ALMA MATER STUDIORUM
UNIVERSITÁ DI BOLOGNA

SCHOOL OF ENGINEERING AND ARCHITECTURE
-Forlì Campus-

SECOND CYCLE MASTER'S DEGREE in INGEGNERIA AEROSPAZIALE / AEROSPACE ENGINEERING
Class LM-20

GRADUATION THESIS
in Atmospheric Flight Dynamics

Multi-rotor aircraft flying qualities assessment by means of inverse simulation

CANDIDATE:
Bernabé Matteo

SUPERVISOR:
Prof. Giulietti Fabrizio
Dr. Rossetti Gianluca

Academic Year
2016/2017

Session III

Introduction

This thesis work wants to investigate in how complex the mathematical model of a multi-rotor drone has to be, in order to simulate three different flight conditions: impulsive responses, vertical and horizontal flight. This is done by exploiting a technique called inverse simulation.
In the first chapters it is described how data were obtained and how they were used in order to characterize the frame and its motors. Also the two autopilot used to control the model has been designed to be as close as possible to the real one.
The results obtained with Simulink are then compared with the telemetry data collected during the real flight by the Pixhawk flight controller.

Contents

List of Figures

List of Tables

Chapter 1

Radio controlled models

1.1 Unmanned Aerial Vehicle

An Unmanned Aerial Vehicle (UAV), commonly known as a drone, is an aircraft without a human pilot aboard. UAVs are a component of an unmanned aircraft system (UAS); which include a UAV, a ground-based controller, and a system of communications between the two.
The flight of UAVs may operate with various degrees of autonomy: either under remote control by a human operator or autonomously by on-board computers.
Compared to manned aircraft, UAVs were originally used for missions too "dull, dirty or dangerous" for humans. While they originated mostly in military applications, their use is rapidly expanding to commercial, scientific, recreational, agricultural, and other applications, such as policing, peacekeeping, and surveillance, product deliveries, aerial photography, agriculture, smuggling, and drone racing.
Nowadays civilian UAVs vastly outnumber military UAVs and they are considered the first commercial application of "autonomous things", to be followed by autonomous cars and home robots [1].

1.2 Pixhawk flight controller

Usually drones are sold with complete avionics provided by the constructors. These autopilots are "closed" and the built-in autopilot logic, as well as its data sheets, are not of public domain which means that the operator can modify and tune only a limited amount of parameters.
This is the reason behind the choice of using a totally different flight controller for the experiments.
Pixhawk autopilot is an open-source autopilot system oriented toward inexpensive autonomous aircraft. Low cost and availability enable hobbyist use in small remotely piloted aircraft. The project started in 2009 and is being further developed and used at Computer Vision and Geometry Lab of ETH Zurich (Swiss Federal Institute of Technology) and supported by the Autonomous Systems Lab and the Automatic Control Laboratory [2].
Due to its open-source characteristic a great quantity of additional modules have been designed which greatly increase the capabilities of the system. Moreover thanks to the possibility of retrieving all the necessaries flight data and the possibility to design from the scratch a brand new flight control logic, Pixhawk represent the best solution to study and analyze the flight qualities through inverse simulation.

Chapter 2

Drone DJI S800 PRO

2.1 Specifications

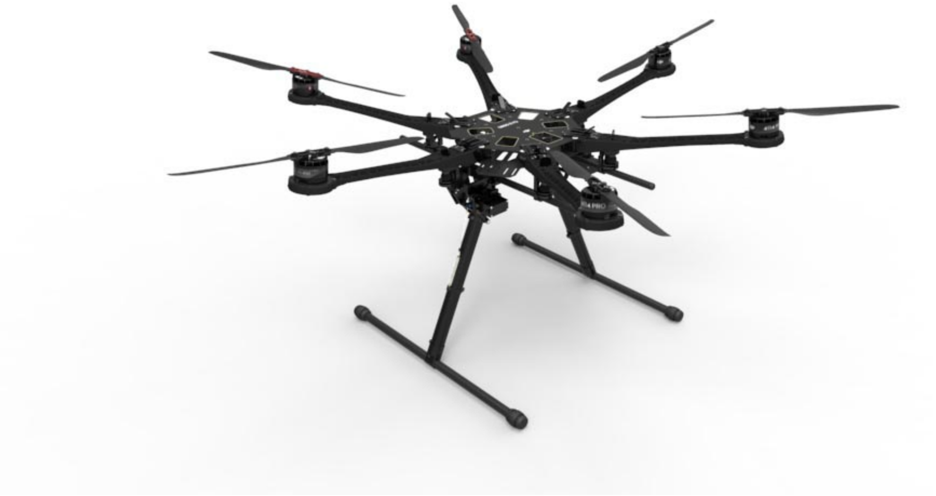

Figure 2.1: DJI S800 PRO (from DJI website (2018) [3])

The S800 EVO is a new generation of hex-rotor designed for professional aerial photography. Its structure is composed of carbon fiber, aluminum and high quality plastic. The diagonal wheelbase is approximately 800mm wide and 320mm tall and the whole system weights about 5807g.

2.2 Propulsive system

The motor is a DJI 4114 PRO with a built-in centrifugal fan with optimized heat dissipation channels. Its characteristics are:

- Total size = $46 \times 35mm$
- Stator size = $41 \times 14mm$
- KV = $400rpm/V$
- Max power = $500W$
- Weight (with cooling fan) = $158g$

On the top of each motor rotor case is mounted a DJI foldable propeller 1552 with:

- Material = Engineering plastic
- Size = $15 \times 5.2in$
- Weight = $13g$
- $c_{3/4} = 0.022m$

The Electronic Speed Controllers (ESC) which converts the PWM signal coming from the autopilot are directly mounted below the motors. Its specifications are:

- Current = $40A$ OPTO
- Voltage = $6S$ LiPo
- Signal frequency = $30Hz \sim 450Hz$
- Drive PWM Frequency = $8kHz$
- Weight (with radiators) = $35g$

The battery is a TATTU 16000 LiPo battery pack and its characteristics are:

- Capacity = $16000mAh$
- Voltage = $22.2V$
- Discharge Rate = 15C
- Weight = $1938g$
- Dimensions = $193 \times 77 \times 66mm$

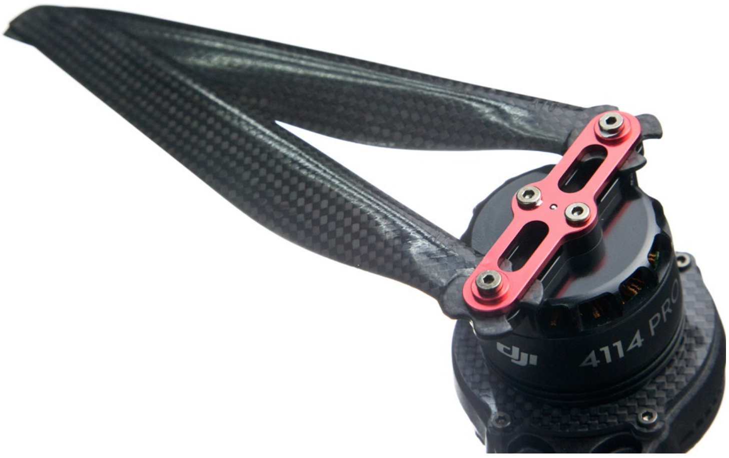

Figure 2.2: detail on motor and propeller (from DJI website (2018) [3])

2.3 Avionics

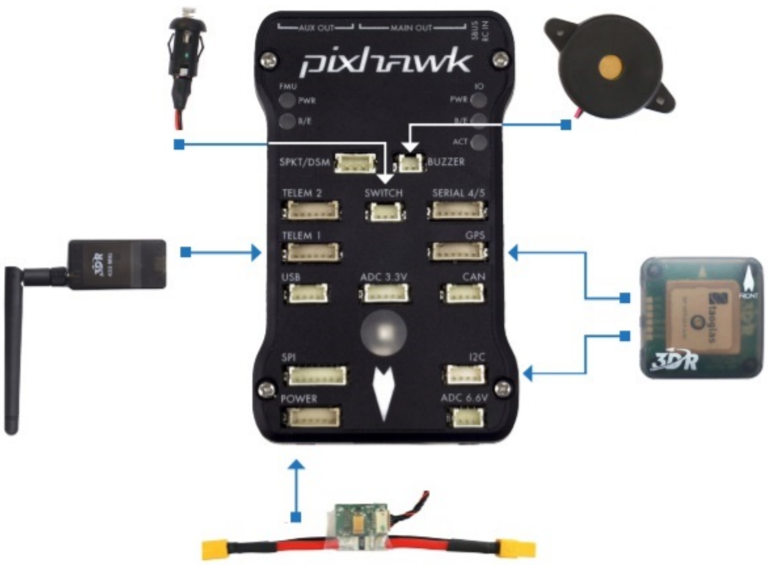

Figure 2.3: Pixhawk wiring (from Ardupilot website (2018) [4])

The HKPilot32 flight controller, based on the Pixhawk open hardware design, is an high-performance autopilot-on-module suitable for fixed wing, multi-rotors, helicopters, cars, boats and any other robotic platform that can move. It is targeted towards high-end research, amateur and industry needs [5].

In order to control the vehicle a full set of gyroscopes, accelerometers, a magnetometer and a barometer are already embedded in it.

There are 6 devices plugged to the flight controller:

- Buzzer: together with the LEDs, it is used by the pilot to understand in which modes the drone is into while on ground
- Safety switch: used to enable the connection with the motors and start the system
- GPS: gives position information to the flight controller about the Nord East Down (NED) reference system
- Power module: supplies the flight controller and the motors with power from the battery and also sends information about the analog current and voltage supplied via the module
- Radio control: used to manually control the vehicle from a ground station
- Telemetry radio: used to communicate and control a vehicle in flight from a ground station (for example, you can direct the UAV to a particular position, or upload a new mission)

2.4 Digital signal modulation protocols

The hardware installed on the drone communicates and shares data using digital signals. Having and insight on how they work is useful in order to understand their capacities and limits.

2.4.1 Pulse-Width Modulation

The communication between the flight controller and motors takes place through a Pulse-width modulation (PWM) signal.
PWM is a modulation technique used to encode a message into a pulsing signal. Although this technique can be used to encode information for transmission, its main use is to allow the control of the power supplied to electrical devices, especially to inertial loads such as motors.
The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast rate. The longer the switch is on compared to the off periods, the higher the total power supplied to the load. The PWM switching frequency has to be much higher than what would affect the load (the device that uses the power) in order to have a resultant waveform as smooth as possible. In case of small electrical motors the switching velocity is between few kilohertz to tens of kilohertz.
The second important term in order to control a device is the duty cycle. It describes the proportion of "on" respect to the time of the period; a low duty cycle corresponds to low power, because the power is "off" for most of the time. Duty cycle is expressed in percent, 100% being fully on.
The main advantage of PWM is that power loss in the switching devices is very low. When a switch is off there is practically no current, and when it is on and power is being transferred to the load, there is almost no voltage drop across the switch. Power loss, being the product of voltage and current, is thus in both cases close to zero.
PWM also works well with digital controls, which, because of their on/off nature, can easily set the needed duty cycle [7].

Working principle

Pulse-width modulation uses a rectangular pulse wave whose pulse width is modulated resulting in the variation of the average value of the waveform. If we consider a pulse waveform $f(t)$, with period T, low value y_{min}, an high value y_{max} and a duty cycle D, the average value of the waveform is given by:

$$\bar{y} = \frac{1}{T}\int_0^T f(t)dt \tag{2.1}$$

As $f(t)$ is a pulse wave, its value is y_{max} for $0 < t < DT$ and y_{min} for $DT < t < T$. The above expression then becomes:

$$\begin{aligned} \bar{y} &= \frac{1}{T}\left(\int_0^{DT} y_{max}dt + \int_{DT}^{T} y_{min}dt\right) \\ &= \frac{1}{T}\left(D \cdot T \cdot y_{max} + T(1-D)y_{min}\right) \\ &= D \cdot y_{max} + (1-D)y_{min} \end{aligned} \tag{2.2}$$

This latter expression can be fairly simplified in many cases where $y_{min} = 0$ as $\bar{y} = D \cdot y_{max}$. From this, it is obvious that the average value of the signal is directly dependent on the duty cycle D [7].

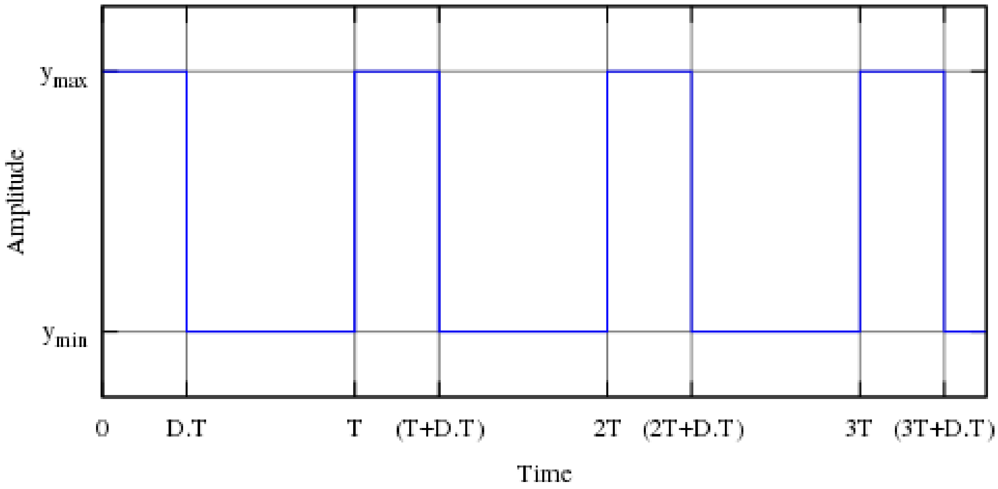

Figure 2.4: a pulse wave, showing the definitions of y_{min}, y_{max}, D and T (from Wikipedia website (2018) [7])

2.4.2 Pulse-Position Modulation

Radio controller and the on board receiver instead communicates with PPM signals.

Pulse-position modulation (PPM) is a form of signal modulation in which M message bits are encoded by transmitting a single pulse in one of 2^M possible required time shifts. This is repeated every T seconds, such that the transmitted bit rate is M/T bits per second [8].

Differently from a PWM signal where the duty cycle governs the power given to the device (while the frequency is mainly relaed to the device's inertia), the PPM signal has to be interpreted by the microcontroller. This means that it has to be instructed on the number of channels and the duration of each of them. The advantages are that usually a PPM signal updates faster than a PWM and that only a single wire is required.

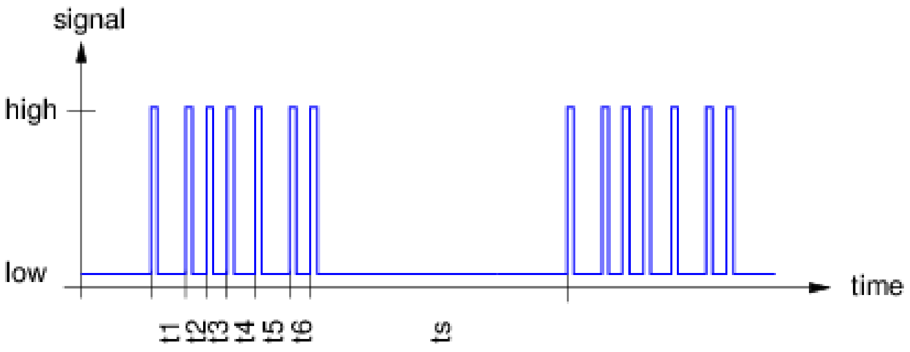

Figure 2.5: example of a six channel PPM signal (from Pabr.org website (2018) [9])

2.5 Typical implementation

Usually in PWM modulation the signal goes high for $1ms$ to $2ms$ and repeats every $20ms$ ($50Hz$). When controlling a motor through its ESC the values are expressed in the amount of time (in microseconds) the signal stays high rather than its duty cycle value. in this case those values will range from 1000 to 2000.
The PPM modulation still repeats its pulse every $20ms$ but this time it is divided into smaller time period of $2ms$, one for each channel. This means that if we have 8 channels we are going to use $16ms$ and the remaining $4ms$ are used as delay between two consecutive pulses.
In This way the update frequency does not change and the two protocols can be converted without any information losses.

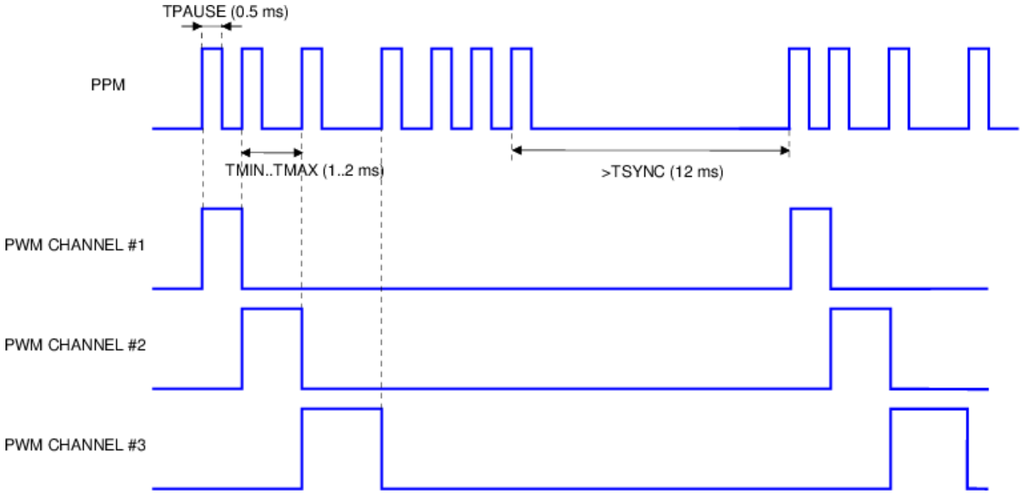

Figure 2.6: example of PPM to PWM conversion (from Pabr.org website (2018) [9])

Chapter 3

Parameters acquisition

3.1 3D model and inertia matrix

The inertia matrix, the first step in order to characterize the drone, has been obtained with the software SolidWorks from its 3D model.
Due to the drone structural complexity like the retractable landing gear, or the advance vibration damping system, some parts have not been perfectly modelled but still the volumes and mass concentrations which greatly affect the definition of the inertia tensor, are as close as possible to the real ones.

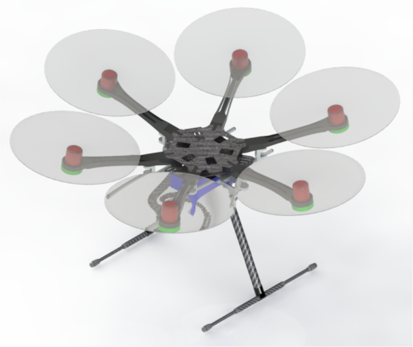

Figure 3.1: S800 SolidWorks rendered 3D model

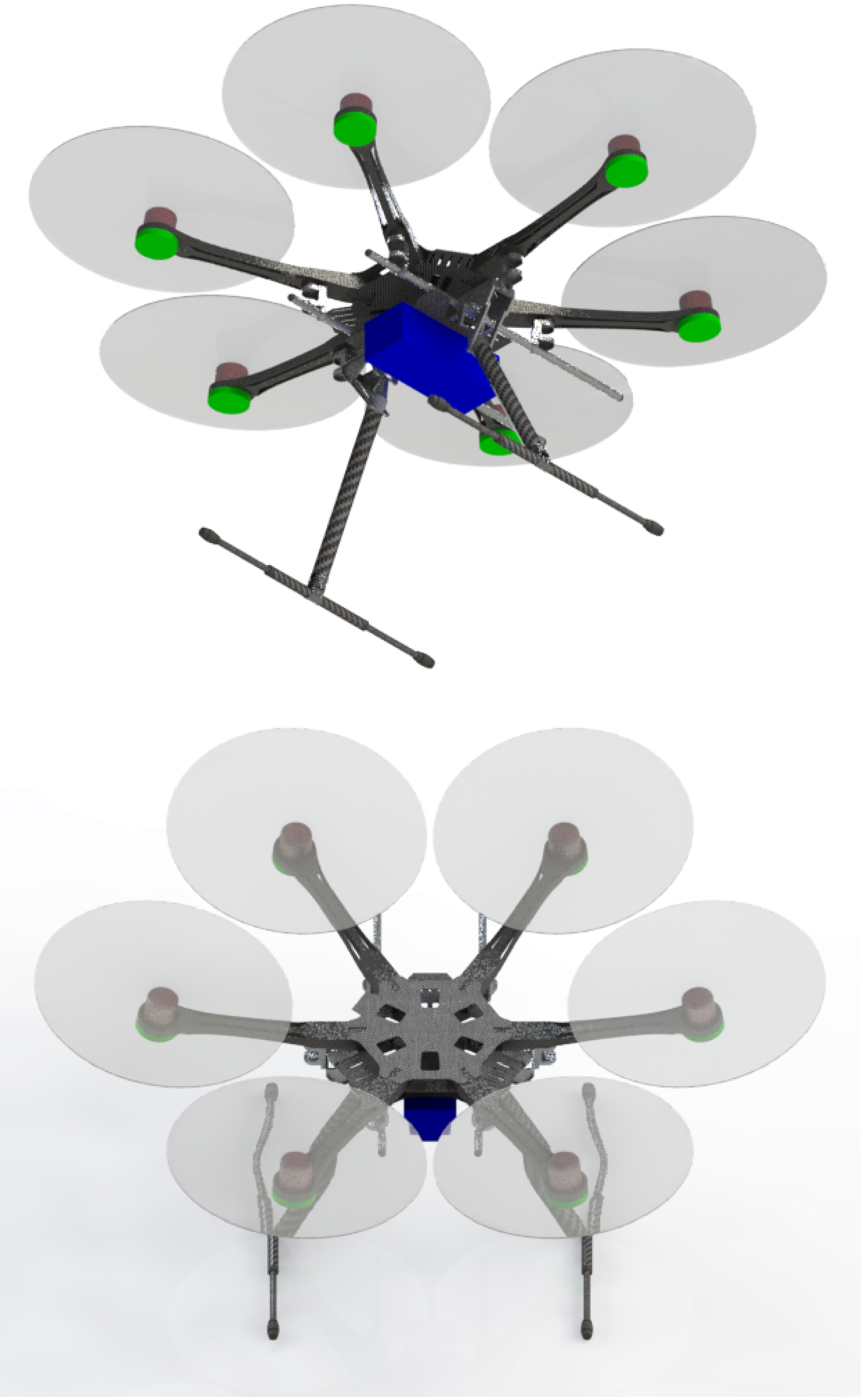

Figure 3.2: DJI S800 rendered images

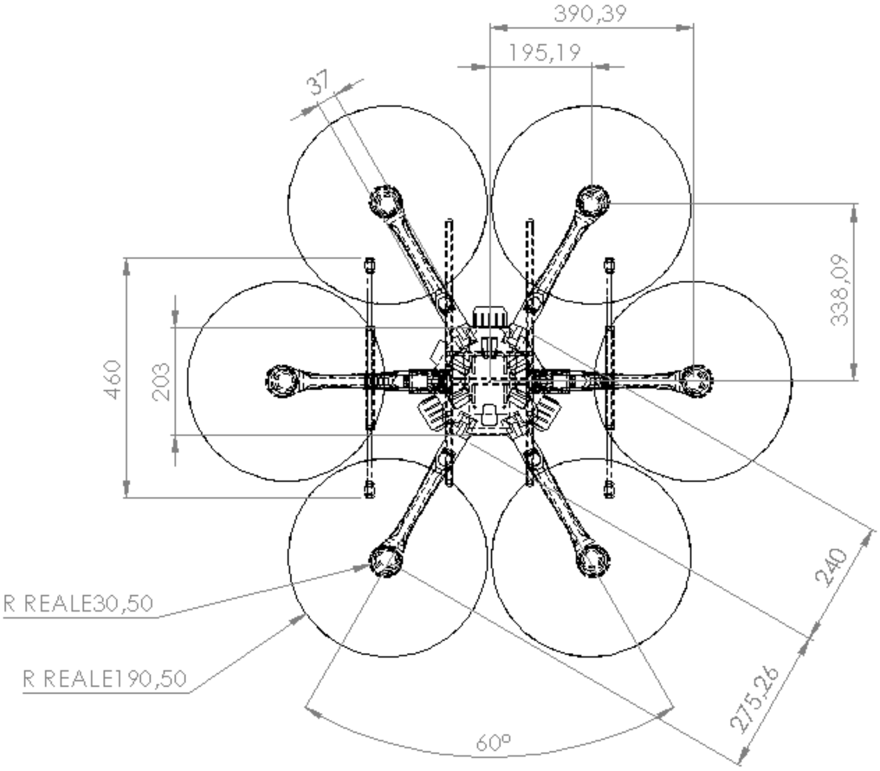

Figure 3.3: drone top view with upper plate and arms dimensions

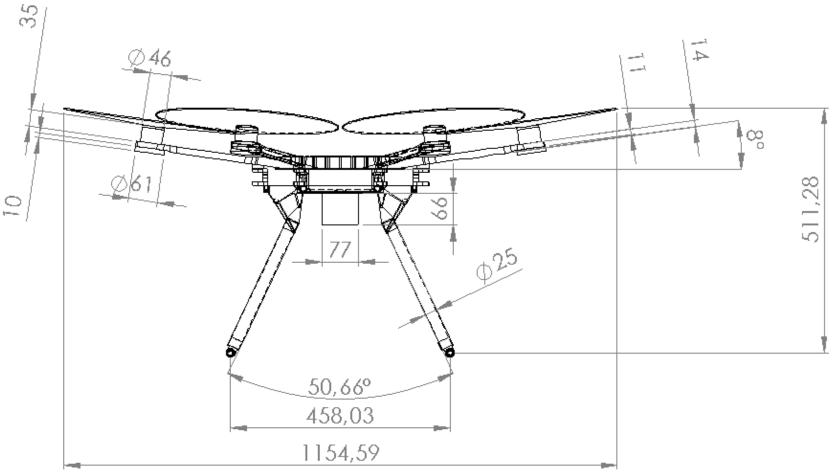

Figure 3.4: drone front view with landing gear, motors, ESC and battery dimensions

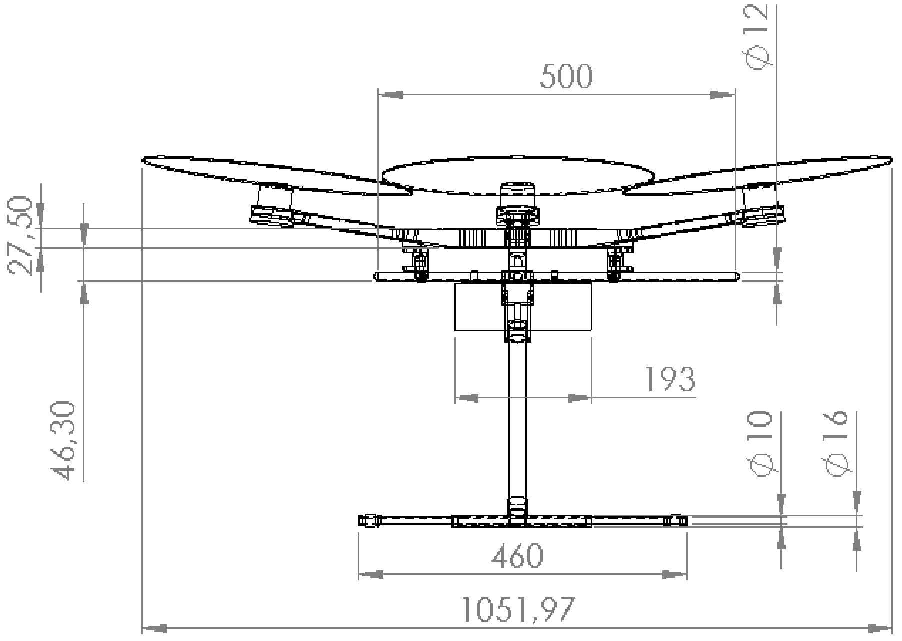

Figure 3.5: drone side view with dimensions

The intertia tensor is

$$\boldsymbol{J} = \begin{pmatrix} 179584764.54 & -40140.55 & -1292084.86 \\ -40140.55 & 177914370.54 & -42106.35 \\ -1292084.86 & -42106.35 & 272526580.62 \end{pmatrix} \tag{3.1}$$

measured in $\frac{g}{mm^2}$.

This matrix is calculated at the centre of gravity with respect to the body axis frame where x points forward, y is pointing out of the right motor and z is the axis perpendicular to the plane defined by x and y.

The fact that the inertia matrix is not diagonal means that the axes chosen are not "principal axes", however the non diagonal terms are at least two order of magnitude smaller than the diagonal ones.

Moreover it can be seen that the rotation about x and y, roll and pitch movement respectively, have a similar and smaller inertia with respect to the rotation about z or yaw, due to the fact that the heaviest part of the drone like its arms and motors are far from it.

3.2 Motor characterization

In order to model the motor a workbench has been used. The whole system composed of an ESC, a load cell and an RPM counter sensor is controlled by an Arduino Due board and permits to calibrate the Electronic Speed Controller and command the motor, while detecting its thrust, RPM, current, voltage and absorbed power.
The results are then exported to a text file and ”given” to the software Matlab.

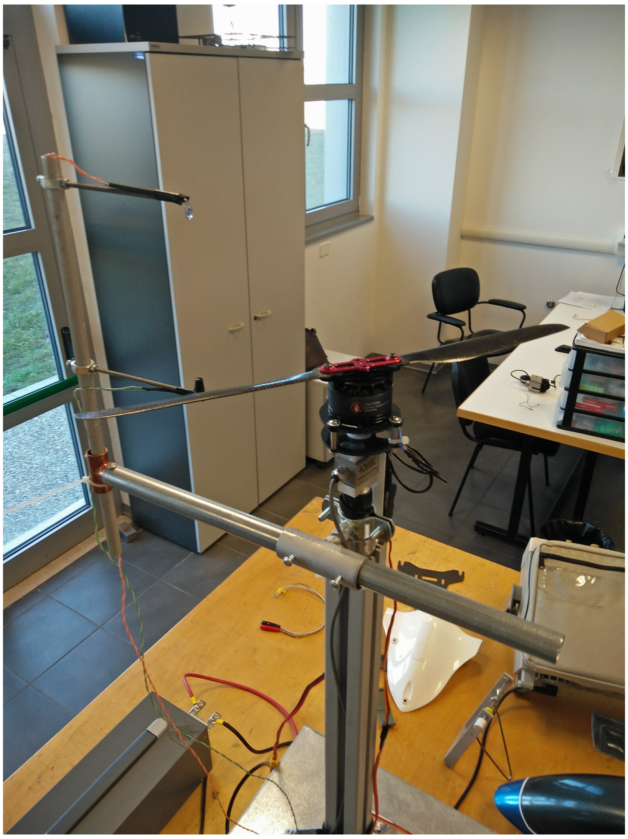

Figure 3.6: motor test bench in the university laboratory

3.2.1 Measured thrust and RPM

Below are plotted the results coming from the test bench together with the $2nd$ order degree curves which better fit the experimental results.
Those curves are obtained with the Matlab command *polyfit* which returns the coefficients of their polynomial, this will permit the mathematical model to calculate the values of thrust and revolutions per minute for the requested amount of throttle.

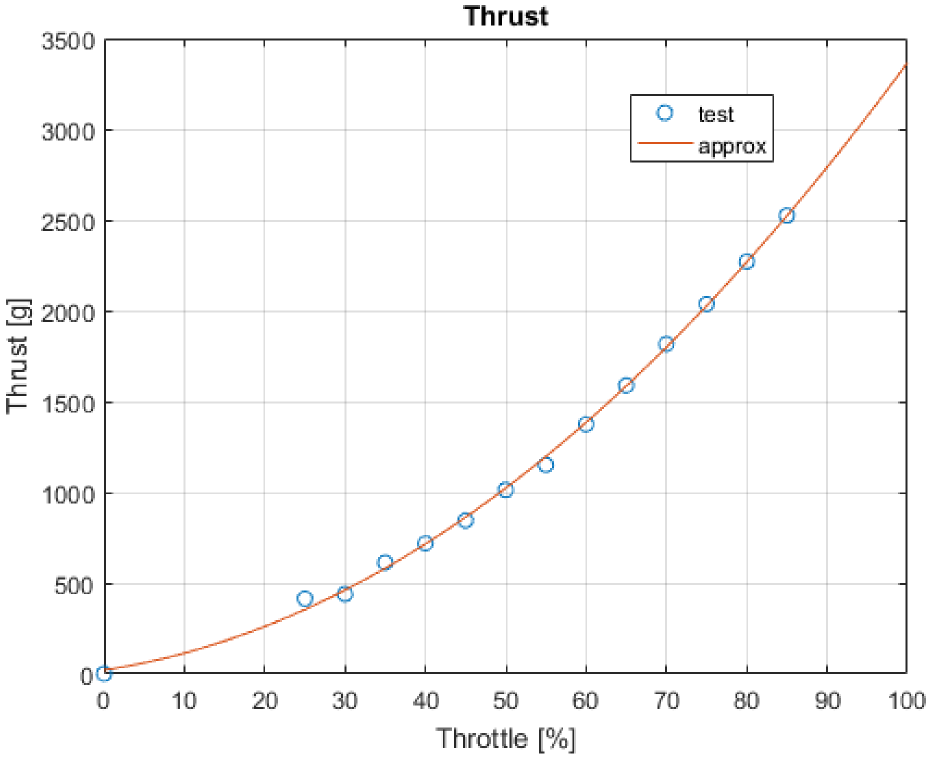

Figure 3.7: experimental results from test bench (blue circles) and approximated curve used in the simulator

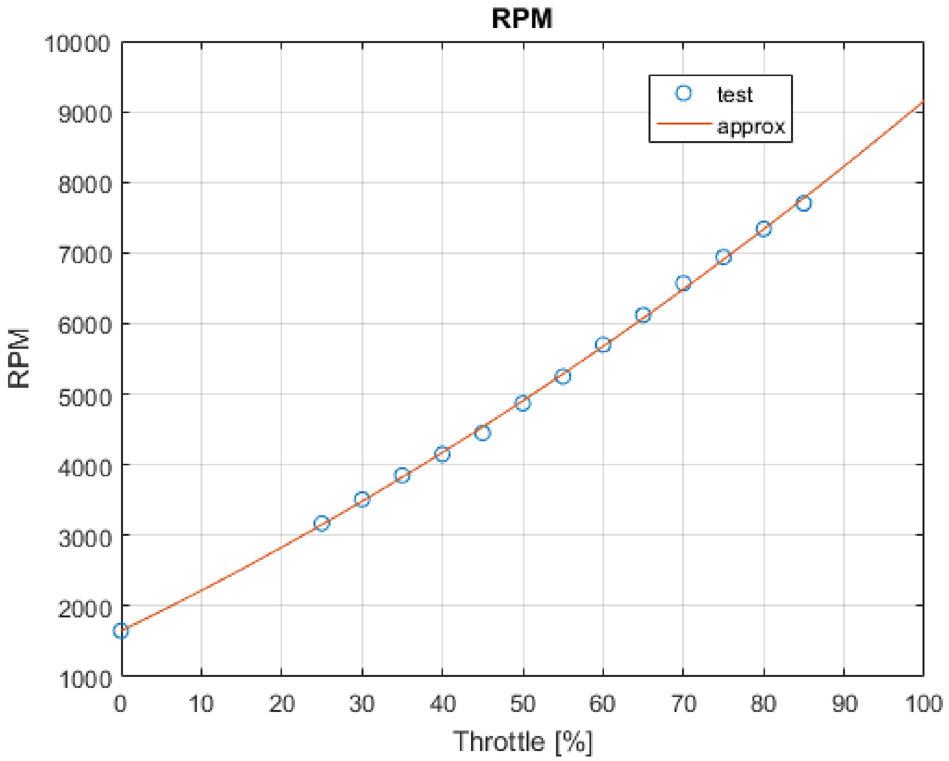

Figure 3.8: experimental results from test bench (blue circles) and approximated curve used in the simulator

3.2.2 Torque estimation

The torque values are not directly derivable from the experimental data and a further estimation is mandatory.
The method consists into exploit the non-dimensional coefficients of a propeller in order to obtain the real power consumed and hence the torque.
From the momentum theory analysis in hovering flight (Appendix A) we know that the induced velocity through the rotor is function of the thrust, a reference area such as the disk area, a reference velocity such as blade tip speed and the density of the flow.

$$v_i = f(T, A, V_{tip}, \rho)$$

According to the Buckingham method of dimensional analysis this means that there are five variables and three fundamental dimensions (namely: mass, length and time) for a total of two non-dimensional terms.
Choosing the linearly independent variables ρ, A and V_{tip} and one additional variable T, we finally get the first of the two Π term.

$$\Pi_1 = \rho^a A^b V_{tip}^c T$$

In dimensional form

$$M^a L^{-3a} | L^{2b} | L^c T^{-c} | M^1 L^1 T^{-2}$$

$$\begin{cases} a + 1 = 0 \\ -3a + 2b + c + 1 = 0 \\ -c - 2 = 0 \end{cases} \Rightarrow \begin{cases} a = -1 \\ b = -1 \\ c = -2 \end{cases}$$

$$\Pi_1 = \frac{T}{\rho A V_{tip}^2} = C_T \tag{3.2}$$

with $V_{tip} = \Omega R$.

Equivalently

$$C_Q = \frac{Q}{\rho A V_{tip}^2 R} \tag{3.3}$$

$$C_P = \frac{P}{\rho A V_{tip}^3} \tag{3.4}$$

respectively called *thrust coefficient*, *torque coefficient* and *power coefficient.* Notice that because power is related to torque by $P = \Omega Q$, then numerically $C_P \equiv C_Q$.

If we add V_i as additional variable we get

$$\Pi_2 = \rho^a A^b V_{tip}^c V_i$$

in dimensional form

$$M^a L^{-3a} | L^{2b} | L^c T^{-c} | L^1 T^{-1}$$

$$\begin{cases} a = 0 \\ -3a + 2b + c + 1 = 0 \\ -c - 1 = 0 \end{cases} \Rightarrow \begin{cases} a = 0 \\ b = 0 \\ c = -1 \end{cases}$$

$$\Pi_2 = \frac{V_i}{V_{tip}} = \lambda_i \tag{3.5}$$

This is called *inflow coefficient* (or *inflow ratio*) and using the result for the induced velocity from momentum theory (Eq. B.12) and Eq. 3.2, we can relate it to the thrust coefficient in hover

$$\lambda_h \equiv \lambda_i = \sqrt{\frac{T}{2\rho A}} \frac{1}{V_{tip}} = \sqrt{\frac{T}{2\rho A V_{tip}^2}} = \sqrt{\frac{C_T}{2}} \tag{3.6}$$

Based on momentum theory the power coefficient for the hovering rotor is

$$C_P = \frac{Tv_i}{\rho A \Omega} = \left(\frac{T}{\rho A (\Omega R)^2}\right)\left(\frac{v_i}{\Omega R}\right) = C_T \lambda_i = \frac{C_T^{3/2}}{\sqrt{2}} \tag{3.7}$$

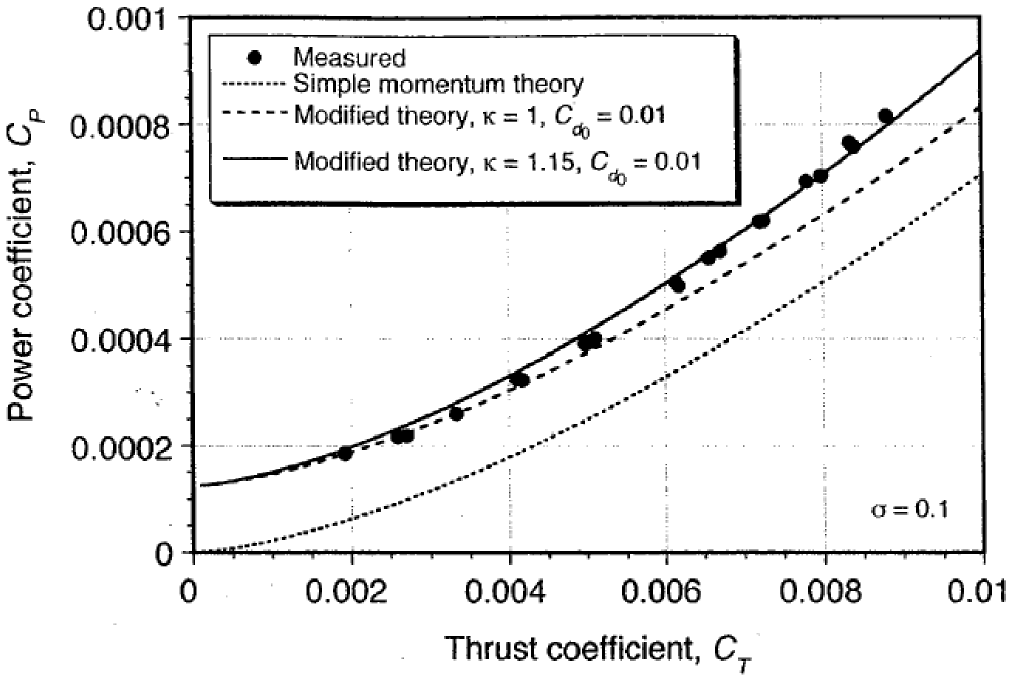

Figure 3.9: comparison of prediction made with momentum theory to measured power for a hovering rotor (from J. Gordon Leishman (2006), p. 68 [10])

Fig. 3.9 shows a comparison of the simple momentum theory with thrust and power measurements made for a hovering rotor using the last calculated equation.

Notice that the momentum theory underpredicts the actual power required, but the predicted trend that $C_P \propto C_T^{3/2}$ is essentially correct. These differences between the momentum theory and experiment occur because viscous effects (i.e., non ideal effects) have been totally neglected so far [10].

Correction for non ideal effects on rotor performances

To take into account these performance losses an *induced power correction factor* is multiplied to increase the curve's slope. This coefficient is derived from rotor measurements or flight tests and it encompasses a number of non ideal, but physical effects, such as:

- nonuniform inflow
- tip losses
- wake swirl
- less than ideal wake contraction
- finite number blades
- etc...

Usually for preliminary design a $k = 1.15$ is used to get the *ideal power coefficient*

$$C_{Pi} = \frac{kC_T^{3/2}}{\sqrt{2}} \tag{3.8}$$

The next thing to do is to estimate the profile power consumed by the rotor in order to "lift" the C_P curve to the measured values, this requires a knowledge of the drag coefficients of the airfoils that make up the rotor blades.
The airfoil drag coefficient will be a function of both Reynolds number Re and Mach number M which obviously vary along the span of the blade.

A result for the profile power can be obtained from a blade element method in which the total profile power P_0 can be obtained from an element-by-element analysis of sectional drag forces and by radially integrating the sectional drag force along the length of the blade using

$$P_0 = \Omega N_b \int_0^R D y dy \tag{3.9}$$

where N_b is the number of blades and D is the drag force per unit span at a section on the blade at a distance y from the rotational axis.
The drag force can be expressed conventionally as

$$D = \frac{1}{2}\rho U^2 c C_d = \frac{1}{2}\rho(\Omega y)^2 c C_d \tag{3.10}$$

where c is the blade chord.
If the section profile drag coefficient C_d, is assumed to be constant ($= C_{d0}$) and independent of Re and M (which is not an unrealistic first assumption), and the blade is not tapered in planform (i.e., a rectangular blade), then the profile power integrates out to be

$$P_0 = \frac{1}{8}\rho N_b \Omega^3 c C_{d0} R^4 \tag{3.11}$$

converting to a standard power coefficient by dividing through by $\rho A(\Omega R)^3$ and grouping the term $\frac{N_b c R}{A}$ into the so called *solidity ratio* σ, gives

$$C_{P0} = \frac{1}{8}\left(\frac{N_b c R}{A}\right) C_{d0} = \frac{1}{8}\left(\frac{N_b c R}{\pi R^2}\right) C_{d0} = \frac{1}{8}\sigma C_{d0} \tag{3.12}$$

Armed with these estimates of the induced and profile power losses, it is possible to recalculate the rotor power requirements by using the modified momentum theory

$$C_P = C_{Pi} + C_{P0} = \frac{k C_T^{3/2}}{\sqrt{2}} + \frac{1}{8}\sigma C_{d0} \tag{3.13}$$

This alternative results are shown in Fig. 3.9 as a continue line and labeled "modified theory" [10].

Plotting results

Using the method just described above, I was able to obtain a first estimation of the torque produced by each motor; the value of C_T (and hence C_{Pi}) comes directly from Eq. 3.2 and the measured thrust.
Instead to obtain C_{d0} another set of estimations were necessary for which came in our help the software XFLR5 (or XFoil), an analysis tool for airfoils, wings and planes operating at low Reynolds numbers.
The airfoil chosen from the UIUC database [11] is the ARA-D 6% which recall very closely the carbon fiber airfoil used on the rotor blades due to its small positive camber and its almost constant thickness along the chord.

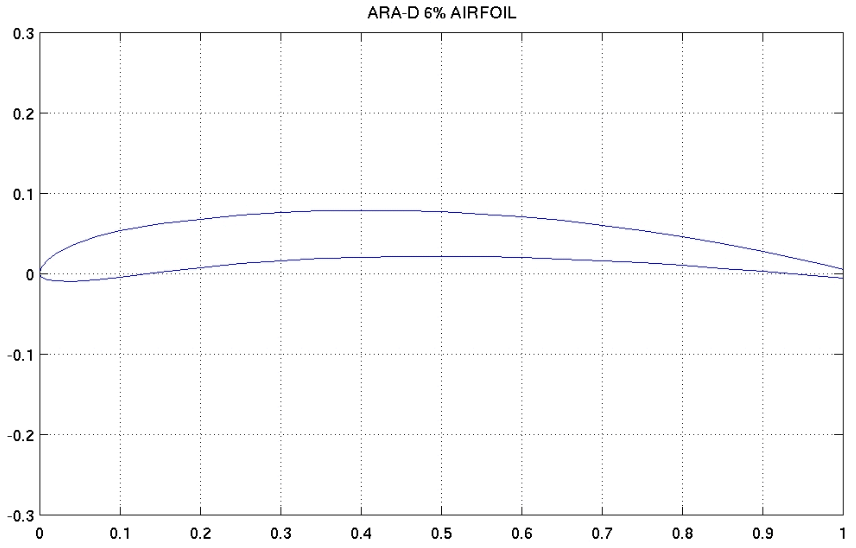

Figure 3.10: ARA-D 6% airfoil (from UIUC Airfoil Coordinates Database website (2018) [11])

Then Re number is calculated from the well known formula

$$Re = \frac{vl}{\nu} \tag{3.14}$$

where v and l are respectively the velocity of the spinning blade and the airfoil chord at 3/4 of the rotor radius ($c_{3/4} = 0.022m$) and $\nu = 1.48e - 5m^2/s$ is the kinematic viscosity of air at $15°C$.
The blade speed at 3/4 of rotor radius is given by the equation

$$v_{3/4} = \frac{2\pi}{60} RPM R_{3/4} \tag{3.15}$$

and results $47.179m/s$ at 25% and $136.979m/s$ at 100% of throttle.
The parameters set to perform the batch foil analysis are:

- Analysis type 1, corresponds to a given wing at a fixed velocity going over an angle of attack range
- Bach variables by range
  - min $Re = 70131$, obtained at 25% of throttle
  - max $Re = 203617$, obtained at 100% of throttle
  - Re increment = 10000, determines the number of lines in the C_l/C_d graph
  - $M = 0$, the Mach number can be left = 0 for a first approximation
  - $Ncrit = 4$, which correspond to a dirty wind tunnel using the transition e^n criterion
- Forced transition
  - Top transition location (x/c) = 1.00
  - Bottom transition location (x/c) = 1.00
- Analysis range with specified α
  - min $\alpha = -5deg$
  - max $\alpha = 5deg$
  - α increment = $0.1deg$, determines the number of points
- max iterations = 100

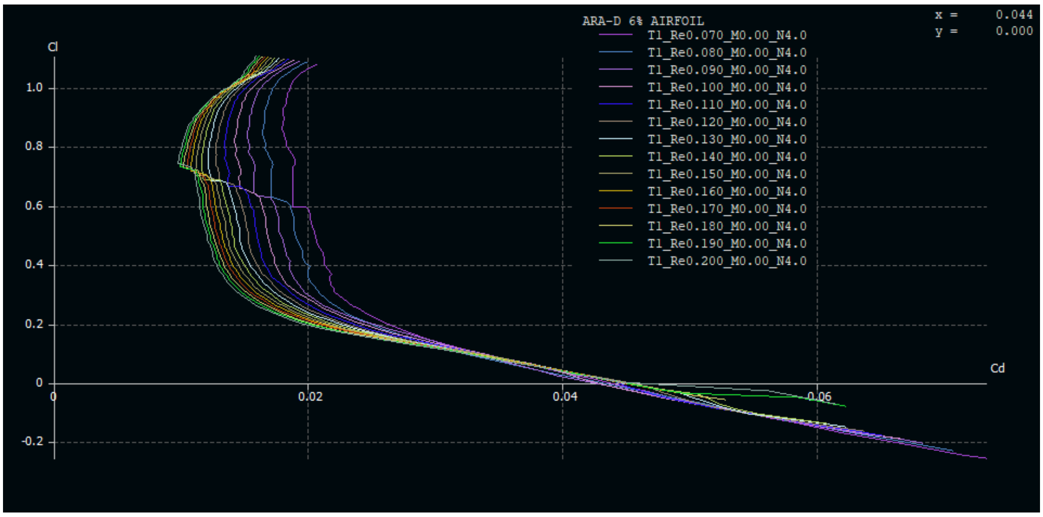

Figure 3.11: ARA-D 6% C_l/C_d coefficients. The different curves represent different rotation regimes of the propeller. The points in which the curves intersects the axis of abscissas ($C_l = 0$) correspond to C_{d0}

The graph shows that the C_{d0} value is in the neighborhood of 0.044 and does not change significantly with Re.

Finally, by recalling that $C_P \equiv C_Q$ and multiplying for $\rho A V_{tip}^2 R$ we get a first estimation of the torque to apply at the drone mathematical model.

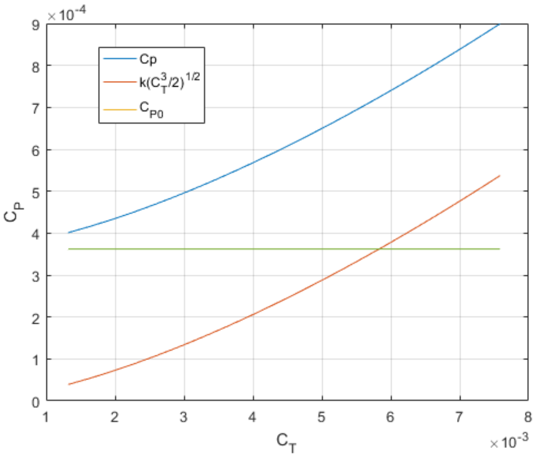

Figure 3.12: C_p estimation with momentum theory for an hovering rotor

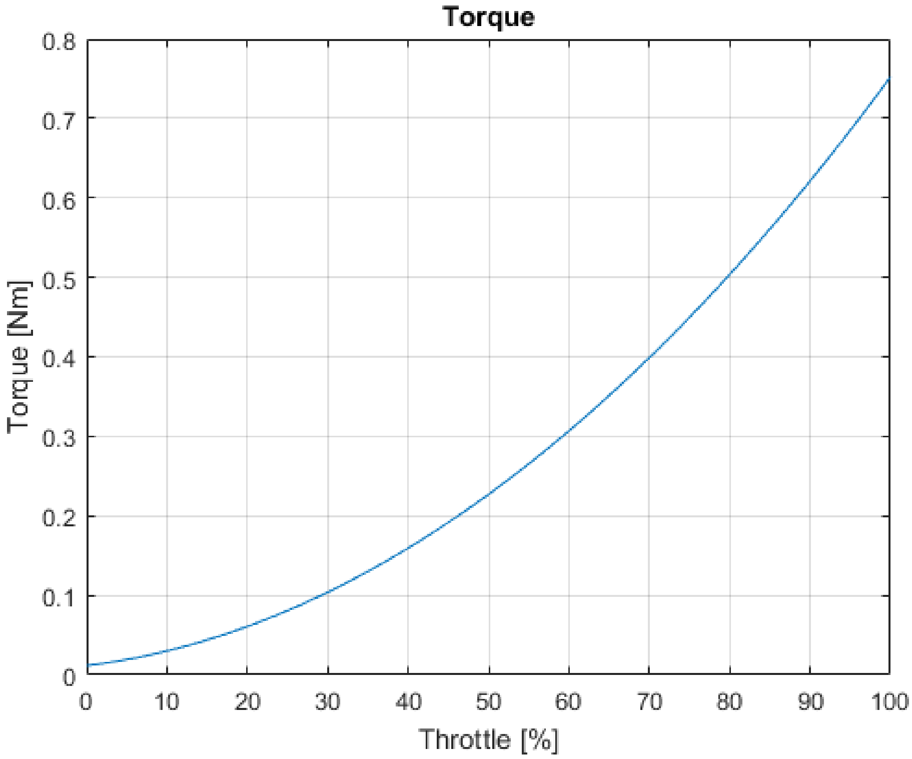

Figure 3.13: approximated curve used in the simulator

This value will most likely be different from the real one and it might be refined in a second moment by comparing it to the real drone dynamics.

3.3 Drone's performances estimation

From the data collected until now it is possible to have a first estimation of the maximum performances that the simulated drone will have:

- max thrust = $20.196kg$
- max torque_x = $19.336Nm$
- max torque_y = $22.328Nm$
- throttle for hover = 48.32%
- max angle for level flight = $73deg$
- max yaw acceleration = $474deg/s^2$

Of course these values represent the flight envelope limits reachable only in ideal conditions.
Moreover the flight controller itself is usually designed with far more conservative limits than those listed above to ensure a safety margin from dangerous flight attitudes and velocities.

Chapter 4

Drone mathematical model

4.1 State variables

In this section are derived the expressions for the kinematics and the dynamics.
While these expressions are general to any rigid body, some notation and coordinate frames used are typical in the aeronautics literature.

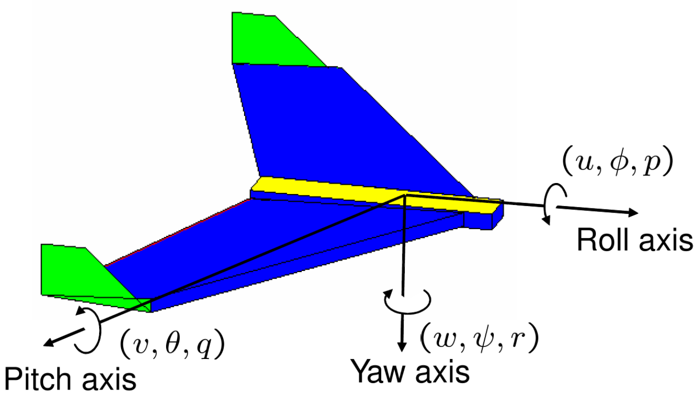

Figure 4.1: definition of axes (from Randal W. Beard(2008), p. 12 [12])

Below are listed the state variables which describe the position and attitude of the rigid body:

- p_n = the inertial (north) position of the drone along $\hat{\imath}^i$ in $\mathcal{F}^i$
- p_e = the inertial (east) position of the drone along $\hat{\jmath}^i$ in $\mathcal{F}^i$
- h = the altitude of the drone measured along $-\hat{k}^i$ in $\mathcal{F}^i$
- u = the body frame velocity measured along $\hat{\imath}^b$ in $\mathcal{F}^b$
- v = the body frame velocity measured along $\hat{\jmath}^b$ in $\mathcal{F}^b$
- w = the body frame velocity measured along $\hat{k}^b$ in $\mathcal{F}^b$
- ϕ = the roll angle defined with respect to $\mathcal{F}^{v2}$
- θ = the pitch angle defined with respect to $\mathcal{F}^{v1}$
- ψ = the yaw angle defined with respect to $\mathcal{F}^v$
- p = the roll rate measured along $\hat{\imath}^b$ in $\mathcal{F}^b$
- q = the pitch rate measured along $\hat{\jmath}^b$ in $\mathcal{F}^b$
- r = the yaw rate measured along $\hat{k}^b$ in $\mathcal{F}^b$

4.2 Kinematics

Position - velocity relationship

The state variables p_n, p_e, and $-h$ are inertial frame values, whereas the velocities u, v, and w are body frame quantities.
Therefore the relationship between position and velocities is given by

$$\frac{d}{dt}\begin{pmatrix} p_n \\ p_e \\ -h \end{pmatrix} = \boldsymbol{R}_b^v \begin{pmatrix} u \\ v \\ w \end{pmatrix} = (\boldsymbol{R}_v^b)^T \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

$$\Downarrow$$

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{h} \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ s\theta & -s\phi c\theta & -c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{4.1}$$

[12]

Absolute angle - angular velocity relationship

The relationship between absolute angles ϕ, θ, and ψ, and the angular rates p, q, and r is also complicated by the fact that each of these quantities are defined in different coordinate frames. The angular rates are defined in the body frame $\mathcal{F}^b$, whereas the roll angle is defined in $\mathcal{F}^{v2}$, the pitch angle is defined in $\mathcal{F}^{v1}$, and the yaw angle is defined in the vehicle frame $\mathcal{F}^v$.
We need to relate p, q, and r to $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \boldsymbol{R}^b_{v2}(\dot{\phi}) \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + \boldsymbol{R}^b_{v2}(\phi)\boldsymbol{R}^{v2}_{v1}(\dot{\theta}) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \boldsymbol{R}^b_{v2}(\phi)\boldsymbol{R}^{v2}_{v1}(\theta)\boldsymbol{R}^{v1}_{v}(\dot{\psi}) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}$$

since $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ are small and noting that $\boldsymbol{R}^b_{v2}(\dot{\phi}) = \boldsymbol{R}^{v2}_{v1}(\dot{\theta}) = \boldsymbol{R}^{v1}_{v}(\dot{\psi}) = \boldsymbol{I}$

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + \boldsymbol{R}^b_{v2}(\phi) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \boldsymbol{R}^b_{v2}(\phi)\boldsymbol{R}^{v2}_{v1}(\theta) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}$$

$$\Downarrow$$

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \tag{4.2}$$

inverting the rotation matrix

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{4.3}$$

[12]

4.3 Rigid body dynamics

Translational motion

Let $\vec{V}$ be the velocity vector of the drone.
Newton's laws only hold in inertial frames therefore

$$m\frac{d\vec{V}}{dt_i} = \vec{f} \tag{4.4}$$

where m is the mass of the aircraft, $\vec{f}$ is the total force applied and $\frac{d}{dt_i}$ is the time derivative in the inertial frame.
From the equation of Coriolis (Eq. A.5) is it possible to write

$$m\frac{d\vec{V}}{dt_i} = m\left(\frac{d\vec{V}}{dt_b} + \vec{\omega}_b \times \vec{V}\right) = \vec{f} \tag{4.5}$$

Since the control force is computed and applied in the body coordinate system, and since $\vec{\omega}_b$ is measured in body coordinates, we will express the last equation in body coordinates using

$$\vec{V}_b = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{4.6}$$

$$\vec{\omega}_b = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{4.7}$$

$$\vec{f} = \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \tag{4.8}$$

Therefore

$$\frac{d\vec{V}_b}{dt_b} = -(\vec{\omega}_b \times \vec{V}_b) + \frac{1}{m}\vec{f}$$

$$\Downarrow$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = -\begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \frac{1}{m}\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}$$

$$\Downarrow$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m}\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \tag{4.9}$$

[12]

Rotational motion

For rotational motion Newton's second law states that

$$\frac{d\vec{h}}{dt_i} = \vec{m} \tag{4.10}$$

where $\vec{h}$ is the angular momentum and $\vec{m}$ is the applied torque.
Using the equation of Coriolis (Eq. A.5)

$$\frac{d\vec{h}}{dt_i} = \frac{d\vec{h}}{dt_b} + \vec{\omega}_b \times \vec{h} = \vec{m} \tag{4.11}$$

This equation is most easily solved in body coordinates where

$$\vec{h}_b = \boldsymbol{J}\vec{\omega}_b \tag{4.12}$$

$$\vec{m} = \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \tag{4.13}$$

$\boldsymbol{J}$ is the constant inertia matrix given by

$$\boldsymbol{J} = \begin{pmatrix} J_x & -J_{xy} & -J_{xz} \\ -J_{xy} & J_y & -J_{yz} \\ -J_{xz} & -J_{yz} & J_z \end{pmatrix} \tag{4.14}$$

The drone is essentially symmetric about all three axes therefore $J_{xy} = J_{xz} = J_{yz} = 0$ which implies that

$$\boldsymbol{J} = \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \tag{4.15}$$

and its inverse simplify to

$$\boldsymbol{J}^{-1} = \begin{pmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{pmatrix} \tag{4.16}$$

Now it is possible to write the Newton's second law in body coordinates

$$\boldsymbol{J}\frac{d\vec{\omega}_b}{dt_b} + \vec{\omega}_b \times \boldsymbol{J}\vec{\omega}_b = \vec{m}$$

$$\Downarrow$$

$$\frac{d\vec{\omega}_b}{dt_b} = -\boldsymbol{J}^{-1}[(\vec{\omega}_b \times \boldsymbol{J}\vec{\omega}_b) + \vec{m}]$$

$$\Downarrow$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = -\begin{pmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{pmatrix} \left[\begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} + \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \right]$$

$$\Downarrow$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix} \tag{4.17}$$

[12]

4.4 6 DOF model

Resuming the four equation obtained in the previous sections we get the six-degrees-of-freedom equations of motions in Euler angles.
"This system considers the rotation of a body-fixed coordinate frame $\mathcal{F}^b$ about a flat Earth reference frame $\mathcal{F}^i$. The origin of the body-fixed coordinate frame is the center of gravity of the body, and the body is assumed to be rigid, an assumption that eliminates the need to consider the forces acting between individual elements of mass.
The flat Earth reference frame is considered inertial, an excellent approximation that allows the forces due to the Earth's motion relative to the "fixed stars" to be neglected [12]."

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{h} \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ s\theta & -s\phi c\theta & -c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{4.1}$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \tag{4.3}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{4.9}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix} \tag{4.17}$$

4.5 Complex model

To complete the objective set by this thesis, it is necessary to build at least two different models, one more accurate and complex and one simpler, which uses some simplified hypotheses and neglect some secondary effects.
In these terms the 6 DOF model and the Simulink developing environment comes in our help because of the "modular" structure that the forces and moments acting on the model have. In fact one can add, for example, the drag effect on the drone body by simply summing it term by term with the force and moment vectors. The only thing that need to be checked is that these components have to be in body frame.
The complex model described below takes into account six "phenomena":

1. motor thrust
2. gravity
3. aerodynamic drag
4. change in thrust during non hovering flights
5. rotor inflow during attitude change
6. gyroscopic torque generated by spinning rotors

4.5.1 Motors

Each motor produces a thrust T and a torque Q. The total force acting on the drone is given by

$$F = T_1 + T_2 + T_3 + T_4 + T_5 + T_6 \tag{4.18}$$

The rolling torque is produced by the forces of the right and left motors as

$$\tau_\phi = -b_2 T_1 + b_2 T_2 + b T_3 + b_2 T_4 - b_2 T_5 - b T_6 \tag{4.19}$$

Similarly, the pitching torque is produced by the forces of the front and back motors as

$$\tau_\theta = b_1 T_1 + b_1 T_2 - b_1 T_4 - b_1 T_5 \tag{4.20}$$

where b, b_1 and b_2 are the distances of the motors from the center of mass. Due to the Newton's third law, the drag of the propellers produces a yawing torque on the body of the drone. The direction of the torque will be in the opposite direction of the motion of the propeller, therefore the total yawing torque is given by

$$\tau_\psi = Q_1 - Q_2 + Q_3 - Q_4 + Q_5 - Q_6 \tag{4.21}$$

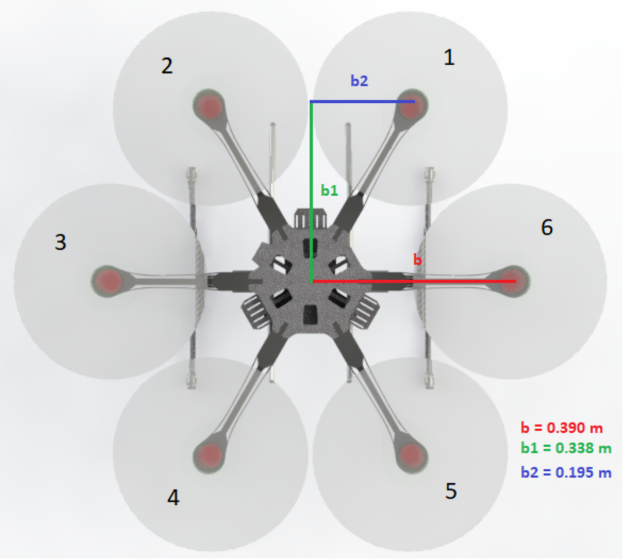

Figure 4.2: Motors' number and dimensions

4.5.2 Gravity

In addition to the force exerted by the motor, gravity also exerts a force on the drone. In the inertial frame $\mathcal{F}^i$, the gravity force acting on the center of mass is given by

$$\vec{f_g^i} = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \tag{4.22}$$

However, since $\vec{V}$ in Eq. 4.3 is expressed in $\mathcal{F}^b$, we must transform to the body frame

$$\vec{f_g^b} = \boldsymbol{R}_i^b \vec{f_g^i} = \begin{pmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{pmatrix} \tag{4.23}$$

4.5.3 Drag

Mainly due to the fact that on an hexacopter there are no lifting surfaces and the the velocity reached are relatively small, the drag modelling is done with a simple *equivalent flat plate area* method.
Exploiting the well known drag equation

$$D = \frac{1}{2}\rho C_d A V^2 \tag{4.24}$$

which can be written with respect to the body frame as

$$-f_{drag} = -\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = -\frac{1}{2}\rho C_d \begin{pmatrix} A_{front} & A_{side} & A_{top} \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}^2 \tag{4.25}$$

where C_d is the drag coefficient for a flat plate perpendicular to the 3D flow and equal to 1.28 and the areas are obtained from the SolidWorks 3D model

$$\begin{pmatrix} A_{front} & A_{side} & A_{top} \end{pmatrix} = \begin{pmatrix} 0.061372 & 0.056258 & 0.145728 \end{pmatrix}$$

measured in $[m^2]$.

4.5.4 Thrust in non hovering flights

Momentum theory assumptions

Any rotating-wing vehicle operates in a variety of flight regimes as hover, climb, descent and forward flight and of course a combination of these four. In hover or axial flight the flow is axisymmetric and the flow through the rotor is either upward or downward. These are the easiest flow regimes to analyze and can usually be predicted by means of mathematical models. Although it must be remembered that the actual physical flow about the rotor will comprise a complicated vortical wake structure the basic performance of the rotor can be analyzed by a simpler approach that has become known as the *Rankine-Froude momentum theory.*

In this theory the flow through the rotor is assumed to be:

- one-dimensional (flow properties change only with axial position relative to the rotor)
- quasi-steady (flow properties at a point do not change with time)
- incompressible
- inviscid (the flow is an ideal fluid)

The ideal fluid considered, the one that generates no viscous shear between fluid elements and therefore no viscous losses, will lead to the assumption that the sole source of losses in the fluid are induced losses which come directly from the rotor thrust production [10].

Thrust equation

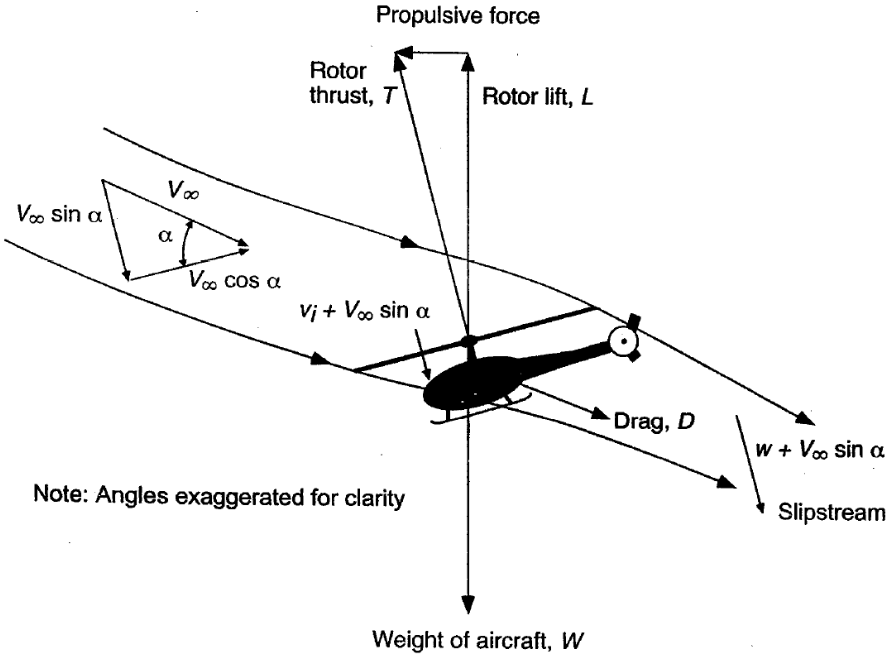

Figure 2.23 Glauert's flow model for the momentum analysis of a rotor in forward flight.

Figure 4.3: Glauert's flow model for the momentum analysis of a rotor in forward flight (from J. Gordon Leishman (2006), p. 93 [10])

In multi-rotor aircraft each rotor provides both lifting force and propulsive force. To do so they have to be tilted at an AoA relative to the oncoming flow.
Under these forward flight conditions the axisymmetry of the flow through the rotor is lost but, with some changes, the simple momentum theory can be extended to encompass these complications on the basis of certain assumptions.
The mass flow rate $\dot{m}$ through the actuator disk is

$$\dot{m} = \rho A U \tag{4.26}$$

where U is the resultant velocity at the disk and is given by

$$U = \sqrt{(V_\infty \cos\alpha)^2 + (V_\infty \sin\alpha + v_i)^2} = \sqrt{V_\infty^2 + 2V_\infty v_i \sin\alpha + v_i^2} \tag{4.27}$$

The application of the conservation of momentum in a direction normal to the disk gives

$$T = \dot{m}(w + V_\infty \sin\alpha) - \dot{m}V_\infty \sin\alpha = \dot{m}w \tag{4.28}$$

and by the application of conservation of energy between the ∞ and the 0 cross plane, we obtain

$$\begin{aligned} P &= T(v_i + V_\infty \sin\alpha) \\ &= \frac{1}{2}\dot{m}(V_\infty \sin\alpha + w)^2 - \frac{1}{2}\dot{m}V_\infty^2 \sin^2\alpha \\ &= \frac{1}{2}\dot{m}(2V_\infty w \sin\alpha + w^2) \end{aligned} \tag{4.29}$$

Using the above written equations

$$2wv_i + 2V_\infty w \sin\alpha = 2V_\infty w \sin\alpha + w^2 \Rightarrow 2v_i = w \tag{4.30}$$

which is the same result of hover case. Therefore

$$T = 2\dot{m}v_i = 2\rho AUv_i = 2\rho Av_i\sqrt{(V_\infty \cos\alpha)^2 + (V_\infty \sin\alpha + v_i)^2} \tag{4.31}$$

[10].

Please note that in hovering condition $V_\infty \equiv 0$ and the thrust equation returns equal to Eq. B.11

$$T = 2\rho Av_i^2 \tag{4.32}$$

In axial flight instead we will have $\alpha = 90deg$ and the resultant velocity at the disk became

$$U = V_c \sin\alpha + v_i \tag{4.33}$$

where $V_c = V_\infty$ is the climb velocity. Its thrust will be

$$T = 2\rho Av_i(V_c + V_i) \tag{4.34}$$

The equations Eq. 4.32 and Eq. 4.34 obtained from the more general thrust formula (Eq. 4.31) confirm the results obtained by the momentum theory.

Induced velocity

From Eq. 4.31 it is clear that thrust is directly related to the induced velocity at the rotor disk. The above cited equation can be written as

$$v_i = \frac{T}{2\rho A\sqrt{(V_\infty \cos\alpha)^2 + (V_\infty \sin\alpha + v_i)^2}} \tag{4.35}$$

recalling that for hovering flight $v_h^2 = T/2\rho A$ (Eq. B.12)

$$v_i = \frac{v_h^2}{\sqrt{(V_\infty \cos\alpha)^2 + (V_\infty \sin\alpha + v_i)^2}} \tag{4.36}$$

In order to get this equation in a non dimensional form both sides are now divided by the tip speed velocity getting

$$\mu = \frac{V_\infty \cos\alpha}{\Omega R} = \frac{\sqrt{(-u)^2 + (-v)^2}}{\Omega R} \tag{4.37}$$

$$\lambda = \frac{V_\infty \sin\alpha + v_i}{\Omega R} = \frac{-w + v_i}{\Omega R} \tag{4.38}$$

called respectively *advance* and *inflow ratio.*
Being α the angle between the rotor disk and the incoming flow and being the rotor disk plane lying on the drone's body frame $x-y$ plane, it is evident that $V_\infty \cos\alpha$ is the component of wind parallel to the rotor disk and directed in the opposite direction with respect to $\hat{i}^b + \hat{j}^b$. In the same way $V_\infty \sin\alpha$ is the wind component perpendicular to the rotor disk and directed towards $-\hat{k}^b$. Using Eq. 4.36, the induced inflow ratio can be defined as

$$\lambda_i = \frac{v_i}{\Omega R} = \frac{\lambda_h^2}{\sqrt{\mu^2 + \lambda^2}} \tag{4.39}$$

recalling that in hover case $\lambda_h = \sqrt{C_T/2}$ (Eq. 3.6)

$$\lambda_i = \frac{C_T}{2\sqrt{\mu^2 + \lambda^2}} \tag{4.40}$$

Finally substituting this into Eq. 4.38 we get the solution for the inflow ratio

$$\lambda = \frac{-w}{\Omega R} + \frac{C_T}{2\sqrt{\mu^2 + \lambda^2}} \tag{4.41}$$

which is solved numerically with an iterative method.

Iterative method

The iterative method used is the Newton-Raphson procedure which, for the price of computing a simple first derivative, gives back a much more rapid convergence. The iteration scheme is

$$\lambda_{n+1} = \lambda_n - \frac{f(\lambda_n)}{f'(\lambda_n)} \tag{4.42}$$

where n is the iteration number and f is Eq. 4.41 rearranged in the form $f(\lambda) = 0$

$$f(\lambda) = \lambda - \frac{-w}{\Omega R} - \frac{C_T}{2\sqrt{\mu^2 + \lambda^2}} = 0 \tag{4.43}$$

$$f'(\lambda) = 1 + \frac{C_T}{2}(\mu^2 + \lambda^2)^{-3/2}\lambda \tag{4.44}$$

The execution of this method is described point by point in the list below:

1. As starting value is used the hover condition with

   $$\lambda_0 = \lambda_h = \frac{v_h}{\Omega R} = \sqrt{\frac{T}{2\rho A(\Omega R)^2}} = \sqrt{\frac{C_T}{2}}$$

   where T is the requested thrust from the flight controller and ωR is the velocity at the blade tip. Both values come from the experimental data obtained at the test bench (see Cap. 3). Please note that this equation gives to us also the starting value for the thrust coefficient, C_T, used in f and f';

2. After the first iteration is complete a new value of inflow ratio is calculated and with this a new estimation of the induced velocity. Using this result into Eq. 4.31 we get the new thrust and its C_T which can be used together with λ in the next iteration;

3. The whole system is enclosed inside a *while* cycle which repeats the iterations until the difference between the newer and the older value of λ is less than 1%.

The difference in thrust obtained becomes clear when looking at the graph below which shows the thrust produced by the motor no. 1 in a constant $30deg$ nose down maneuver.
This attitude produces an accelerating forward flight in which, after the brief transient, the induced flow increases the thrust of an additional $0.5N$.

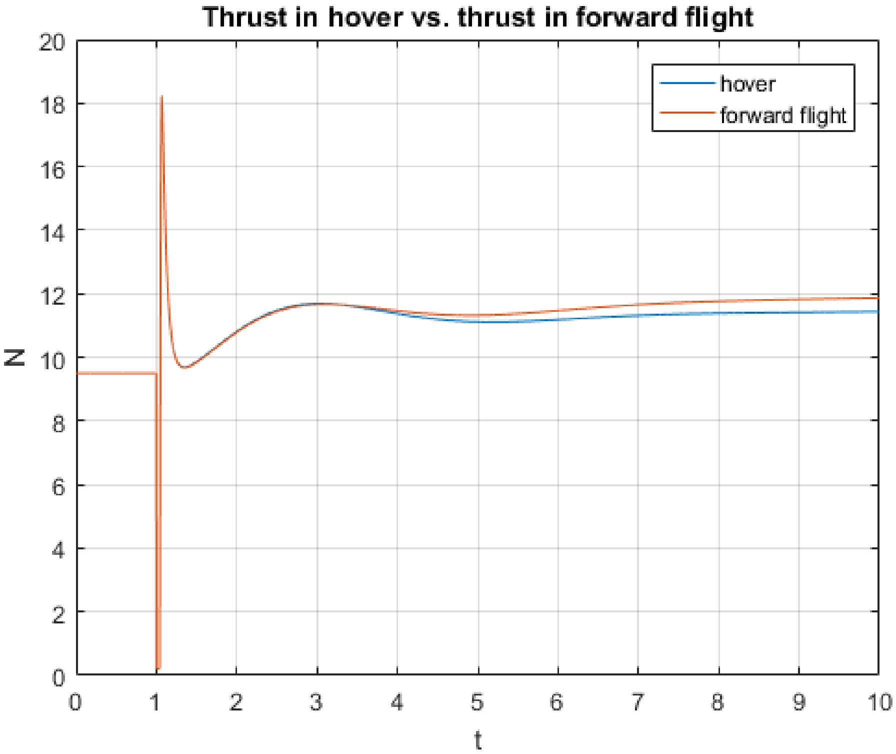

Figure 4.4: amount of thrust obtained from experimental data (yellow curve) and iterative method (blue curve) during a constant $30deg$ nose down level flight

Limitations during descending flights

”The inflow equation as given by Eq. 4.41 is widely employed for practical calculations involving rotors in climbing and descending flight in both axial and forward flight. However, a nonphysical solution will always be obtained if there is a descent or an upward component of velocity normal to the rotor disk that is between 0 and $2v_i$ (i.e., if $-2v_i \leq V_\infty \sin\alpha \leq 0$ in level flight). Under these conditions there can always be two possible directions for the flow and there can be no well-defined slipstream boundary as was assumed in the physical model. Therefore, the momentum theory cannot be applied under these conditions [10].”

In the mathematical model this happen each time the vertical velocity with respect to the body frame is equal or smaller than 0. In those conditions, as first estimation, the thrust used is the one obtained from the experimental data, hoping that vertical velocity and acceleration remains constrained inside the limit set by the flight controller ($CR_{max} = 2.5m/s$, $v\_acc_{max} = 2.5m/s$) during the simulations and the real flight.

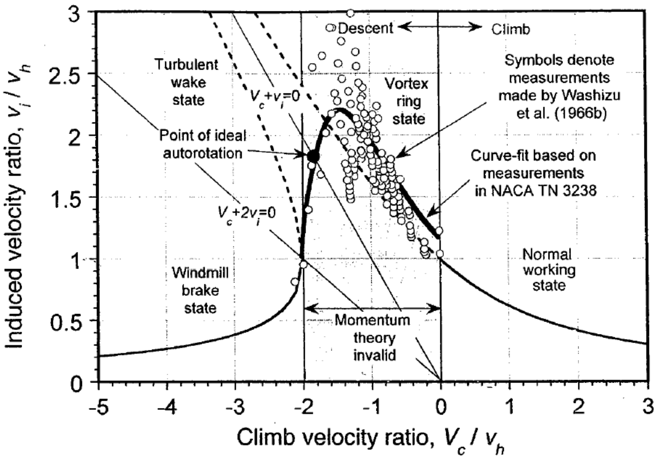

Figure 4.5: induced velocity variation as a function of climb and descent velocity based on momentum theory (from J. Gordon Leishman (2006), p. 84 [10])

4.5.5 Rotor inflow during attitude change

When the aircraft is changing its attitude, especially during fast maneuvers, each rotor will feel a "new" wind component depending on its position on the frame and the angular velocity the drone is experiencing. The equation used is

$$\begin{aligned}\vec{V}_{rel} = \vec{\omega}_b \times a\vec{r}m = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \begin{pmatrix} arm_x \\ arm_y \\ arm_z \end{pmatrix} = \begin{pmatrix} \hat{\imath}^b & \hat{\jmath}^b & \hat{k}^b \\ p & q & r \\ arm_x & arm_y & arm_z \end{pmatrix} \\ = (q\ arm_z - arm_y\ r)\hat{\imath}^b - (p\ arm_z - arm_x\ r)\hat{\jmath}^b + (p\ arm_y - arm_x\ q)\hat{k}^b\end{aligned} \tag{4.45}$$

with p, q e r the roll, pitch and yaw rate and arm is the vector which define the rotor position with respect to the center of the vehicle frame.

For example if we take the rotor number 1 positioned in the front-right side of the drone's frame (see Fig. 4.2) the arm components are

$$a\vec{r}m = \begin{pmatrix} 0.338 \\ 0.195 \\ 0 \end{pmatrix}$$

measured in $[m]$.

Finally to get the velocity in a rotor centered coordinate frame this vector has to be summed with the body frame velocity components

$$\vec{V}_{rot} = \vec{V}_b + \vec{V}_{rel} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \begin{pmatrix} q\ arm_z - arm_y\ r \\ arm_x\ r - p\ arm_z \\ p\ arm_y - arm_x\ q \end{pmatrix} \tag{4.46}$$

Please pay attention that the values obtained correspond to the drone velocity plus the relative rotor velocity (i.e., positive when the vehicle is moving forward) but when calculating the induced inflow we need the values of the wind and hence a change of sign is necessary.

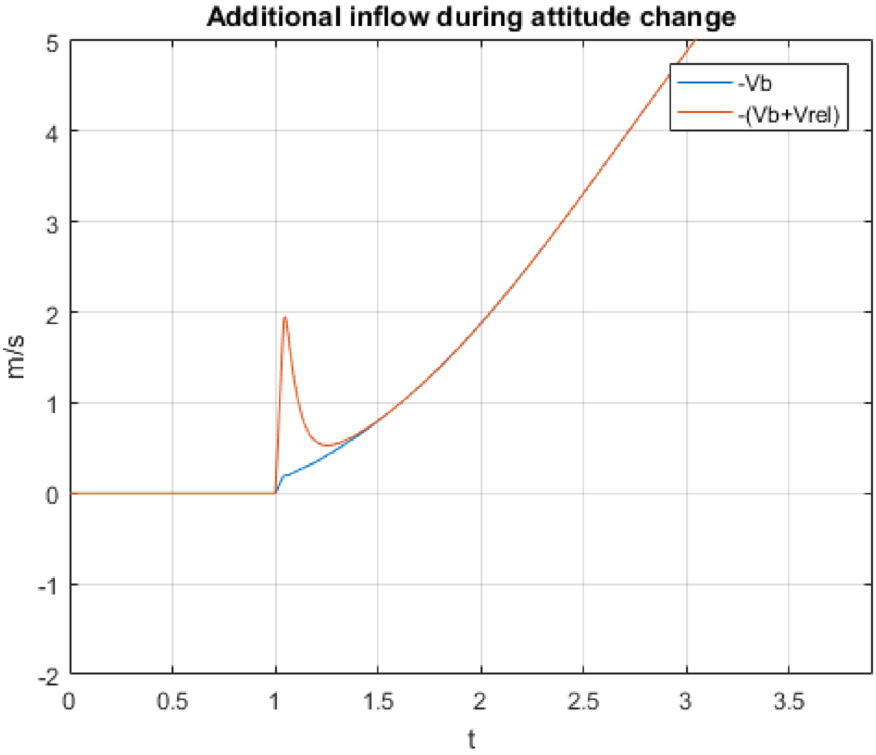

Figure 4.6: real inflow (blue curve) vs. vertical velocity in body axis (yellow curve) in motor no. 4 during a $30deg$ nose down step input

In the graph above it is depicted the effect of a $30deg$ nose down step input on the inflow in motor no. 4.

The attitude change lasts only $0.4s$ and during this time the rotor is lifting up experiencing an additional positive inflow of almost $2m/s$. When the transient is passed ($V_{rel} = 0m/s$) the rotor inflow returns equal to the vertical velocity $-w$ of the aircraft [14].

4.5.6 Gyroscopic torque from spinning rotors

Spinning rotors behaves like gyroscope, that's why when the drone is maneuvering it experiences an additive torque in a direction perpendicular with respect to the axis of rotation (i.e., $\hat{\imath}^b$ for roll and $\hat{\jmath}^b$ for pitch) and the rotor axis $\hat{k}^b$

$$\vec{M}_{gyro} = \vec{\omega}_b \times \begin{pmatrix} 0 \\ 0 \\ -\boldsymbol{J}_{rotor}\Omega_{rotor} \end{pmatrix} = \begin{pmatrix} \hat{\imath}^b & \hat{\jmath}^b & \hat{k}^b \\ p & q & r \\ 0 & 0 & -\boldsymbol{J}\Omega \end{pmatrix} \tag{4.47}$$

$$\vec{M}_{gyro} = (-q\boldsymbol{J}\Omega)\hat{\imath}^b - (-p\boldsymbol{J}\Omega)\hat{\jmath}^b \tag{4.48}$$

where $\boldsymbol{J}_{rotor}$ is the inertia tensor of the rotor obtained from the SolidWorks 3D model

$$\boldsymbol{J}_{rotor} = \begin{pmatrix} 90155.01 & 0 & 0 \\ 0 & 90155.01 & 0 \\ 0 & 0 & 180138.42 \end{pmatrix} \tag{4.49}$$

measured in $\frac{g}{mm^2}$ [15].

For example in the transient of a $30deg$ nose down maneuver, it appears an additional negative torque on axis $\hat{i}^b$ of almost $0.6Nm$.

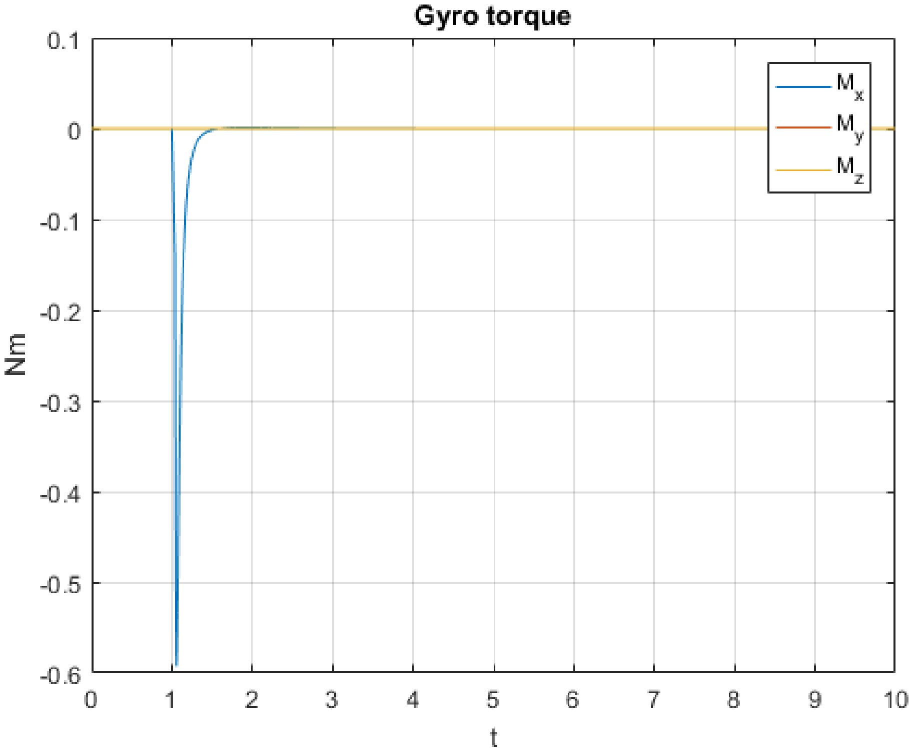

Figure 4.7: gyroscopic torque produced by the rotor of motor no. 1 during a $30deg$ nose down step input

4.6 Simplified model

Since there are no aerodynamic lifting surfaces and the velocities reached are relatively low, during the design of the simplified model the aerodynamic forces and moments as well as small other contributes, which have been considered into the more complex model, are neglected. This means that forces and moments are primarily due to gravity and motor thrust.

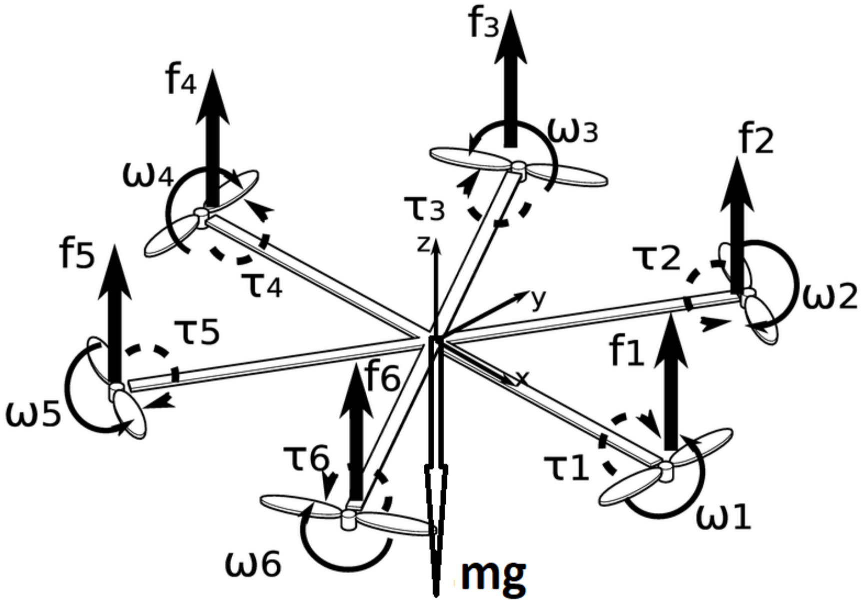

Figure 4.8: Forces and moments acting on the drone (from Researchgate website (2018) [13])

Assuming small angles (ϕ, θ, ψ) and small perturbations (p, q and r) hypothesis, equation Eq. 4.9 simplify to

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{4.50}$$

Similarly assuming that the Coriolis terms qr, pr and pq are small, equation Eq. 4.17 simplify to

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix} \tag{4.51}$$

and combining the two equations above

$$\begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix} \tag{4.52}$$

Neglecting again the Coriolis terms from Eq. 4.3

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \frac{1}{m}\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \tag{4.53}$$

rewriting the force vector as the sum of gravity (Eq. 4.23) and motor thrust components

$$\vec{f} = \vec{f}_g^b + \vec{f}_{motor} = \begin{pmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -F \end{pmatrix} \tag{4.54}$$

and with the small angle approximation, we get

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} -g\theta \\ g\phi \\ g \end{pmatrix} - \frac{1}{m}\begin{pmatrix} 0 \\ 0 \\ F \end{pmatrix} \tag{4.55}$$

Differentiating Eq. 4.1 and neglecting $\dot{\boldsymbol{R}}_b^v$

$$\frac{d}{dt}\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} = \dot{\boldsymbol{R}}_b^v \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \boldsymbol{R}_b^v \frac{d}{dt}\begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

$$\Downarrow$$

$$\begin{pmatrix} \ddot{p}_n \\ \ddot{p}_e \\ \ddot{p}_d \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} \tag{4.56}$$

using Eq. 4.55 and small angle hypothesis

$$\begin{pmatrix} \ddot{p}_n \\ \ddot{p}_e \\ \ddot{p}_d \end{pmatrix} = \begin{pmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{pmatrix} \left[\begin{pmatrix} -g\theta \\ g\phi \\ g \end{pmatrix} - \frac{1}{m}\begin{pmatrix} 0 \\ 0 \\ F \end{pmatrix} \right]$$

$$\Downarrow$$

$$\begin{pmatrix} \ddot{p}_n \\ \ddot{p}_e \\ \ddot{p}_d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} - \begin{pmatrix} \theta \\ -\phi \\ 1 \end{pmatrix} \frac{F}{m} \tag{4.57}$$

Therefore, the simplified model is given by

$$\begin{pmatrix} \ddot{p}_n \\ \ddot{p}_e \\ \ddot{p}_d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} - \begin{pmatrix} \theta \\ -\phi \\ 1 \end{pmatrix} \frac{F}{m} \tag{4.57}$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} -g\theta \\ g\phi \\ g \end{pmatrix} - \frac{1}{m} \begin{pmatrix} 0 \\ 0 \\ F \end{pmatrix} \tag{4.55}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{4.50}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix} \tag{4.51}$$

Chapter 5

Flight controller

5.1 Flight modes and limitations

The flight controller is the same for each drone's model and it is based on PID controllers. It includes two basic complementary flight modes: *Stabilize* and *Altitude Hold*.
In *Stabilize* mode, pilot's roll and pitch inputs control the lean angle of the copter which automatically level itself after they are released while yaw input controls the rate of change of the heading.
By means of the *Altitude Hold* mode the throttle on each motor is automatically controlled to maintain the current altitude until the throttle stick is between its dead zone defined by THR_DZ. Outside this mid-throttle dead zone the vehicle will descend or climb depending upon the deflection of the stick [16].
To be more precise the *Altitude Hold* mode combines two different autopilot: a *Climb-Rate Hold* autopilot and an *Altitude Hold* autopilot.

Below are listed the maximum lean angles, angular and ascending/descending velocities:

- ANGLE_MAX = $45deg$
- MAX_P = $180deg/s$
- MAX_Q = $180deg/s$
- MAX_R = $200deg/s$
- MAX_CR = $2.5m/s$
- MAX_VACC = $2.5m/s^2$

And here there are the PWM values used to scale the signal coming from the radio controller

- RC_MIN = 1094
- RC_MAX = 1933
- RC_TRIM = 1510
- RC_DZ = 20
- THR_DZ = 100

5.2 PID controller

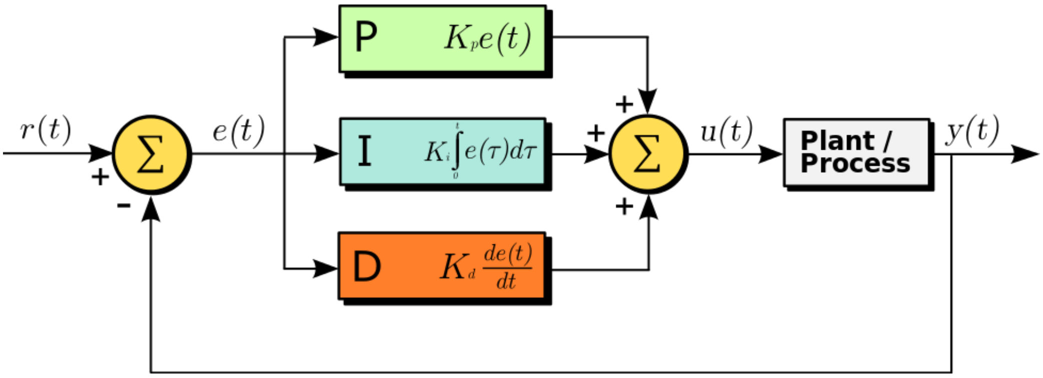

Figure 5.1: PID controller in a feedback loop (from Wikipedia website (2018) [17])

As mentioned before, the flight controller take advantage of simplicity of the Proportional Integral Derivative controller.
"A PID controller continuously calculates an error value $e(t)$ as the difference between a desired setpoint $SP = r(t)$ and a measured variable $PV = y(t)$ and applies a correction based on proportional, integral and derivative terms which give controller its name. The controller attempts to minimize the error over time by adjustment of a control variable $u(t)$ to a new value determined by a weighted sum of the control terms [17]."

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{de(t)}{dt} \tag{5.1}$$

Effects of K_P, K_I and K_D

K_P is the weight of the proportional term. As the name suggests this is the term proportional to the current value of the error and hence if it is large and positive, the control output will be proportionally large and positive. This usually converts in reducing the rise time and into reducing, but never eliminate, the steady-state error which is required in order to drive a pure proportional controller.

The integral control will eliminate this error while accelerating the movement of the process toward the setpoint. However, since the integral term is proportional to both the magnitude and the accumulated errors from the past, it may cause the overshoot of the desired setpoint.

Finally the derivative term is used to predict the system behaviour and thus improves the settling time and stability of the system. This is done by determining the slope of the error over time and multiplying it by the derivative gain K_D.

Parameter	Rise time	Overshoot	Settling time	S-s error	Stability
K_P	decrease	increase	small chance	decrease	degrade
K_I	decrease	increase	increase	eliminate	degrade
K_D	minor change	decrease	decrease	no effect	improve if K_D small

Table 5.1: effects of increasing a parameter independently (from Wikipedia website (2018) [17])

5.3 Roll, pitch and yaw

The *Stabilize* mode permits to command the roll and pitch attitude by imposing the desired angle (*phi_c* and *theta_c*) while the yaw is directly controlled with its rate (*r_c*). That's why the first proportional gains, which convert the error between desired and actual angle into rate, are present only on the roll and pitch branches.
The last block is a PID controller which converts the difference between desired and actual rates into a motor speed output.
Although Matlab has an internal algorithm capable of tuning the gains of each PID I found that in this way the dynamic behavior of the simulated drone is much more stable than the real one and less ready when it comes into compensating disturbs and small fluctuations.

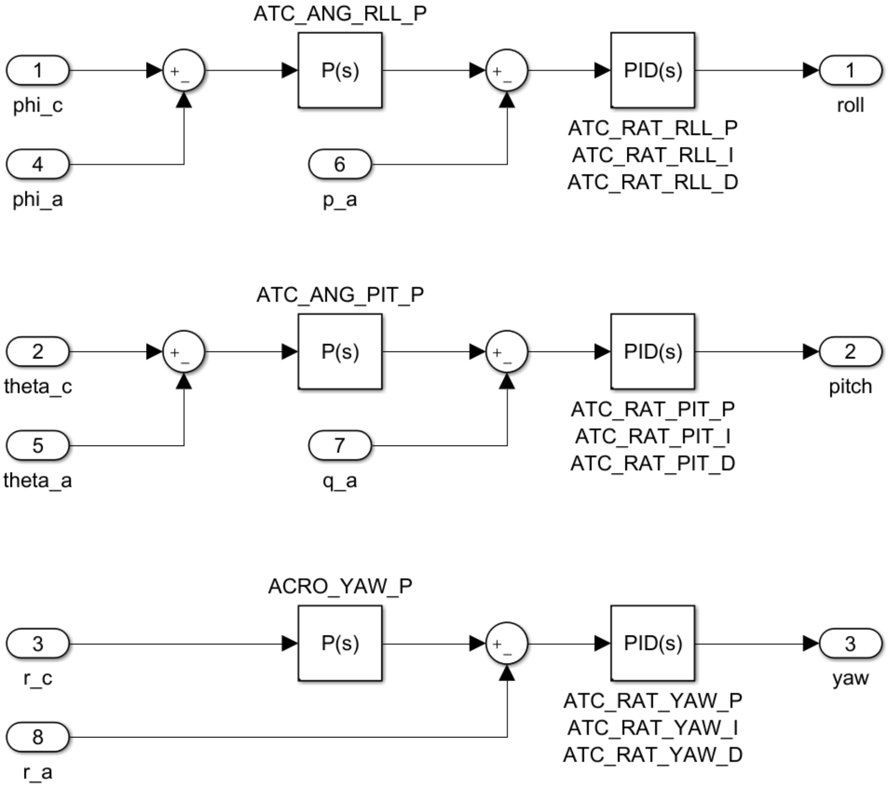

Figure 5.2: roll, pitch and yaw autopilot scheme for *stabilize* mode

For this reason the gains have been taken directly from the one used on the drone:

- ATC_ANG_RLL_P = 18
- ATC_RAT_RLL_P = 0.1921063
- ATC_RAT_RLL_I = 0.1921063
- ATC_RAT_RLL_D = 0.01537548
- ATC_RAT_RLL_FILT = 20
- ATC_ANG_PIT_P = 18
- ATC_RAT_PIT_P = 0.217424
- ATC_RAT_PIT_I = 0.217424
- ATC_RAT_PIT_D = 0.0210767
- ATC_RAT_PIT_FILT = 20
- ACRO_YAW_P = 4.5
- ATC_RAT_YAW_P = 1.532144
- ATC_RAT_YAW_I = 0.1535144
- ATC_RAT_YAW_D = 0
- ATC_RAT_YAW_FILT = 5

5.4 Altitude mode flight controller

As said before the *Altitude* mode controller is composed of an *Altitude Hold* (lower part) and a *Climb-Rate Hold* (upper part) activated by two switches. The two autopilots have a very similar structures which differs mainly in the setpoint "pursued", a reference altitude for the first and a desired climb rate in the second.

When the trigger is equal to 0 the drone will maintain its altitude even in presence of an external disturb. This is possible because the desired altitude *h_c* is saved and used as system input. In the lower part it is computed the altitude error which is then converted into a vertical speed by the proportional block POS_Z_P. In the same way the vertical speed error is calculated and multiplied by VEL_P_Z to obtain the desired vertical acceleration. Finally the vertical acceleration error is computed and transformed into motor speed output by the PID block.

If the operator wants to fly at a different altitude, his input is converted into a desired climb-rate (*cr_c*) which is then processed as described above. During these climbing or descending phases the commanded altitude keeps updating with the actual value of altitude (*h_a*) as long as the throttle is outside the altitude hold dead zone.

- POS_Z_P = 1
- VEL_Z_P = 5
- ACCEL_Z_P = 0.7
- ACCEL_Z_I = 1.4
- ACCEL_Z_D = 0
- ACCEL_Z_FILT = 20

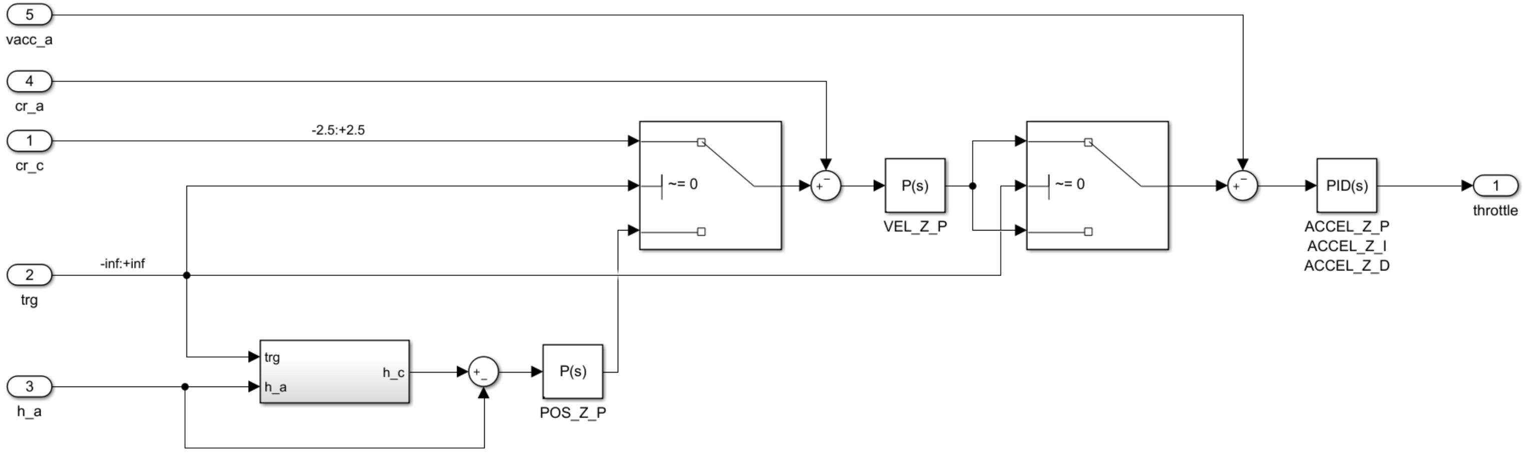

Figure 5.3: climb rate and altitude hold autopilot scheme for *altitude hold* mode

5.5 Motor mixer

The throttle that each motor requires to react correctly at the pilot's input is decided into the *Motor Mixer* block. The control logic is based on the sum of all the motor speed output coming out from the *Stabilize* and *Altitude Hold* blocks with some adjustments based on the aircraft geometry and motor position.

Being *roll*, *pitch*, *yaw* and *thrust* the commands for each maneuver, the amount of throttle in percentage for every motor is composed as:

- $m1 = pitch - roll/2 + yaw + throttle$
- $m2 = pitch + roll/2 - yaw + throttle$
- $m3 = roll + yaw + throttle$
- $m4 = -pitch + roll/2 - yaw + throttle$
- $m5 = -pitch - roll/2 + yaw + throttle$
- $m6 = -roll/2 - yaw + throttle$

Note that in all motors excluded no. 3 and 6 which are located respectively on the left and right "wing", the roll command is reduced of an half. This is a conservative choice to ensure the conservation of yaw authority even during hard bank angle maneuvers.

For example it can happen that during a positive roll (right wing down) maneuver the three left motors are at full power, while the other three at idle; it is clear that the first three rotors produce an unbalanced torque that makes the vehicle yaw. Instead with this solution, the motors no. 1 and 5 are always balanced by 2 and 4 with only a small loss the maximum roll torque (see Section 3.3).

Chapter 6

Resulsts

6.1 Outdoor tests and models comparison

The outdoor tests were done on a flight field near Forlì. The flight data were saved on an SD card stored inside the HKPilot32 flight controller and compared with the ones obtained from the simulations.

Figure 6.1: outdoor flight tests

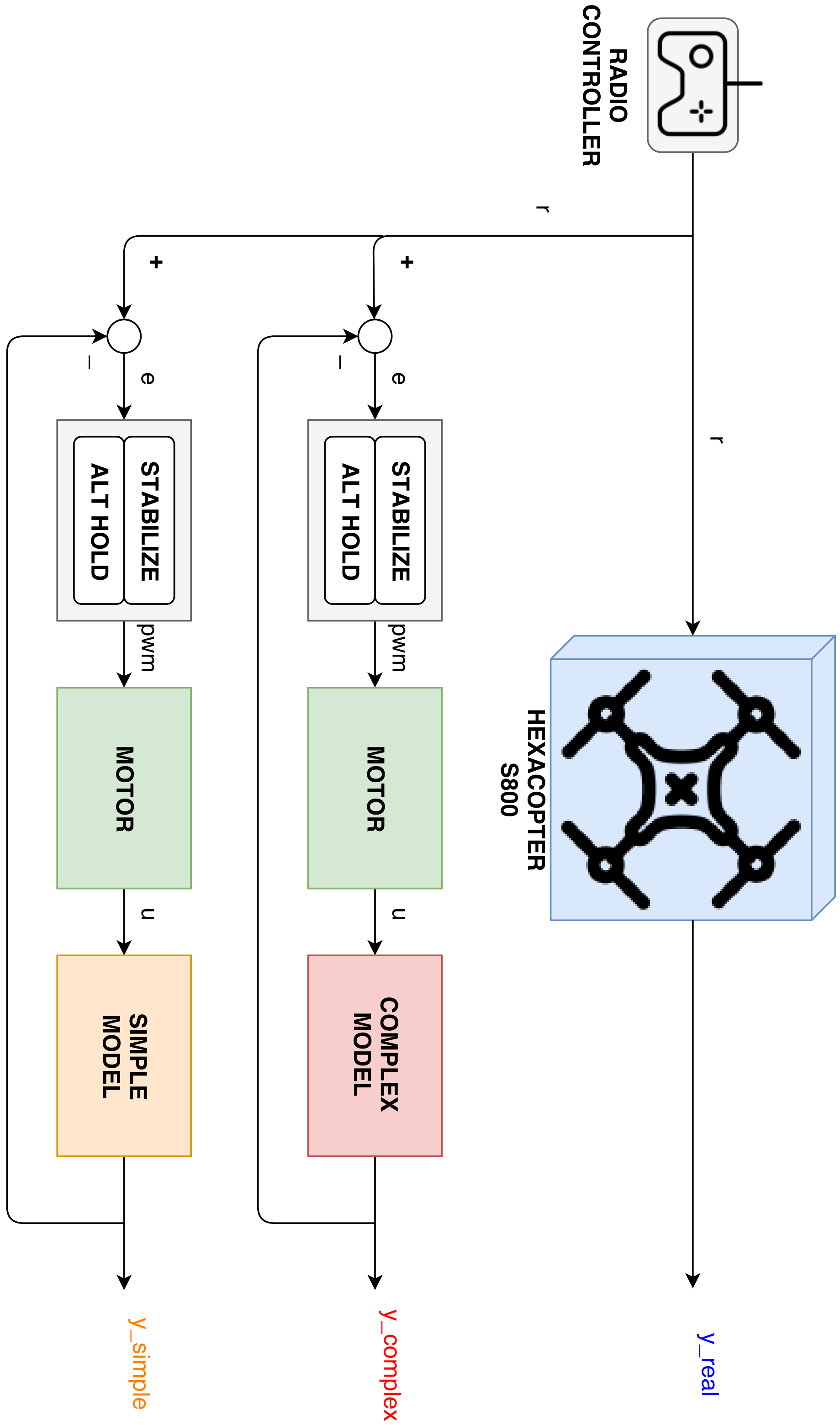

Figure 6.2: block diagram used to compare the real drone, the complex and simple mathematical model

6.2 Impulsive inputs

The first data were obtained during a preplanned sequence of small maneuvers starting from roll till yaw. During this flight mode the vehicle was transported by the wind and no input was required from the operator; that's why from a repeatability point of view, this flight mode is perfect.

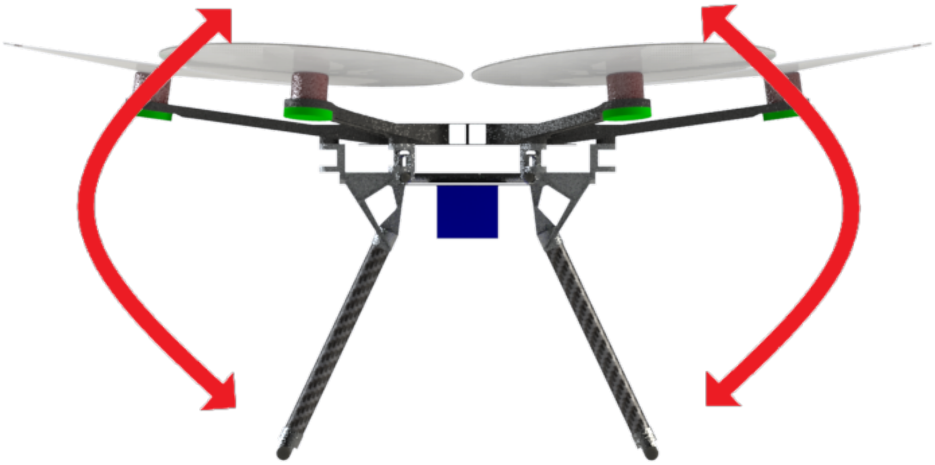

Figure 6.3: example of impulsive movement

6.2.1 Roll

Below are depicted the sequence of roll inputs in degree.
The response of the complex model is fast and very close, if not identical, to the inputs due to the "ideal" nature of the mathematical model. In fact there are no perturbations like wind or sensors' errors and the motors have the exact same response each time required.

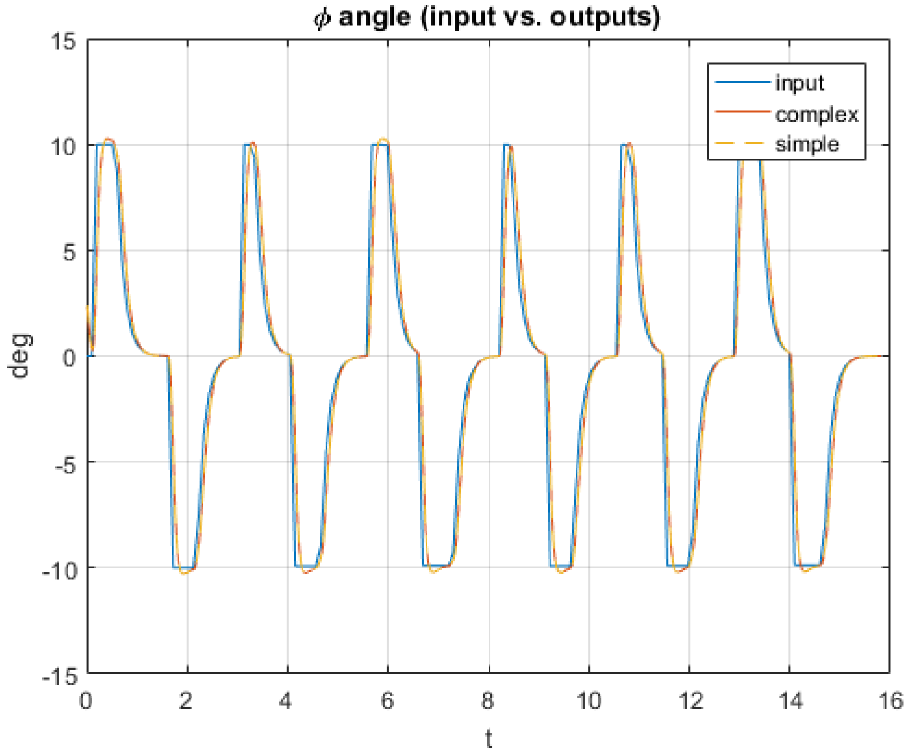

Figure 6.4: roll input in *deg* (blue curve) and simulated responses (red and yellow curves)

Below we can see the response of all the systems together.
The simple model behaves very good too, mainly because the horizontal and vertical velocity are relatively small and drag, induced velocity in forward flight, and other aspects which have been considered into the complex model, are negligible during these maneuvers.
The main difference between real and simulated results manifests into an overshoot when the flight program commands rapid roll rate changes. This is due to the non ideal motors which requires some time to accelerate and reach the desired thrust.

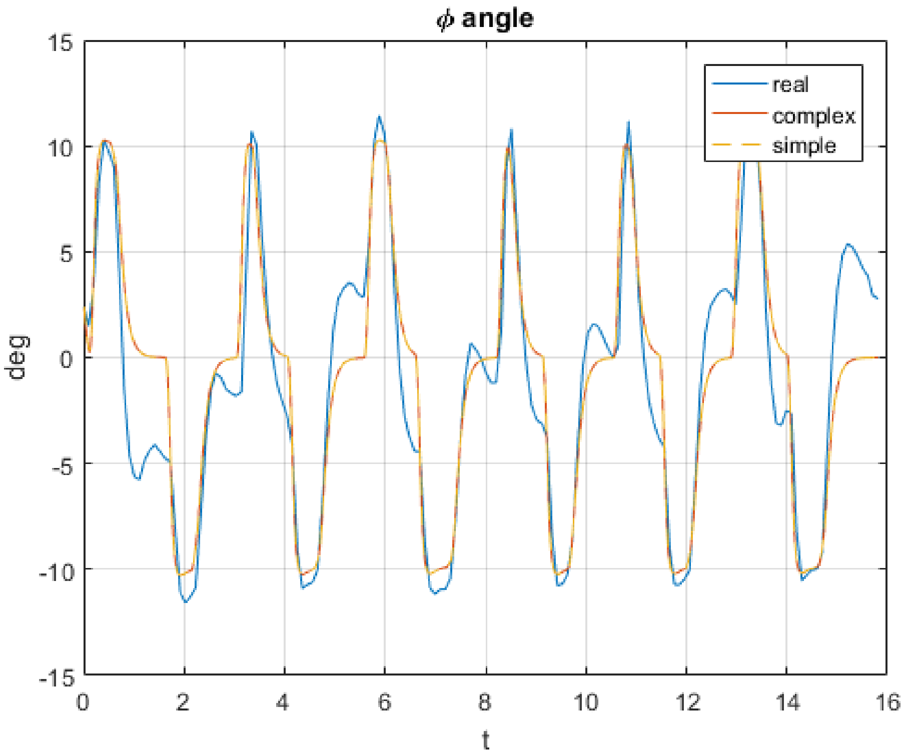

Figure 6.5: roll outputs in *deg*

6.2.2 Pitch

Immediately after the roll, the pitch test started.

The inputs are similar to the one seen in the previous section in both amplitude and shape and so it is the response of all the systems due to the quasi symmetric shape of the hexarotor.

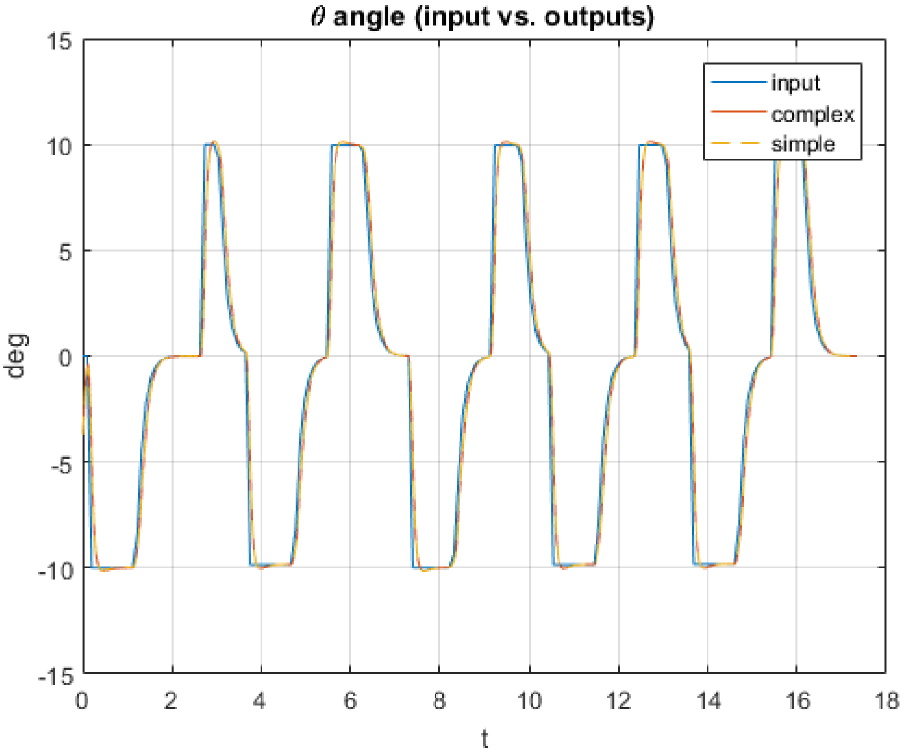

Figure 6.6: pitch input in *deg* (blue curve) and simulated responses (red and yellow curves)

Here can be seen the response of all the systems together with respect to the real one.

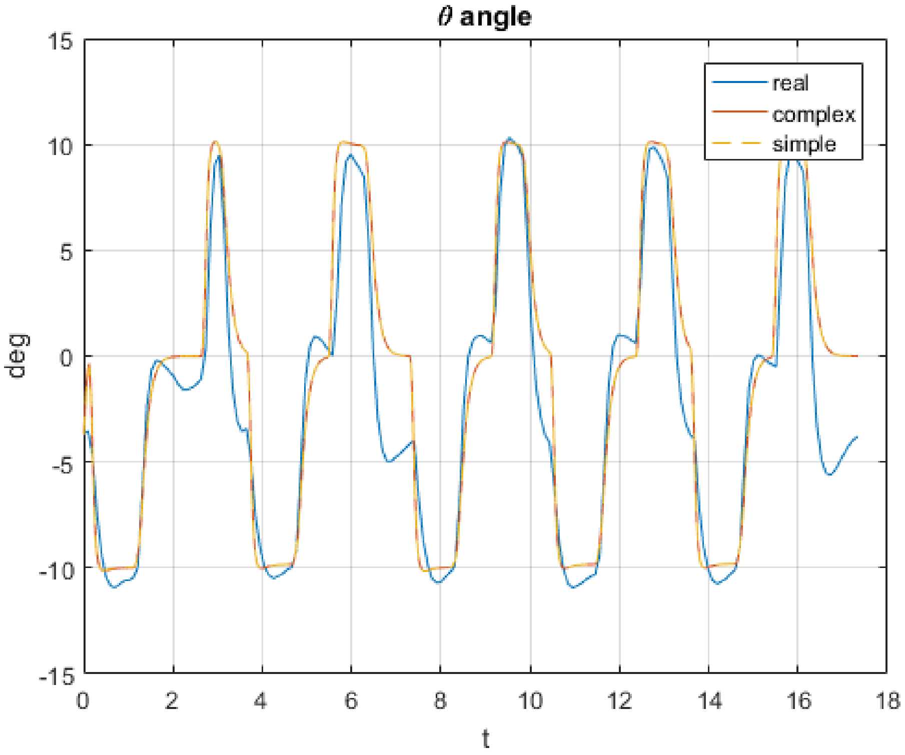

Figure 6.7: pitch outputs in *deg*

As for the roll case, the real results present the overshoot phenomenon.

6.2.3 Yaw

The last maneuver tested by the *Autotune* mode is the yaw.
Unlike before, this maneuvre is controlled by its angular rate rather than a desired angle. Through the graph below it is clear that the simulated response is very close to the the commanded and the real one even when it comes to follow a very "sharp" signal.

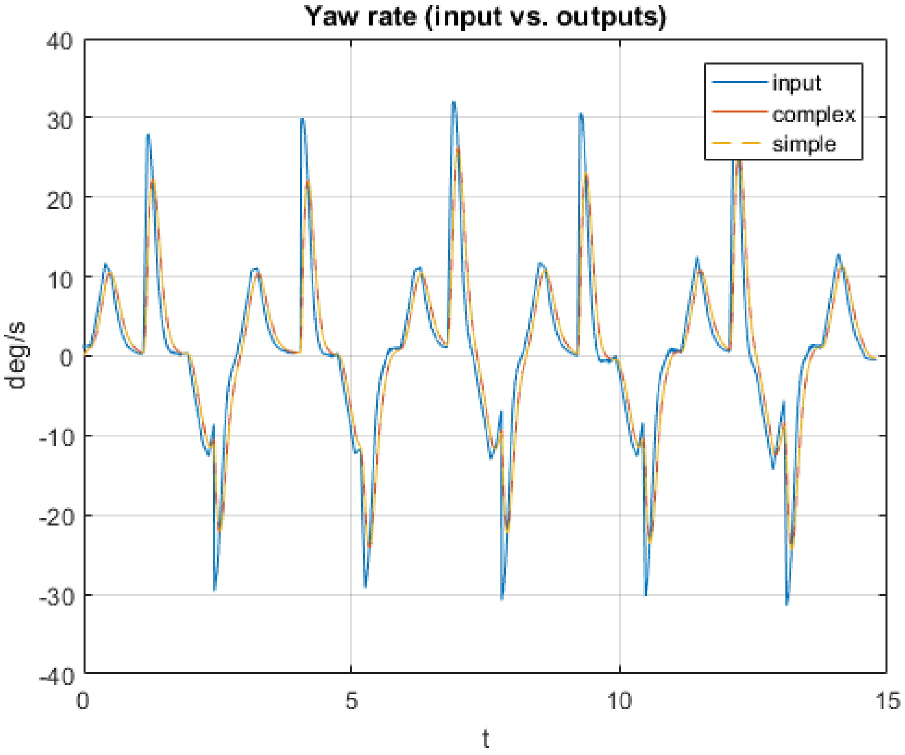

Figure 6.8: yaw input in deg/s (blue curve) and simulated responses (red and yellow curves)

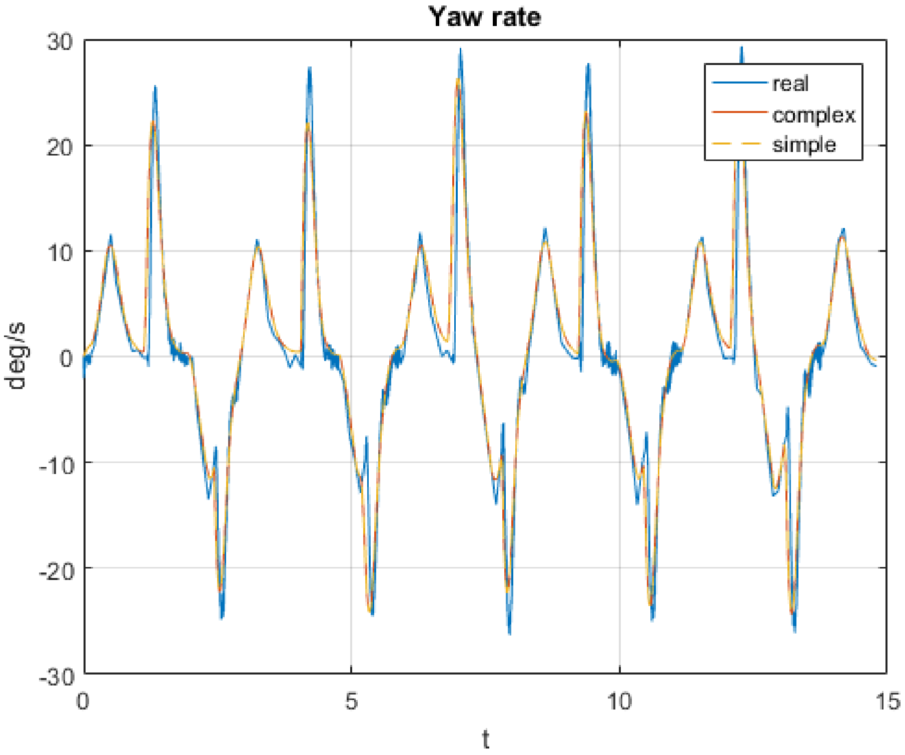

Figure 6.9: yaw outputs in deg/s

When looking at the yaw angle ψ we start to see some disparities.
In particular it seems that the angle is not able to return to the initial position of about $134deg$ and the error starts to accumulate resulting in a ”diverging” solution reaching a value of about $10deg$ after $15s$.
This can be caused by a different autopilot architecture which, in the simulated cases, wasn't provided with an heading reference block. Most probably an incorrect estimate of the torque (see section 3.2.2) and the effect that the additional inflow during the rotation are the cause of a further deterioration of the results.

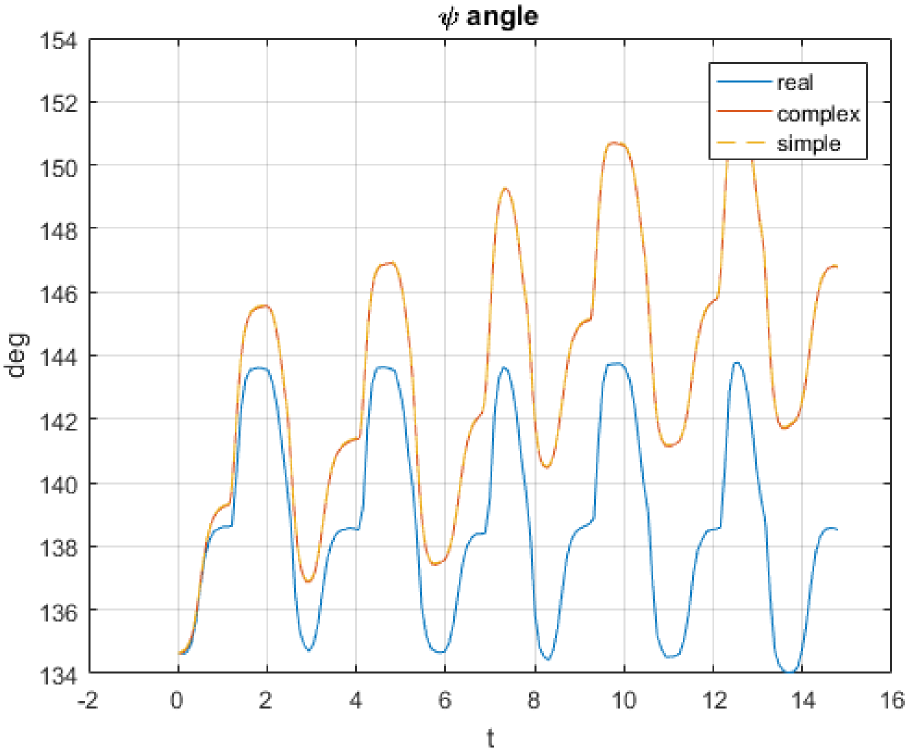

Figure 6.10: heading outputs in *deg*

6.3 Vertical flight

The next flight envelope tested is the vertical flight.
This time the pilot has full control over the drone through the two basic autopilots *Stabilize* and *Altitude hold*.

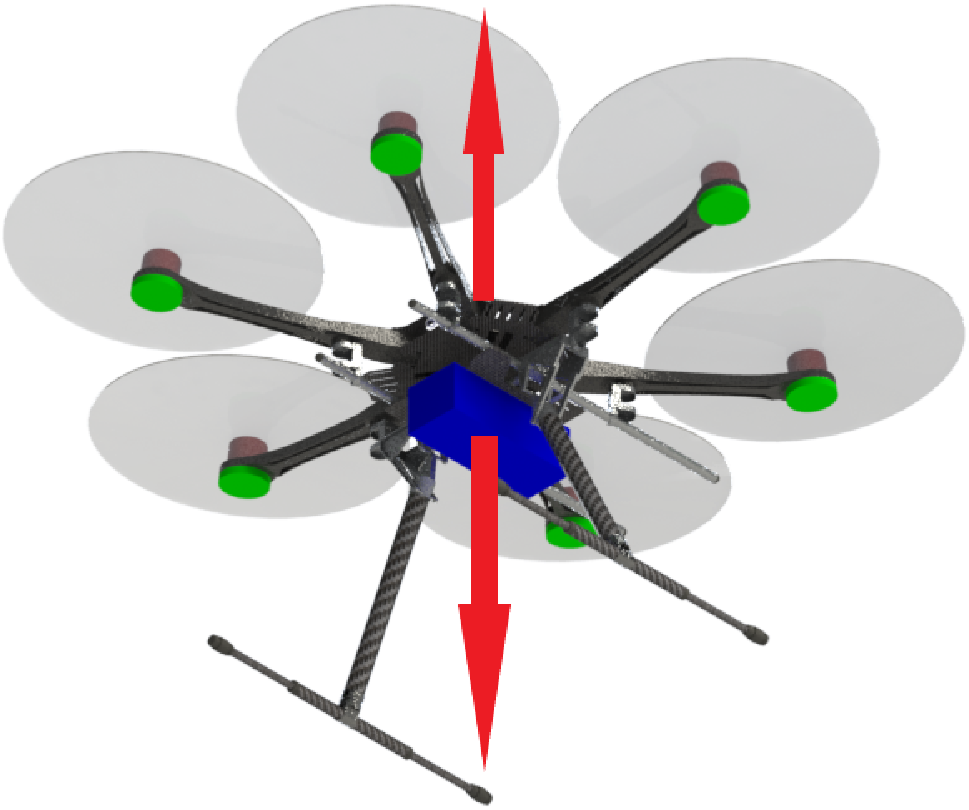

Figure 6.11: vertical flight moving pattern

In the graph below are shown the commands given with the radio controller already converted into rate of climb and the simulated responses.

The pilot inputs consists into two alternated climbs (at $1.1m/s$ and $2.5m/s$) and descents (at $-2.5m/s$ and $-1m/s$).

Compared to the previous cases, the behaviour of the complex and simple models are quite different from the input because even if the command is in climb rate, the signal is further elaborated into vertical acceleration before being processed by the PID block. This is a safety precaution used to avoid hard accelerations especially in a vertical descents where the so called *vortex ring state* may be reached, a condition in which a rotor "works" in its own wake until losing almost all its thrust.

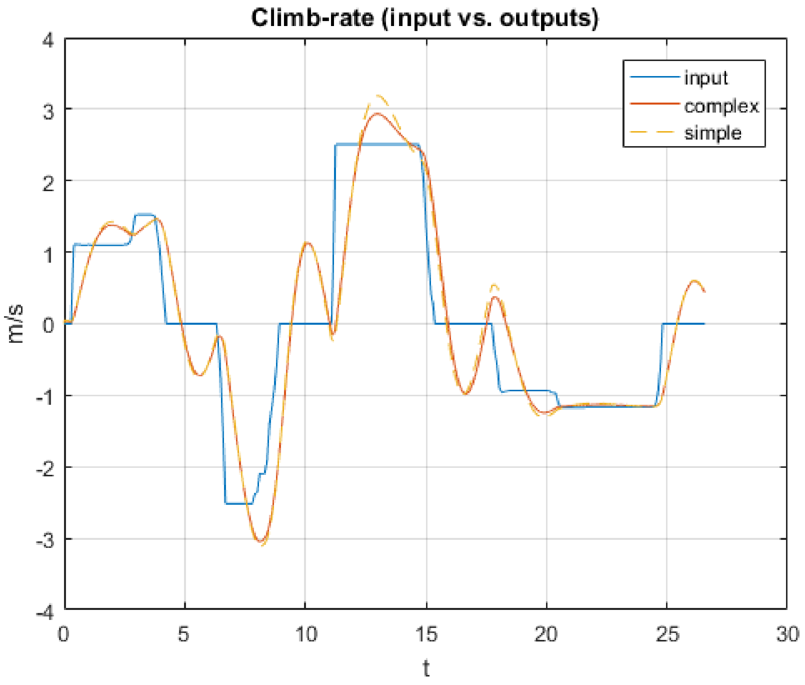

Figure 6.12: rate of climb input in m/s (blue curve) and simulated responses (red and yellow curves)

The output comparison shows a similar behaviour with an overshoot tendency of the simulated results. Moreover it appears that the complex model is slightly better than the simple one probably due to effect of the additional induced velocity through the rotor disk, which is responsible of a decrease in thrust available when climbing.

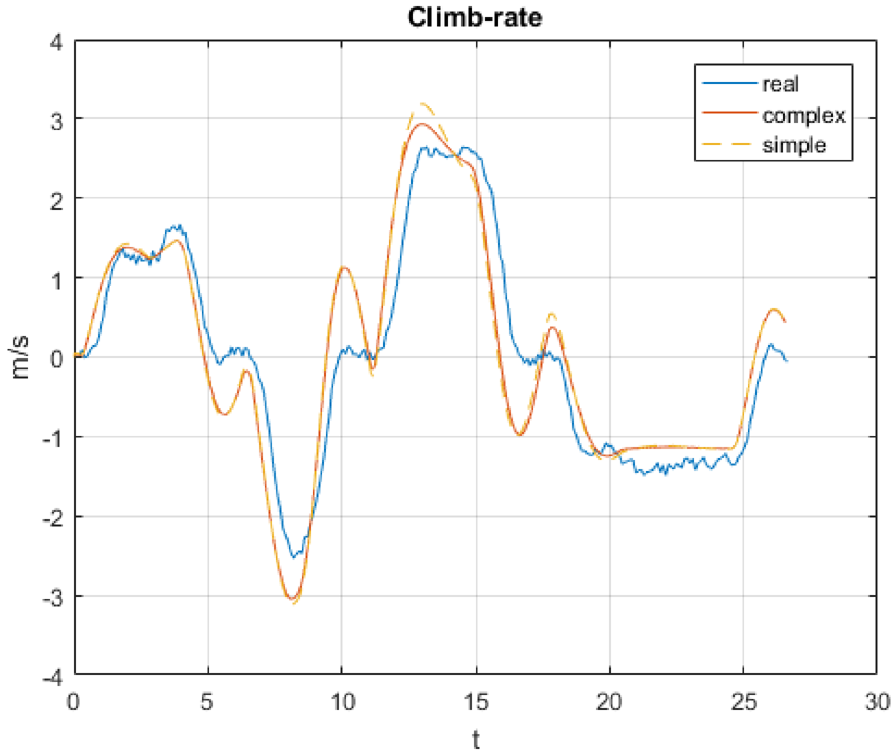

Figure 6.13: rate of climb outputs in m/s

The altitude reached during the simulations is quite similar to the real one. The small oscillations are induced by the flight controller which "switches" autopilot passing from the climb-rate to the altitude hold. Probably the "real" fight controller uses a different and more complex logic which let pass some time before the switch permitting to damp some of the oscillations.

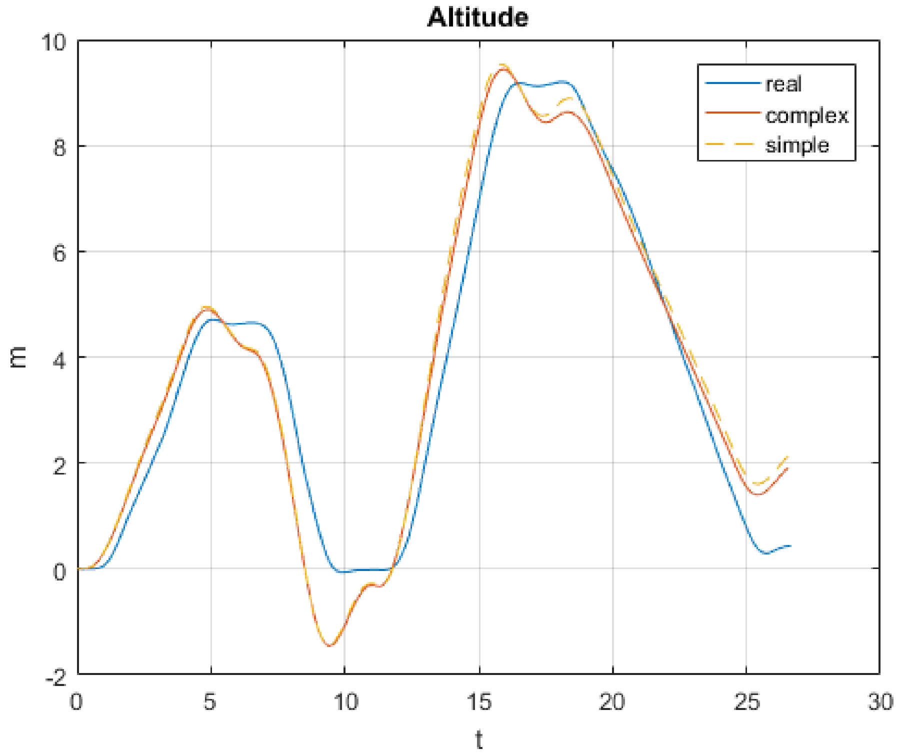

Figure 6.14: altitude in m

6.4 Horizontal flight

The last flight condition considered is a mixture of horizontal flight, heading change and a small climb component. The maneuvers follow this scheme:

1. pitch down (forward flight)
2. pitch up (backward flight)
3. heading change of about $-7deg$
4. roll right (right sideslip)
5. roll left (left sideslip)

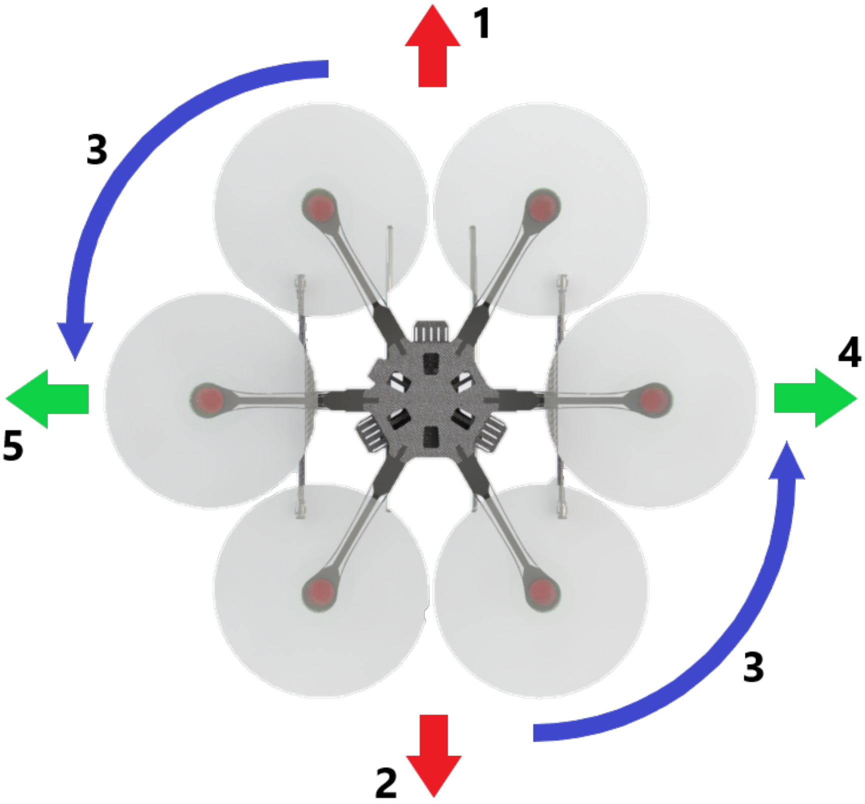

Figure 6.15: horizontal flight moving pattern

As for the impulsive inputs, the simulated results follow faithfully the real outputs. Below are listed the bank and pitch angle with respect to the input and the real output.

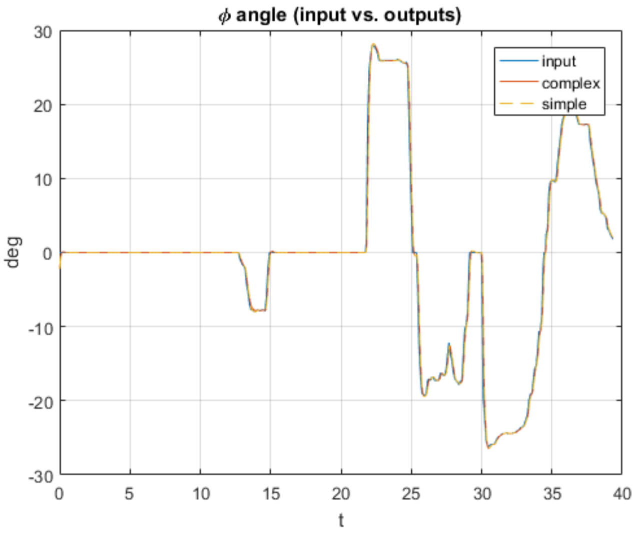

Figure 6.16: roll input in *deg* (blue curve) and simulated responses (red and yellow curves)

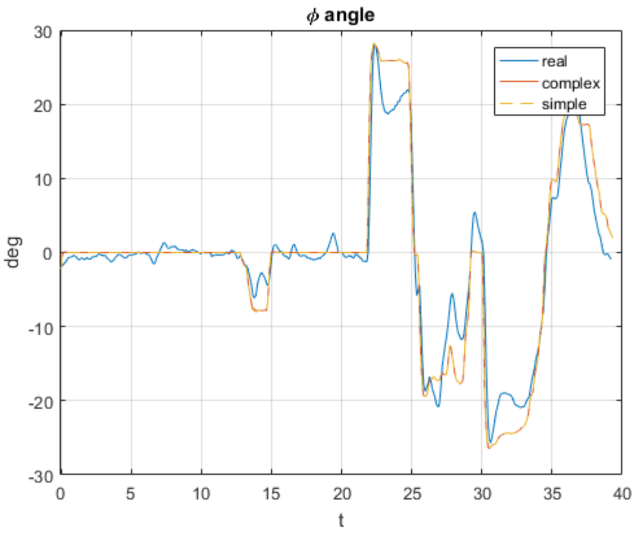

Figure 6.17: roll outputs in *deg*

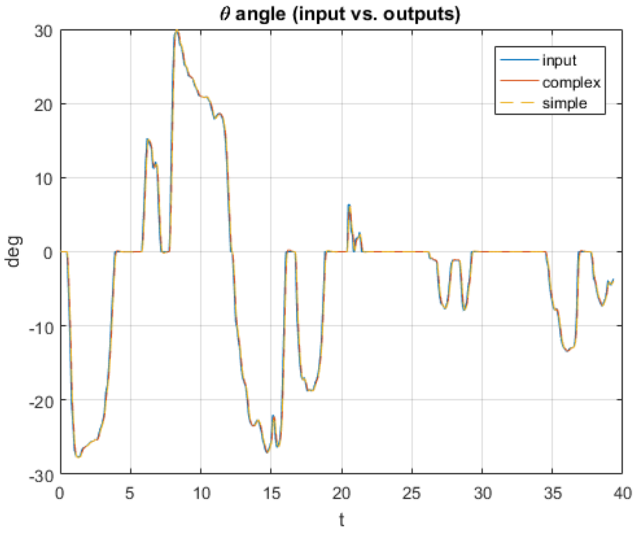

Figure 6.18: pitch input in *deg* (blue curve) and simulated responses (red and yellow curves)

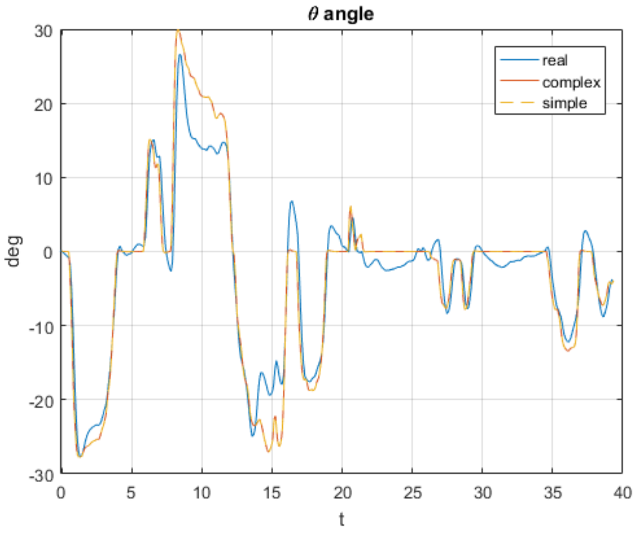

Figure 6.19: pitch outputs in *deg*

Some words have to be spent on the yaw: the first graph shows a $-11deg/s$ of yaw rate followed very closely by complex and simple model. However the graph right below shows that the real yaw rate keeps fluctuating between $-2deg/s$ and $2deg/s$ with some peaks exceeding $8deg/s$.

In my opinion the main reason falls on the non ideal motors, especially the lags which occours between the required and the supplied thrust. This combined with the attitude changes and the disturbs compensation make the motor works in non balanced conditions and hence produce yawing moments.

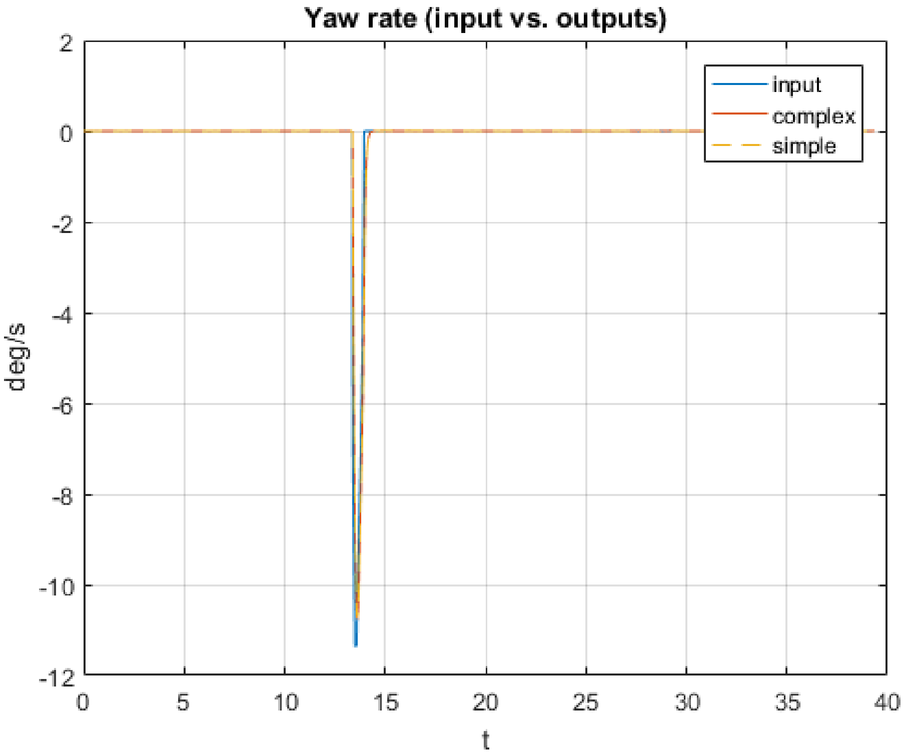

Figure 6.20: yaw input in deg/s (blue curve) and simulated responses (red and yellow curves)

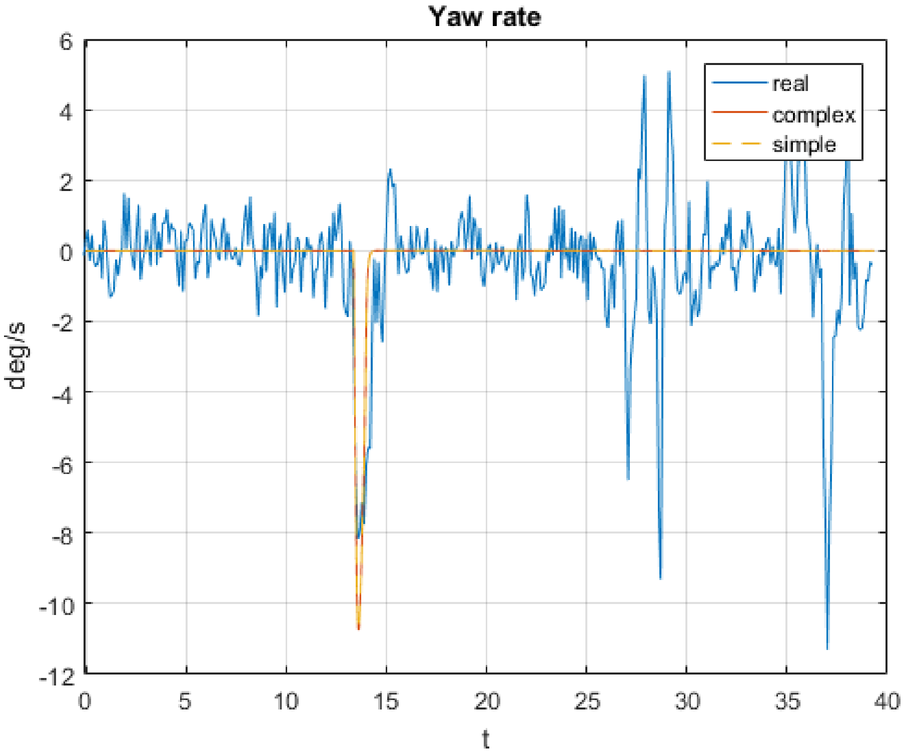

Figure 6.21: yaw outputs in deg/s

All the things written for the impulsive yaw command seems to hold true also in horizontal flight.

It is also interesting to note that the roll movement excites a yaw rate in the complex model while the simple one, due to its constitutive assumptions, seems to not ”feel” this effect.

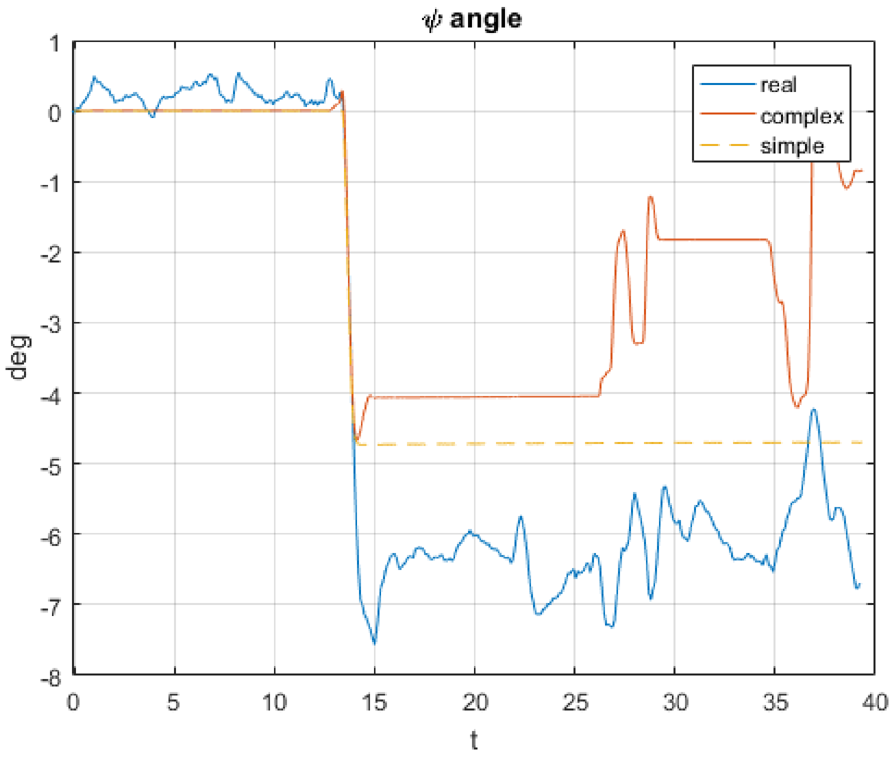

Figure 6.22: heading outputs in *deg*

The same seems to manifest into the desired climb rate which, respect to the vertical flight, this time corresponds only to a small value with respect to the full-scale and its precision seems to be ”sacrificed” in order to bring the drone in the desired attitude as soon as possible. On the other hand, the excessive readiness of the ideal motors produces climb rates excursions of bigger amplitude which yield to a more oscillatory altitude.
Instead the simple model with the Coriolis terms neglected and the less effective propellers that do no benefit of the lifting effect of the forward flight, is much less affected by all these small perturbations resulting in a lower altitude.

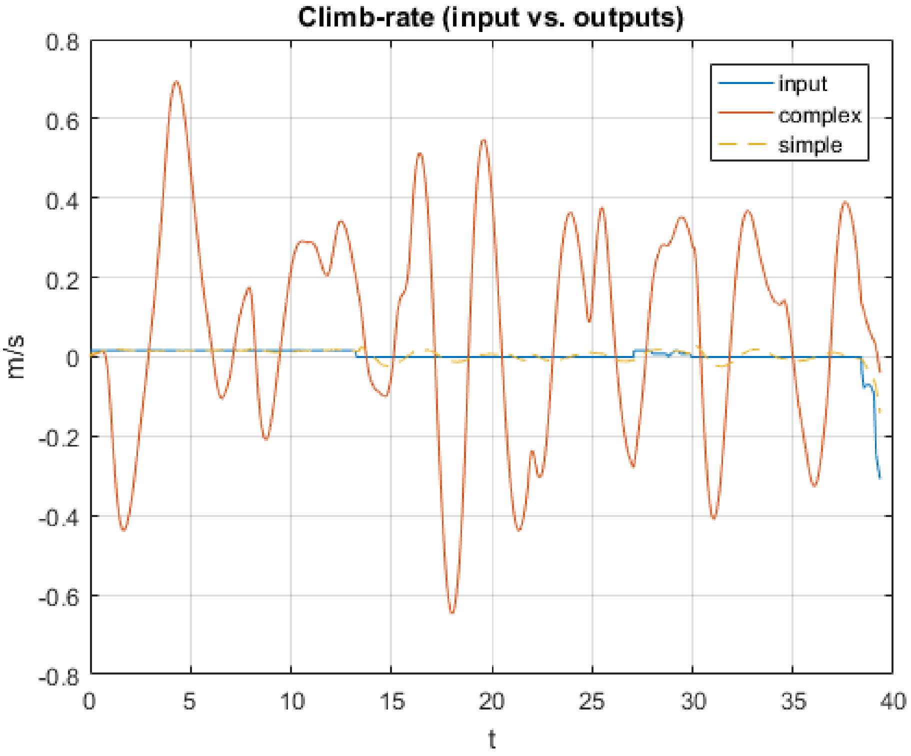

Figure 6.23: rate of climb input in m/s (blue curve) and simulated responses (red and yellow curves)

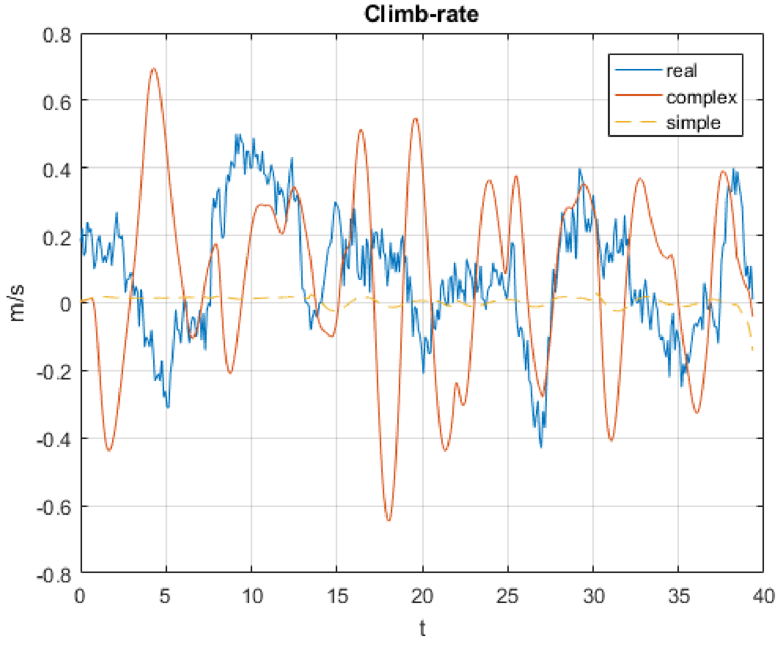

Figure 6.24: rate of climb outputs in m/s

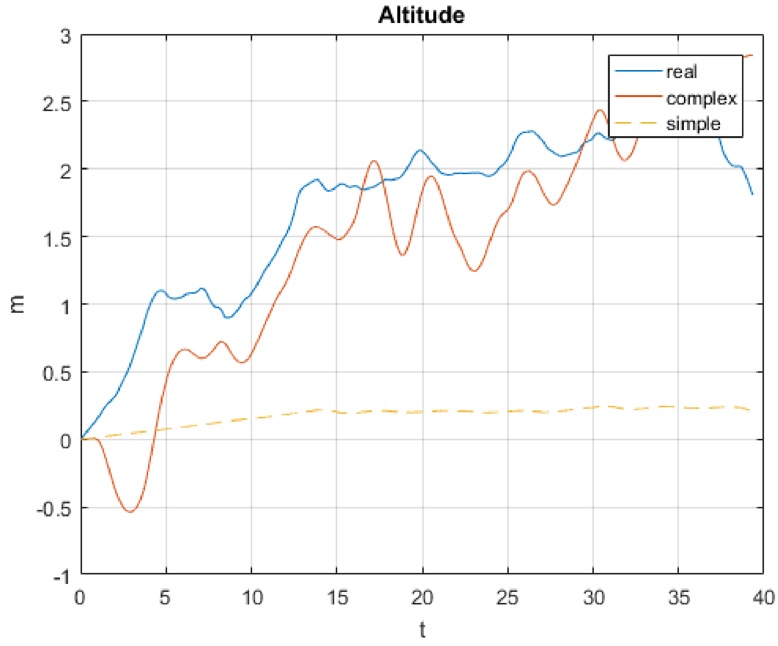

Figure 6.25: altitude outputs in m

At last, when looking at the horizontal velocity, it seems that drag plays an important role into building an error between the complex and simple model. This is confirmed by the entity of the error which seems to increase as the velocity increases.

Despite this, also the complex model shows a very different velocity with respect to the real one detected by the integrated GPS. This could be caused by the wind, infact during the whole time it is present an offset that leaves unchanged the general velocity trend.

Furthermore, after the heading change command at around $14s$, this error appears to suddenly change its amplitude until the end of the control horizon.

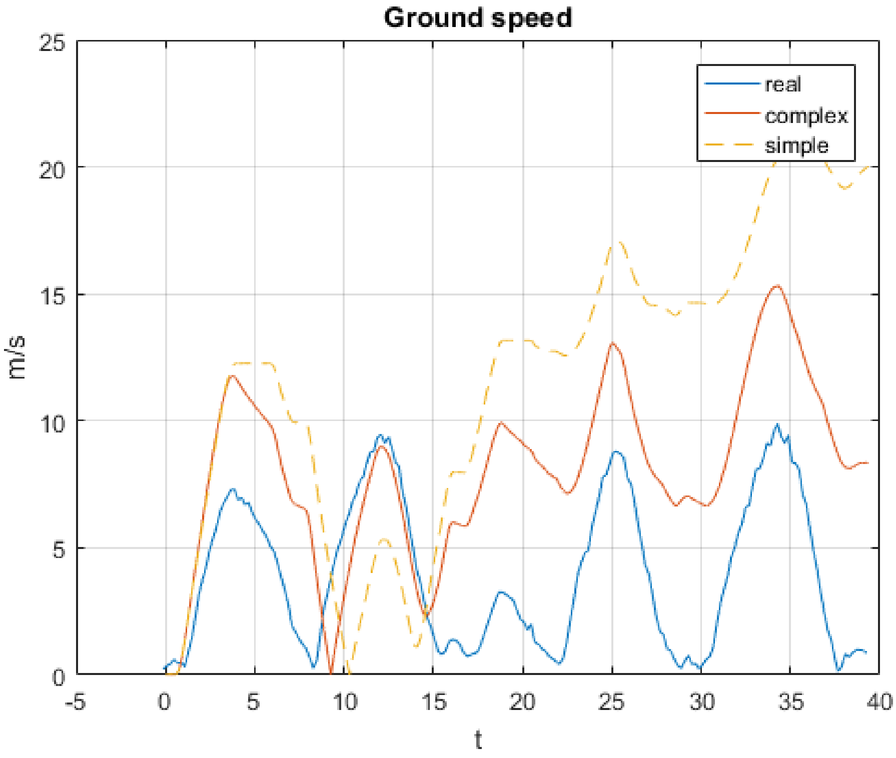

Figure 6.26: velocity outputs in m/s

Although this appears to be a bad result, it has to be reminded that the ground speed it is not used as reference but instead obtained as the result of a combinations of attitude and motor thrust.

Conclusions

The results obtained in Section 6 shows that the mathematical models hold when trying to follow a reference value but start to fail when it comes to estimate variables which are not directly controlled or simulate secondary effects produced by a non ideal behaviour; for example the horizontal velocity or the angular rate oscillations induced by the motors lag. This is mainly due to some simplifications and aspects not modelled in their entirety.
In any conditions the complex model shows a more faithful behavior with respect the simple one, demonstrating that this is the right direction to take.
In my opinion the firsts aspects that need to be improved are the implementation of motors lag and a better estimation of their torque, which could be easily done with a test bench updated with a multi-axis load cell. Secondly, the estimation of the wind components by means of a Pitot tube would greatly increase the results quality.
One could also implement a *gain scheduling* logic into the flight controllers or design new and more complex autopilots.
I think that this dissertation could represent a good starting point into evaluating the flying qualities of a multi-rotor aircraft and I'm sure that its precision will greatly benefits from future improvements capable of thinning the disparities between the real thing and the ideal model.

Appendix A

Coordinate frames

A.1 The Inertial frame $\mathcal{F}^i$

It is an earth fixed coordinate system with origin at the defined home location. $\hat{i}^i$ is directed North, $\hat{j}^i$ is directed East, and $\hat{k}^i$ is directed toward the center of the earth.

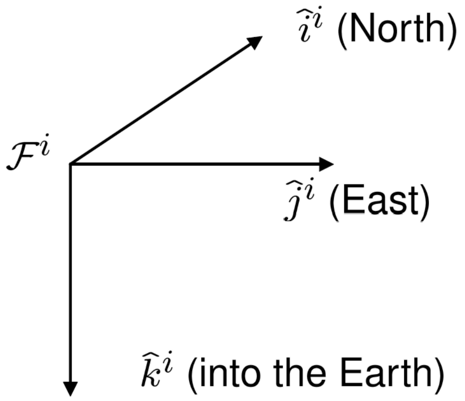

Figure A.1: inertial coordinate frame (from Randal W. Beard (2008), p. 6 [12])

A.2 The vehicle frame $\mathcal{F}^v$

The origin of the vehicle frame is at the center of mass of the aircraft but all the axes are aligned with the axis of the inertial frame $\mathcal{F}^i$.

A.3 The Vehicle-1 frame $\mathcal{F}^{v1}$

The vehicle-1 frame is identical to $\mathcal{F}^v$ positively rotated about $\hat{k}^v$ by the yaw angle ψ so that, with θ and ϕ equal to zero, $\hat{\imath}^{v1}$, $\hat{\jmath}^{v1}$ and $\hat{k}^{v1}$ would respectively point out the nose, the right wing and toward the center of the earth. The rotation matrix between $\mathcal{F}^v$ and $\mathcal{F}^{v1}$ is

$$R_v^{v1}(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{A.1}$$

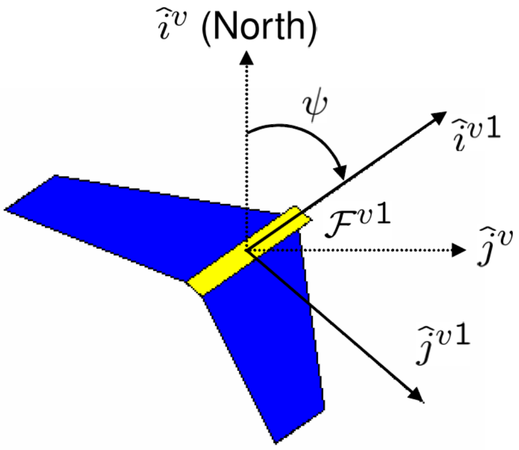

Figure A.2: vehicle-1 coordinate frame (from Randal W. Beard (2008), p. 7 [12])

A.4 The Vehicle-2 frame $\mathcal{F}^{v2}$

The origin of the vehicle-2 frame is again the center of gravity and is obtained by rotating the vehicle-1 frame in a right-handed rotation about the $\hat{j}^{v1}$ axis by the pitch angle θ. As before, if ϕ is null, $\hat{\imath}^{v2}$ point out the nose of the airframe and $\hat{j}^{v2}$ points out of the right wing while $\hat{k}^{v2}$ points out the belly. The rotation matrix between $\mathcal{F}^{v1}$ and $\mathcal{F}^{v2}$ is

$$R_{v1}^{v2}(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \tag{A.2}$$

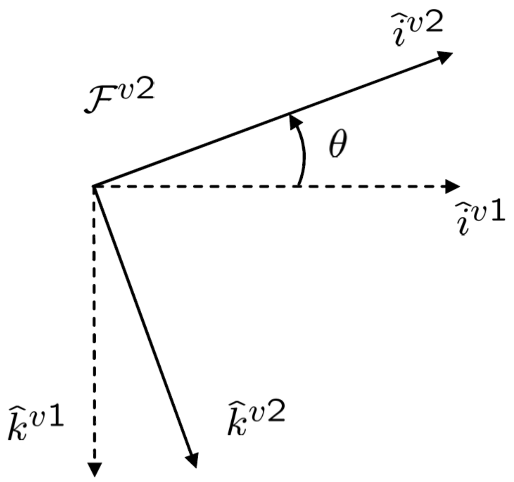

Figure A.3: vehicle-2 coordinate frame (from Randal W. Beard (2008), p. 8 [12])

A.5 The body frame $\mathcal{F}^b$

The body frame is obtained by rotating the vehicle-2 frame in a right handed rotation about $\hat{i}^{v2}$ by the roll angle ϕ. As result of these three subsequent rotations, the origin of $\mathcal{F}^b$ is always the center of gravity, $\hat{i}^b$ points out the nose, $\hat{j}^b$ points out of the right wing and $\hat{k}^b$ points out the belly. The rotation matrix between $\mathcal{F}^{v2}$ and $\mathcal{F}^b$ is

$$R_{v2}^b(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \tag{A.3}$$

The transformation from the vehicle frame to the body frame is given by

$$\begin{aligned} R_v^b(\phi,\theta,\psi) &= R_{v2}^b(\phi) R_{v1}^{v2}(\theta) R_v^{v1}(\psi) \\ &= \begin{pmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{pmatrix} \end{aligned} \tag{A.4}$$

whit $s = \sin$ and $c = \cos$.

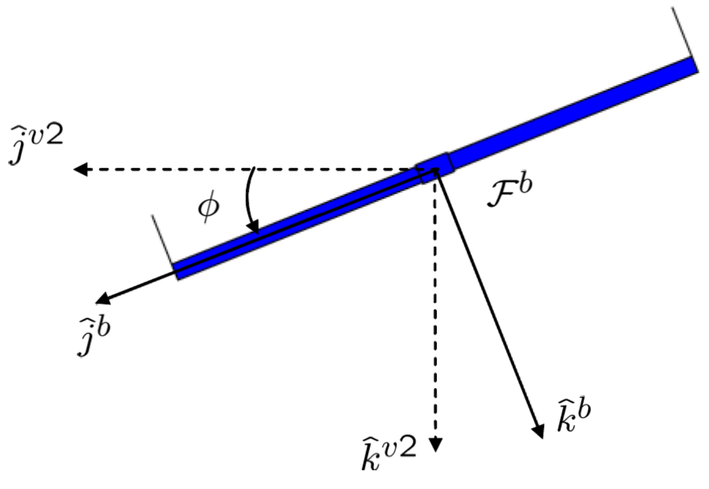

Figure A.4: body coordinate frame (from Randal W. Beard (2008), p. 9 [12])

A.6 Equation of Coriolis

Suppose to have an earth fixed coordinate system $\mathcal{F}^i$ and a body frame $\mathcal{F}^b$ centered on the aircraft center of gravity. Now assume that the vector $\vec{p}$ is moving in $\mathcal{F}^b$ and that $\mathcal{F}^b$ is rotating and translating with respect to $\mathcal{F}^i$. The time derivative of $\vec{p}$ as seen from frame $\mathcal{F}^i$ with small angle approximation is given by

$$\frac{d}{dt_i}\vec{p} = \frac{d}{dt_b}\vec{p} + \vec{\omega}_b \times \vec{p} \tag{A.5}$$

where $\vec{\omega}_b$ is the angular velocity of the aircraft with respect to the inertial frame.

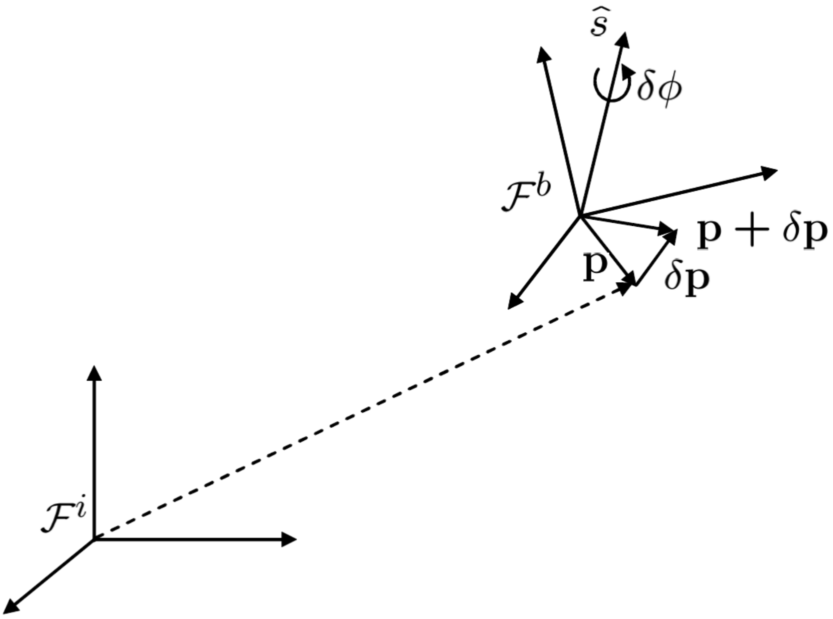

Figure A.5: derivation of the equation of Coriolis (from Randal W. Beard (2008), p. 10 [12])

Appendix B

Rotor momentum theory

B.1 Conservation laws of aerodynamics

Let the control volume surrounding the rotor and its wake have a surface area S and let $d\vec{S}$ be the unit normal area vector, which by convention always points out of the control volume. The conservation of fluid mass into this finite control volume can be written as

$$\iint_S \rho\vec{V}\cdot d\vec{S} = 0 \tag{B.1}$$

where $\vec{V}$ is the local velocity and ρ is the density of the fluid. This equation states that the mass flow into the control volume must equal the mass flow out of the control volume.

Similarly, an equation governing the conservation of fluid momentum can be written as

$$\vec{F} = \iint_S p\ d\vec{S} + \iint_S (\rho\vec{V}\cdot d\vec{S})\vec{V} \tag{B.2}$$

As Glauert (1935) has demonstrate, for an unconstrained flow the net pressure force on the fluid inside the control volume is zero. Therefore, the net force on the fluid reduces only to the second double integral which describes the rate of change with time of the fluid momentum across the control surface. Because the force on the fluid is supplied by the rotor, by the Newton's third law the fluid must exert an equal and opposite force on the rotor. This force is the rotor thrust T.

Finally, an equation governing the conservation of energy in the flow can be written as

$$W = \iint_S \frac{1}{2}(\rho\vec{V} \cdot d\vec{S})|\vec{V}|^2 \tag{B.3}$$

This equation states that the work done on the fluid by the rotor manifests as a gain in kinetic energy of the fluid in the rotor slipstream per unit time [10].

B.2 Induced velocity in hover

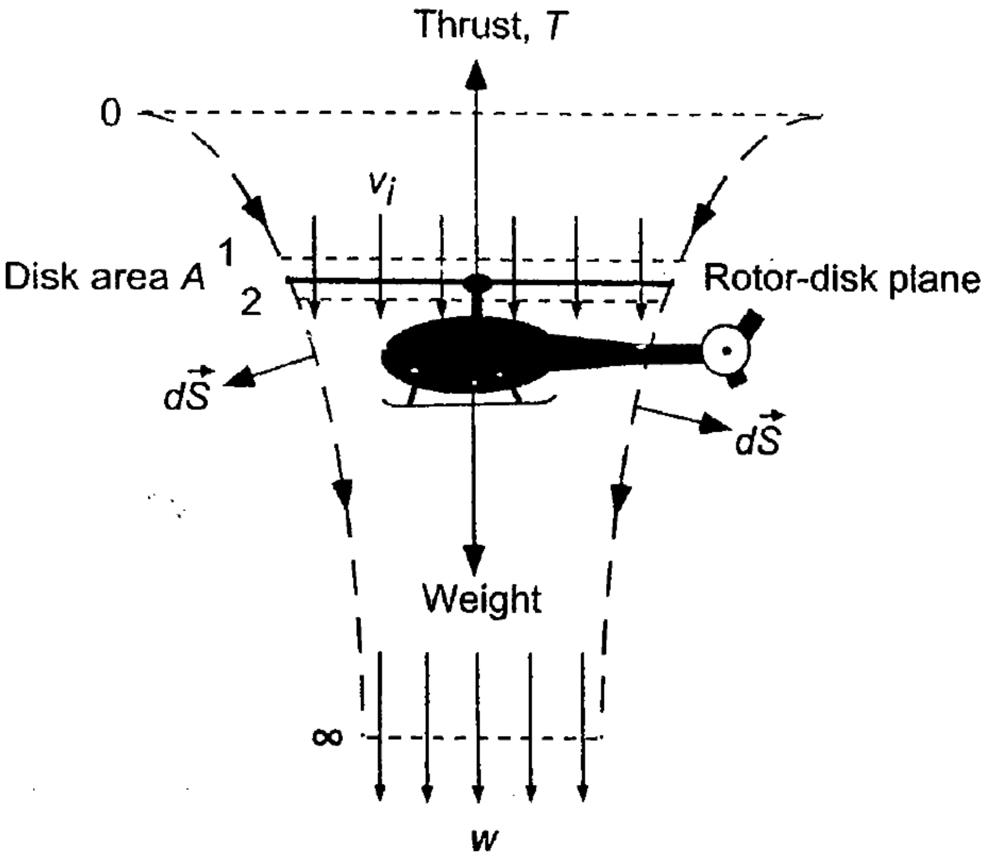

Figure B.1: flow model for momentum theory analysis of a rotor in hovering flight (from J. Gordon Leishman (2006), p. 61 [10])

These general equations of fluid mass, momentum and energy conservation may now be applied to the specific problem of a hovering rotor.

In Fig. B.1 are shown four important cross sections:

- cross section 0 denote the plane far upstream in the rotor where (in the hovering case) the fluid is quiescent and $v_0 = 0$
- cross section 1 is the plane just above the rotor disk
- cross section 2 is the plane just below the rotor disk
- cross section ∞ is the plane at *vena contracta* completely immersed in the rotor wake

At the rotor plane the flow velocity is called induced velocity v_i while in the far wake the velocity is defined as w. From the assumption that the flow is quasi-steady and by the principle of conservation of mass, the mass flow rate $\dot{m}$ must be constant within the boundaries of the control volume, therefore

$$\dot{m} = \iint_\infty \rho\vec{V} \cdot d\vec{S} = \iint_2 \rho\vec{V} \cdot d\vec{S} \tag{B.4}$$

and the one dimensional and incompressible flow assumption reduces this equation to

$$\dot{m} = \rho A_\infty w = \rho A_2 v_i = \rho A v_i \tag{B.5}$$

The rotor thrust is equal and opposite to the force on the fluid so the Eq. B.2 became

$$-\vec{F} = T = \iint_\infty \rho(\vec{V} \cdot d\vec{S})\vec{V} - \iint_0 \rho(\vec{V} \cdot d\vec{S})\vec{V} \tag{B.6}$$

Because in hovering flight the flow velocity at plane 0 is quiescent, the second term on the right-hand side of the above equation is null therefore

$$T = \iint_\infty \rho(\vec{V} \cdot d\vec{S})\vec{V} = \dot{m}w \tag{B.7}$$

From Eq. B.3 the work done on the rotor is equal to the gain in energy of the fluid per unit time. The work done per unit time, or the power consumed by the rotor is Tv_i so

$$Tv_i = \iint_\infty \frac{1}{2}\rho(\vec{V} \cdot d\vec{S})\vec{V}^2 - \iint_0 \frac{1}{2}\rho(\vec{V} \cdot d\vec{S})\vec{V}^2 \tag{B.8}$$

Again in hover $v_0 = 0$ and the second term on the right-hand side of the above equation is zero

$$Tv_i = \iint_\infty \frac{1}{2}\rho(\vec{V} \cdot d\vec{S})\vec{V}^2 = \frac{1}{2}\dot{m}w^2 \tag{B.9}$$

Finally from Eq. B.7 and Eq. B.9 we get

$$v_i = \frac{1}{2}w \tag{B.10}$$

which is a simple relationship between the induced velocity in the plane of the rotor and the velocity in the vena contracta. Substituting Eq. B.10 into Eq. B.7 we obtain the thrust as a function of induced velocity

$$T = \dot{m}w = \dot{m}(2v_i) = 2(\rho a v_i)v_i = 2\rho A v_i^2 \tag{B.11}$$

and solving for v_i gives

$$v_h \equiv v_i = \sqrt{\frac{T}{2\rho A}} \tag{B.12}$$

[10].

Bibliography

[1] Wikipedia website (2018)
Unmanned aerial vehicle
https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle

[2] Wikipedia website (2018)
PX4 autopilot
https://en.wikipedia.org/wiki/PX4_autopilot

[3] DJI website (2018)
https://www.dji.com/spreading-wings-s800-evo/feature

[4] Ardupilot website (2018)
http://ardupilot.org/copter/docs/
common-pixhawk-wiring-and-quick-start.html

[5] Pixhawk website (2018)
https://pixhawk.org/modules/pixhawk

[6] PX4 website (2018)
https://docs.px4.io/en/airframes/airframe_reference.html

[7] Wikipedia website (2018)
Pulse-width modulation
https://en.wikipedia.org/wiki/Pulse-width_modulation

[8] Wikipedia website (2018)
Pulse-position modulation
https://en.wikipedia.org/wiki/Pulse-position_modulation

[9] Pabr.org website (2018)
http://www.pabr.org/pxarc/doc/pxarc.en.html

[10] J. Gordon Leishman (2006)
Principles of Helicopter Aerodynamics
USA, Cambridge University Press
Second edition, from p. 55 to 100

[11] UIUC Applied Aerodynamics Group (2018)
UIUC Airfoil Coordinates Database website
http://m-selig.ae.illinois.edu/ads/coord_database.html

[12] Randal W. Beard (2008)
Quadrotor Dynamics and Control Rev 0.1
All Faculty Publications. Paper 1325
http://scholarsarchive.byu.edu/facpub/1325

[13] Artale V., Ricciardello A., Milazzo C. (2013)
Mathematical modeling of hexacopter
Applied Mathematical Sciences. 7. 4805-4811. 10.12988/ams.2013.37385.

[14] Ferrarese G. (2015)
Dynamics and control issues of multi-rotor platforms
PhD thesis at University of Bologna, from p. 36 to 40

[15] Ferrarese G., Gatti M., Giulietti F.
Modeling and Control of a Quadrotor with enhanced Endurance and Maneuverability
Paper of Università of Bologna, p. 4

[16] Ardupilot website (2018)
http://ardupilot.org/copter/docs/flight-modes.html

[17] Wikipedia website (2018)
PID controller
https://en.wikipedia.org/wiki/PID_controller

Acknowledgments

I would first like to thank my thesis advisor Prof. Fabrizio Giulietti of the University of Bologna which has always supported me and believed in my abilities. Thanks to his efforts and professionality, I was able to work on a very interesting and actual topic that allowed me to deepen my knowledge and grown as engineer. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.
I would also like to thank the professor assistant Dr. Gianluca Rossetti for his disposability and patience; the door to the research laboratory was always open whenever I ran into a trouble spot or had a question about my research or writing.
I must express my very profound gratitude to my parents Giampaolo and Fiorella for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis, this accomplishment would not have been possible without them. Thank you.
I would also like to thank my amazing family for the love and warmth I have gotten over the years.
Last, but not least I must to thank my girlfriend and friends, your love, laughter and company have kept me smiling and inspired.

Matteo Bernabé