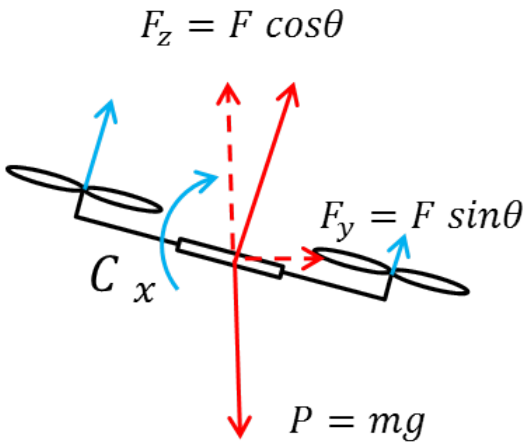


- Integral term guarantee $\epsilon(t) = 0$ in steady state
- Feedforward (ff) allows for more aggressive maneuvers (inversion of equation (9) page 36)

Lateral control

Basic understanding of Attitude control





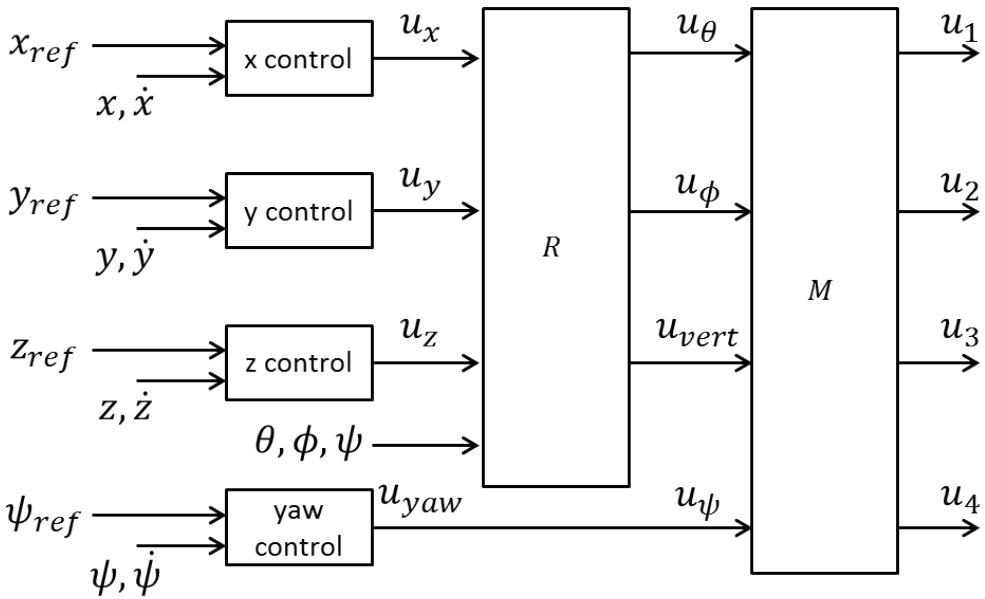
Lateral translation is caused by the horizontal component of thrust. For small angles:

$$m\,g\,\theta \approx m\,\ddot{y}$$

$$g\,\theta \approx \ddot{y}$$

The attitude angle directly controls the lateral translation.

3D control



Vertical, lateral and Yaw control:

- Altitude is controlled via total sum of thrust u_{vert}
- Lateral position $\begin{bmatrix} x & y \end{bmatrix}$ is controlled via attitude $\begin{bmatrix} \theta & \phi \end{bmatrix}$ and must be compensated by actual yaw angle ψ
- Yaw angle is controlled via differential sum of thrust u_{yaw}

Control Challenges

Non linearities

Classical controllers are based on simplified linear models. Advanced non linear control will deal with:

- Saturations of propeller speed
- High angles
- Non symmetry of proppellers efficiency
- Non linear aerodynamic effects

Control Challenges

More robust controller

Classical controllers are based on a specific configuration:

- Robust control will guarantee performance for varying configurations (mass, velocity, etc.)
- Robust control will allow to specify multi-objective performance requirements (perturbation rejection, time response, stability, etc.)
- Adaptive control will detect system's change (ex: mass) and adapt the controller consequently

Control Challenges

Cooperation and contact

Tethered UAV, load transport, cooperative load transport, control with contact

Physical contact will lead to new control challenges:

- Complex system, sometimes overactuated
- Architecture control (centralized or not)
- Transitions

Table of Contents

Conclusion

As a conclusion:

- Quadrotors and More generally Multi-rotors, are everywhere, impossible to give a list of current or potential applications
- Quadrotors are simple, cheap and easy to control (at least for simple tasks)
- Very popular in research lab for control