Quadcopters

Presented by: Andrew Depriest

What is a quadcopter?

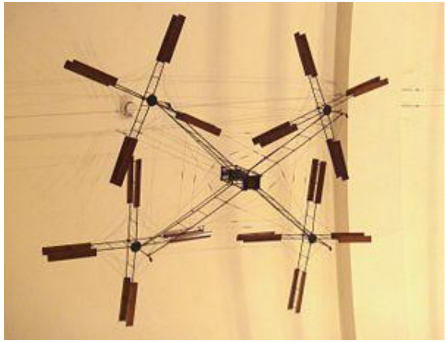
- Helicopter - uses rotors for lift and propulsion
- Quadcopter (aka quadrotor) - uses 4 rotors

Parrot AR.Drone 2.0

History

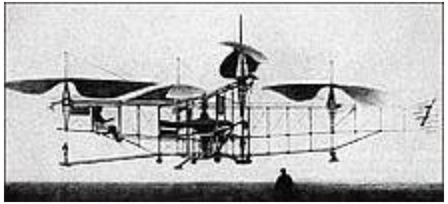
1907 - Breguet-Richet Gyroplane

- Louis Breguet and Prof. Charles Richet
- First rotary wing aircraft to lift off the ground
- Lifted only a few feet while tethered



1920 - Oehmichen No.2

- 2nd of 6 designs by Etienne Oehmichen
- 4 rotors and 8 propellers (stabilize and steer)
- Completed a 1km closed circuit flight



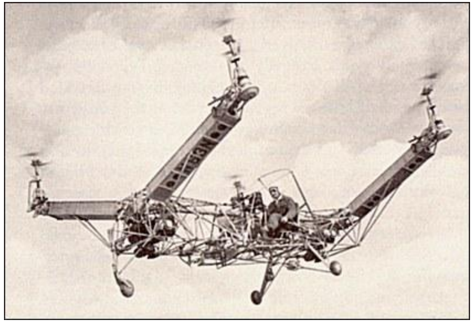
History

1922 - de Bothezat helicopter

- Dr. George de Bothezat and Ivan Jerome
- US Air Service
- Highest altitude of ~5m
- Demonstrated feasibility

1956 - Convertawings Model A Quadrotor

- Intended prototype for larger civil and military copters
- Controlled by varied rotor thrust
- First to demonstrate successful forward flight



History

1958 - Curtiss-Wright VZ-7

- Curtiss-Wright company for US Army
- Performed well during tests
- Didn’t meet Army standards

Modern

- Bell Boeing Quad TiltRotor
- Aermatica Spa's Anteos
- AeroQuad and ArduCopter
- Parrot AR.Drone
- Nixie

Uses

- Research - evaluate new ideas
  - Cheap
  - Variety of sizes
  - Maneuverability
- Military & Law Enforcement
  - Surveillance and reconnaissance
  - Search and rescue
- Commercial
  - Aerial imagery
  - Package delivery

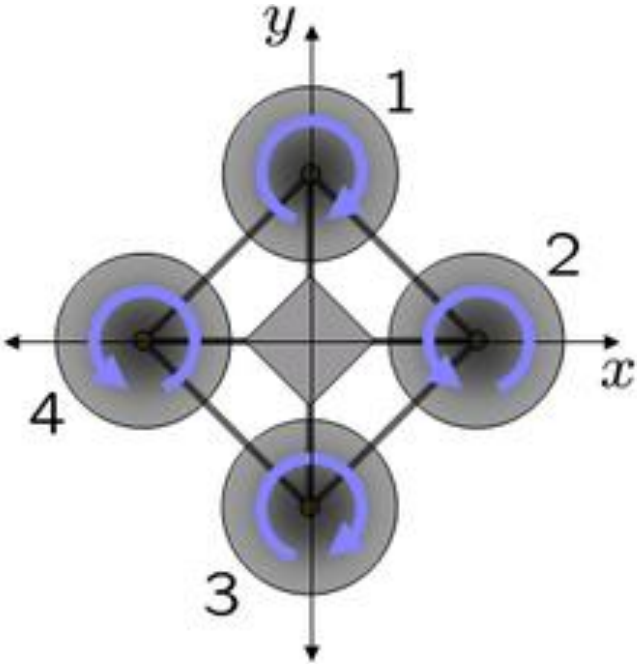
How it works

Rotors produce:

- Thrust
- Torque
- Drag force

Control input:

- Angular Velocity



Modelling and control of quadcopter

Teppo Luukkonen - Aalto University in Espoo, Finland

“Present the basics of quadcopter modelling and control as to form a basis for further research and development”

- Study the mathematical model of the quadcopter dynamics
- Develop proper methods for stabilisation and trajectory control of the quadcopter

“The challenge ... is that the quadcopter has six degrees of freedom but there are only four control inputs”

Mathematical Model

Quadcopter:

- Position
- Pitch, roll, yaw
- Pose

Body Frame:

- Linear Velocity
- Angular Velocity

Body-to-Inertial Frame:

- Rotation matrix
  - orthogonal
  - $R^{-1} = R^T$
    - Inertial-to-body

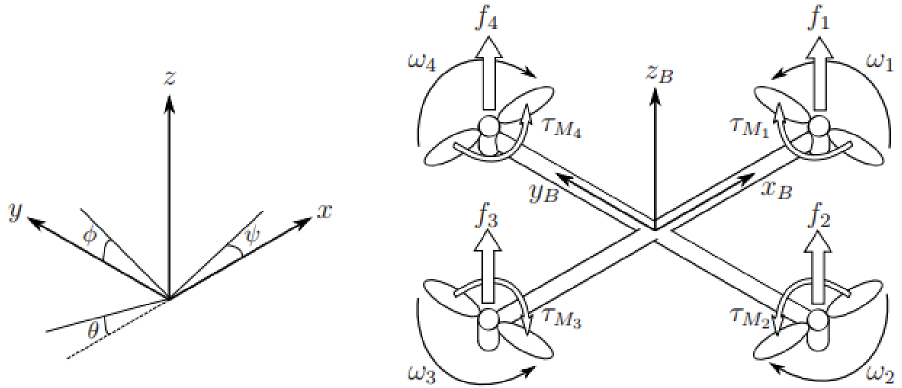


Figure 1: The inertial and body frames of a quadcopter

$$\boldsymbol{\xi} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \boldsymbol{\eta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad \boldsymbol{q} = \begin{bmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{bmatrix} \quad \boldsymbol{V}_B = \begin{bmatrix} v_{x,B} \\ v_{y,B} \\ v_{z,B} \end{bmatrix}, \quad \boldsymbol{\nu} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$$\boldsymbol{R} = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix}$$

$S_x = sin(x)$
and
$C_x = cos(x)$

Mathematical Model (cont'd)

Transformation matrices (angular vel.)

- inertial-to-body
- body-to-inertial

Symmetric structure

- Inertia matrix is diagonal

Lift force - lift constant and angular vel.

Torque - drag constant and angular vel.

- inertia moment term small, omitted

Roll = -2nd rotor, +4th rotor

Pitch = -1st rotor, +3rd rotor

Yaw = +/-(+1st, +3rd, -2nd, -4th)

$$\dot{\boldsymbol{\eta}} = \boldsymbol{W}_\eta^{-1} \boldsymbol{\nu}, \quad \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & S_\phi T_\theta & C_\phi T_\theta \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi / C_\theta & C_\phi / C_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$$T_x = tan(x)$$

$$\boldsymbol{\nu} = \boldsymbol{W}_\eta \dot{\boldsymbol{\eta}}, \quad \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \qquad I_{xx} = I_{yy}$$

$$f_i = k\,\omega_i^2, \quad \tau_{M_i} = b\,\omega_i^2 + I_M\,\dot{\omega}_i,$$

$$T = \sum_{i=1}^{4} f_i = k \sum_{i=1}^{4} \omega_i^2, \quad \boldsymbol{T}^B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}$$

$$\boldsymbol{\tau}_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l\,k\,(-\omega_2^2 + \omega_4^2) \\ l\,k\,(-\omega_1^2 + \omega_3^2) \\ \sum_{i=1}^{4} \tau_{M_i} \end{bmatrix}$$

More math (summarized)

Newton-Euler equations

- Quadcopter is assumed rigid body
- Force for accel. of mass + centrifugal force = gravity + thrust
- Body frame
  - External torque = ang. accel. + centripetal + gyroscopic forces
- In inertial frame
  - Centrifugal is nullified
  - Angular accels. calculated using transformation matrix and it’s time derivative

More math (summarized)

Euler-Lagrange equations

- Lagrangian = Translational + rotational energies - potential energy
- Euler-Lagrange equations
    - Linear and angular components independent $$\begin{bmatrix} f \\ \tau \end{bmatrix} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q}.$$
- Jacobian matrix, Coriolis term, aerodynamical effects ….

Too much math.

Model Simulation

- Used MATLAB 2010
- Initial stable state
- Params used:

Table 1: Parameter values for simulation

Parameter	Value	Unit
g	9.81	m/s^2
m	0.468	kg
l	0.225	m
k	$2.980 \cdot 10^{-6}$	
b	$1.140 \cdot 10^{-7}$	
I_M	$3.357 \cdot 10^{-5}$	kg m^2

Parameter	Value	Unit
I_{xx}	$4.856 \cdot 10^{-3}$	kg m^2
I_{yy}	$4.856 \cdot 10^{-3}$	kg m^2
I_{zz}	$8.801 \cdot 10^{-3}$	kg m^2
A_x	0.25	kg/s
A_y	0.25	kg/s
A_z	0.25	kg/s

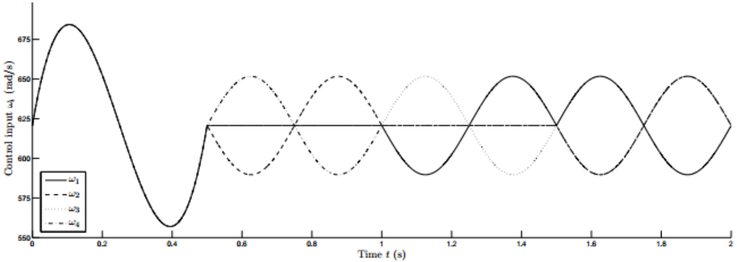


Figure 2: Control inputs ω_i

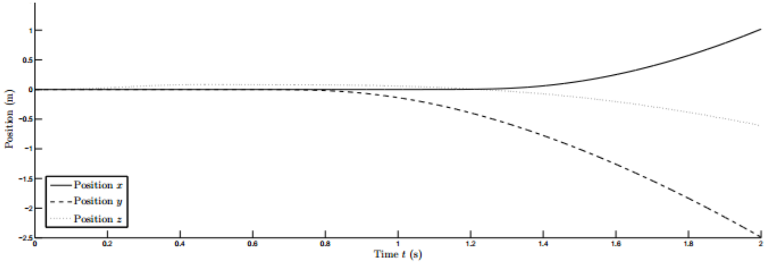


Figure 3: Positions x, y, and z

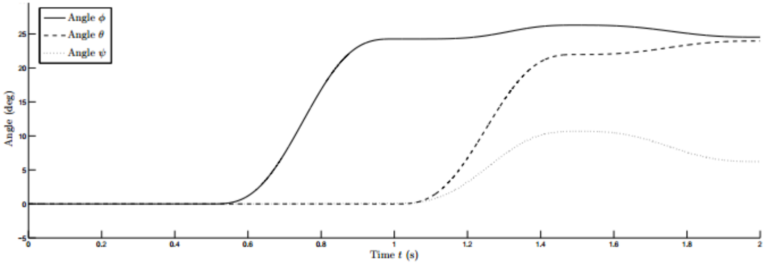


Figure 4: Angles ϕ, θ, and ψ

Stabilisation

PID controller used

- Simple structure
- Easy implementation
- General form
  - Proportional - uses diff. between desired and present positions
  - Integral - uses diff. between desired and present attitudes
  - Derivative - uses diff. between desired and present positions
- Specific form - PD controller
  - Torque calculated taking into account gravity, mass, and moment of inertia

Stabilisation Simulation

Note: The PD only stabilizes hover (altitude and attitude)
It does not consider accel. in the x and y axis
Starting z=1
Desired z=0

Table 2: Parameters of the PD controller

Parameter	Value
$K_{z,D}$	2.5
$K_{\phi,D}$	1.75
$K_{\theta,D}$	1.75
$K_{\psi,D}$	1.75

Parameter	Value
$K_{z,P}$	1.5
$K_{\phi,P}$	6
$K_{\theta,P}$	6
$K_{\psi,P}$	6

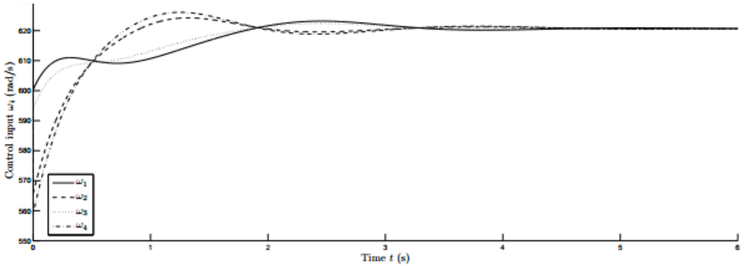


Figure 5: Control inputs ω_i

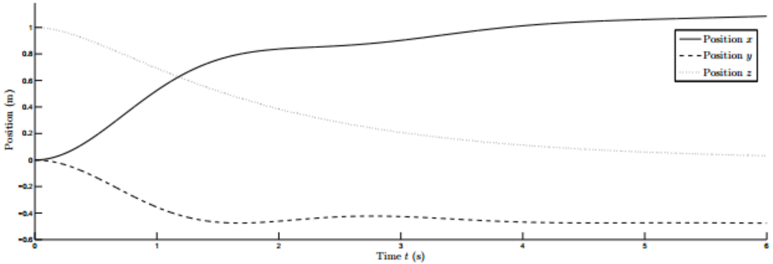


Figure 6: Positions x, y, and z

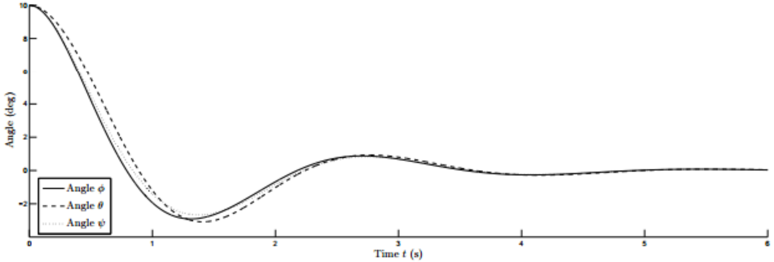


Figure 7: Angles ϕ, θ, and ψ

What’s left (more math)

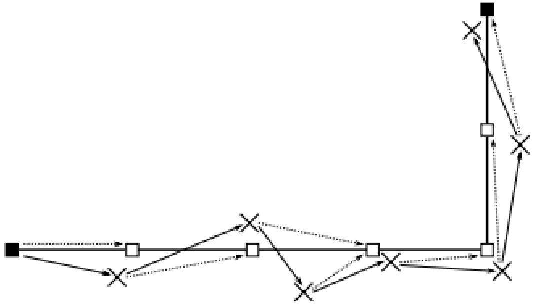


Figure 14: Example of checkpoint flight pattern with external disturbances

- Trajectory control
  - Have desired trajectory
  - Generate linear accelerations to accomplish it
  - Derive the roll, pitch, and thrust values for those
- Heuristic model for trajectory generation
  - Jerk and jounce have to be reasonable (3rd and 4th derivatives of position)
  - Symmetry on acceleration and deceleration
- Integrated PD controller
  - Take into account possible deviations in attitude

Conclusion

- Simulation proved the model to be realistic
- Simulation also proved the PD controller to be efficient in stabilising the altitude and attitude
  - x and y positions were not considered, they varied due to deviation of pitch and roll angles
- Proposed heuristic method produced good trajectories using parameters to generate jounce, and using jounce to derive position, it’s other derivatives, torque, etc
- Integrated PD operated well to take into account unmodelled disturbances like wind, but could perform poorly depending on parameters used
- These were simulations, some aerodynamics were omitted, localization was trivialized, so effects of imprecise measurements and knowledge needs to be further studied

References

- Luukkonen, Teppo. "Modelling and control of quadcopter." Independent research project in applied mathematics, Espoo (2011).
- <http://en.wikipedia.org/wiki/Quadcopter>