ALMA MATER STUDIORUM-UNIVERSITÀ DI BOLOGNA

SECONDA FACOLTÀ DI INGEGNERIA
CON SEDE A CESENA

DIPARTIMENTO DI INGEGNERIA DELLE COSTRUZIONI MECCANICHE, NUCLEARI,
AERONAUTICHE E DI METALLURGIA
Sede di Forlì

DOTTORATO DI RICERCA IN
DISEGNO E METODI DELL'INGEGNERIA INDUSTRIALE

PROGETTO E REALIZZAZIONE DEL SITEMA DI NAVIGAZIONE GUIDA E CONTROLLO PER UN ELICOTTERO CON CAPACITÀ DI VOLO AUTONOMO

S.S.D. ING-IND/03
MECCANICA DEL VOLO

CANDIDATO
Ing. Roberto Pretolani

RELATORE
Prof. Ing. GianMarco Saggiani

COORDINATORE
Chiar.mo Prof. Ing. Franco Persiani

XIX° CICLO
Esame Finale anno 2007

ALMA MATER STUDIORUM-UNIVERSITÀ DI BOLOGNA

SECONDA FACOLTÀ DI INGEGNERIA
CON SEDE A CESENA

DIPARTIMENTO DI INGEGNERIA DELLE COSTRUZIONI MECCANICHE, NUCLEARI, AERONAUTICHE E DI METALLURGIA
Sede di Forlì

DOTTORATO DI RICERCA IN
DISEGNO E METODI DELL'INGEGNERIA INDUSTRIALE

DESIGN AND DEVELOPMENT OF THE NAVIGATION, GUIDANCE AND CONTROL SYSTEM FOR AN AUTONOMOUS HELICOPTER

S.S.D. ING-IND/03
MECCANICA DEL VOLO

CANDIDATO
Ing. Roberto Pretolani

RELATORE
Prof. Ing. GianMarco Saggiani

COORDINATORE
Chiar.mo Prof. Ing. Franco Persiani

XIX° CICLO
Esame Finale anno 2007

Table of Contents

List of Tables

List of Figures

Nomenclature

a	lift curve slope
a_0	blade coning angle
a_1	first order harmonic longitudinal flapping coefficient
A	state-space system matrix
$A_{\delta_{lon}}$	gain from longitudinal cyclic input to main rotor flap angle
b_1	first order harmonic lateral flapping coefficient
B	state-space input matrix
$B_{\delta_{lat}}$	gain from lateral cyclic input to main rotor flap angle
C	state-space output matrix
C_{D0}	main rotor profile drag coefficient
$C_{L\alpha}^{ht}$	horizontal tail lift coefficient curve slope
$C_{l\alpha}^{vf}$	vertical fin lift coefficient curve slope
C_Q	torque coefficient
C_T	thrust coefficient
g	gravity acceleration
g_f	tangent of helicopter configuration geometry angle I
g_i	tangent of helicopter configuration geometry angle II
h_{mr}	main rotor hub height above C.G.
h_{tr}	tail rotor hub height above C.G.
I	moment of inertia
[I]	identity matrix
J	performance index
I_{rot}	total inertia of rotating parts
I_β	blade moment of inertia about flapping hinge
k_S	fly-bar mechanical linkage coefficient
K	Riccati matrix
K_i	integral gain
K_p	proportional gain
K_r	guide proportional gain
K_β	main rotor hub stiffness coefficient
K_λ	wake intensity factor calculated at the tail rotor position
K_μ	flapping derivatives scaling coefficient
l_{ht}	horizontal tail location behind C.G.
l_{tr}	tail rotor hub location behind C.G.
L	roll moment
L	feed-back gain matrix
m	helicopter mass
M	pitch moment

n	gear ratio
N	yaw moment
Q	torque
Q_x	state weighting matrix
p	roll rate in body axis frame
P_0	ISA sea level pressure
$P_{z.ISA}$	ISA pressure at an altitude equal to z
P_e	engine power
q	pitch rate in body axis frame
r	yaw rate in body axis frame
r^{cmd}	commanded yaw rate in body axis frame
R	main rotor radius
R	input weighting matrix
S_{vf}	vertical fin area
S_x^f	frontal fuselage drag area
S_y^f	lateral fuselage drag area
S_z^f	vertical fuselage drag area
T	main rotor thrust
T_0	ISA sea level air temperature
$T_{z,ISA}$	ISA air temperature at an altitude equal to z
T_ψ	NED frame to waypoint frame transformation matrix
u	x body axis velocity
u(t)	state-space input vector
u^{cmd}	commanded x body axis velocity
U_e	trim x body axis forward speed
v	y body axis velocity
v^{cmd}	commanded y body axis velocity
v_{vf}	side velocity relative to air at the location of the vertical fin
V_D	helicopter vertical inertial velocity (positive down)
V_h	helicopter vertical inertial velocity
V_h^{cmd}	commanded helicopter vertical inertial velocity
V_{imr}	main rotor induced velocity
V_{itr}	tail rotor induced velocity
V_x	helicopter ground forward speed
V_y	helicopter ground lateral speed
V_∞^{tr}	axial velocity at the location of the tail rotor hub
w	z body axis velocity
w_{ht}	vertical speed at the horizontal tail location
w_{tr}	vertical speed at the tail rotor location
x(t)	vector of system states
X	force along x body axis

X_{track}	along-track distance
X_{Vx}	longitudinal speed derivative
X_{Vy}	lateral speed derivative
y(t)	state-space output vector
Y	force along y body axis
Y_{track}	cross-track distance
z	altitude
Z	force along z body axis
Z_{col}	vertical speed damping derivative
Z_{Vd}	vertical speed damping derivative
β	blade flapping angle
δ_{col}	collective control input
δ_{lat}	lateral cyclic control input
δ_{lon}	longitudinal cyclic control input
δ_{t}	throttle control input
δ_{tail}	tail rotor collective control input
ε_{vf}^{tr}	fraction of the vertical fin area exposed to tail rotor induced velocity
γ	Lock number
λ_0	inflow ratio
μ	advance ratio
μ_z	normal airflow component
Ω	rotor speed
ϕ	roll Euler angle
η_w	coefficient of non-ideal wake contraction
θ	pitch Euler angle
θ_0	commanded collective angle
ρ	air density
σ	solidity ratio
τ_e	rotor time constant
ψ	blade azimuth angle
Ψ	Euler angle for helicopter heading

1 Introduction

The increasing interest in military UAVs (Unmanned Air Vehicles) is fuelling an equally ambitious build-up in the civil community. It is well known that UAVs may represent a promising and cost-effective alternative to manned aircraft for a large number of civil applications [1]. Compared to traditional air vehicles, UAVs may offer significant advantages in terms of human safety (especially in dull, dirty and dangerous missions), operational cost reduction and work rate efficiency. Nevertheless, while research activities in UAV or Rotary Wing UAV systems are very advanced in the United States, UAV interest in Europe has begun only in the last years. As a result, the European Union has sponsored the UAV development program CAPECON (*Civil UAV APplications & Economic Effectivity of Potential CONfiguration Solutions)*, to attempt to kick-start a civil UAV industry in Europe and try to fill the gap with the United States [2]. In the last years, UNIBO has carried out several research projects concerning the development and manufacturing of fixed wing UAV systems for the civil aviation market. For that reason, when the EU decided to start the CAPECON program, UNIBO didn't hesitate to take part in. Besides its partnership in the CAPECON program, UNIBO has also started a rotary wing UAV research program, since RUAV systems may represent an alternative to fixed wing UAVs (or even a more promising solution) for many UAV civilian applications due to their versatile flight modes, manoeuvrability and vertical take-off and landing capability.
The main goal of UNIBO RUAV research program is to develop a helicopter capable of autonomous flight which could be used inside the Universities as a platform for researches in control and navigation laws; meanwhile it should be proposed as a technological prototype for industries interested in UAV development and manufacturing.
One important aspect, derived from the above mentioned EU program, is the real need of applying proven technologies to the UAV world in order to take advantage of existing and cost effective technology [3,4]. For that reason, UNIBO has decided to evaluate the feasibility of using COTS sensors and electronics for its RUAV avionics package.

During the last 3 years I have been involved both in the CAPECON program and in the UNIBO small RUAV research project (Figure 1).

For the CAPECON project I have mainly worked in the design and development of a real-time full Mission Simulation Environment that could be integrated with standard preliminary design techniques or used as test bench for researches in control laws, man-machine interfaces and system integration. Inside this project my job has been the development of a RUAV dynamics simulator and its related NGCS (Navigation, Guidance and Control System), described in the first part of this thesis, to be integrated in the Mission Simulation Environment.

In the UNIBO small RUAV research project I was responsible for the design of the autopilot to be implemented in the helicopter onboard computer. As a result during February 2007 the first autonomous flight was carried out successfully. A step by step description of the work done is presented in the second part of this thesis.

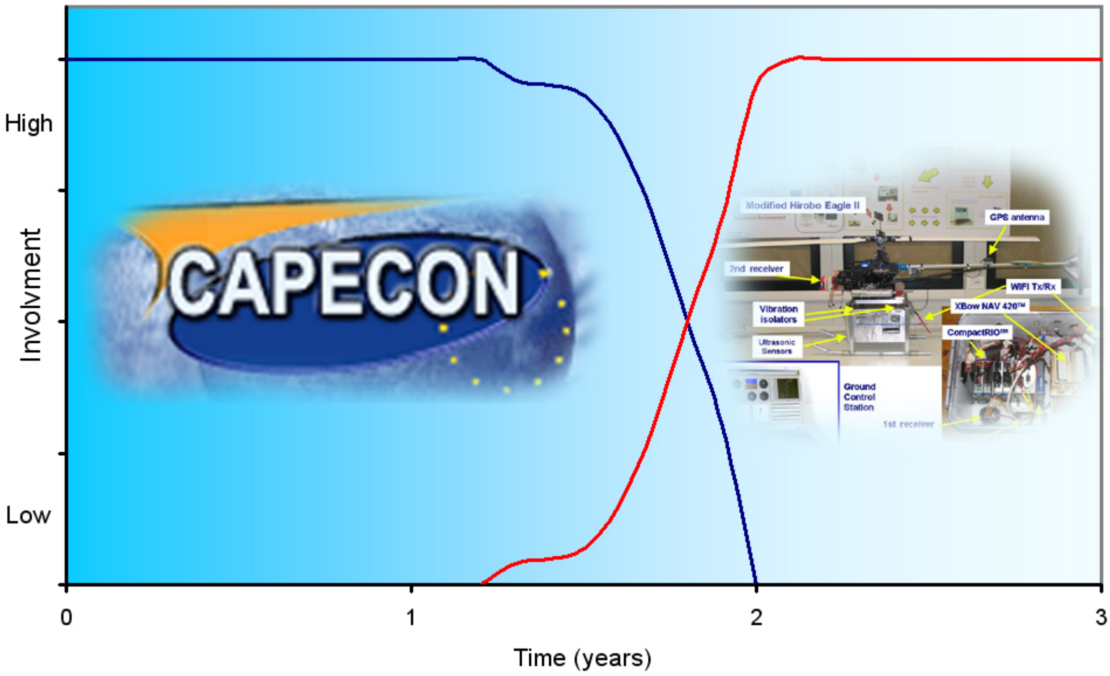


Figure 1: Projects Involvement

Part I: RUAV Mission Simulation Environment

2 Mission Simulation Environment: Introduction

The culmination of the UNIBO CAPECON work was to develop a Mission Simulation Environment for a rotary wing UAV system [5] in order to evaluate the operational capabilities of the configurations designed by the industrial partners (Agusta and Eurocopter). The simulation environment has been developed as an integrated modular system constituted by four independent parts: the ground control station, the visual system, the air vehicle dynamics simulator, the related Navigation Guidance and Control System (NGCS).

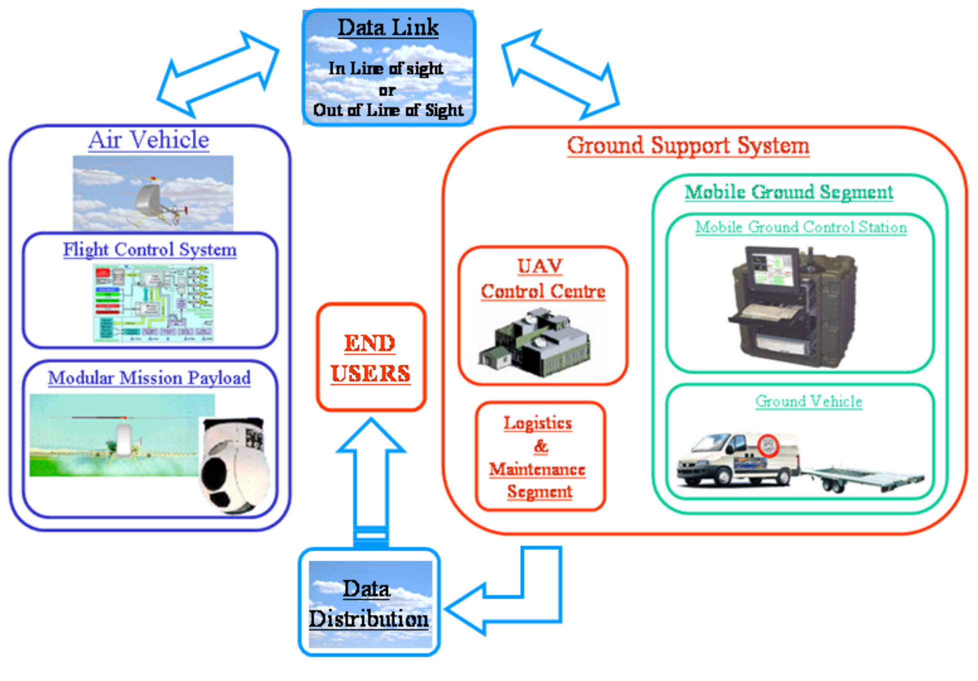


Figure 2: System Definition

Basically an RUAV system consists of the following main sub-systems [6] as shown in Figure 2: The AV (Air Vehicle), the Ground Support System, the DL (Data Link) and the DD (Data Distribution). The AV is made up of the basic helicopter platform, the onboard FCS (Flight Control System) and the MMP (Modular Mission Payload). The Ground Support System includes all those ground infrastructures and equipments to enable AV operations. It is composed mainly by a MGS (Mobile Ground Segment), an UAV Control Centre and a LMS (Logistics & Maintenance Segment). The MGS consists of the MGCS (Mobile Ground Control Station) and a GV (Ground Vehicle). The DL will support video, data, and telemetry communications between the AV and the ground systems. DL communications can be either In Line Of Sight or Out Line of Sight. The DD will transmit annotated significant data, collected at the MGCS or at the UAV Control Centre, to potential users at remote locations. Finally, the LMS includes all elements for supporting the RUAV system operations such as depot level maintenance, overhauls and supplies.

In order to develop a mission simulation environment for an RUAV system, it is necessary to model the main sub-systems of the RUAV. They can be identified as the air vehicle, the ground control station, the data link and the modular mission payload. The simulated RUAV system is therefore constituted by a cluster of computers each one playing a specific role. Figure 3 shows the simulation environment architecture developed at UNIBO laboratories. The simulation environment incorporates three PCs:

- **One "Air Vehicle" computer**: This computer represents "the airborne world". It contains the Simulink™ model of the air vehicle and of the NGCS. UNIBO has developed a Simulink™ simulation model for the classical helicopter configuration, developed by Agusta, in order to test the mission simulation environment. This non linear nine degree of freedom rotorcraft model, developed in Simulink™ emulates sensor output signals and receives back as inputs real-time control signals from the GCS
- **Two "Ground Support System" computers**: for the sake of simplicity the ground support system has been simulated as a unique control station which is able to perform at least the three GCS main functions: Mission Planning, Mission Control, Data Management and Visualization. The ground station is constituted by two computer:
    - the primary master computer is used for real time mission planning and RUAV control. It manages also the datalink between the computers of the mission simulation environment by means of a Labview software. The master computer is connected to two TFT monitors which display the mission planning window and the flight control window
    - the second computer is used for modular mission payload data display. The modular mission payload depends on the simulated mission. It is possible to simulate an EO payload by means of a visual system developed at UNIBO laboratories. The second computer of the ground control station receives data from the primary master and displays a 3D virtual view on a TFT monitor
- **Data Link**: Communication between the AV and the GCS is simulated via LAN (local area network). Bidirectional communication between the AV and the GCS primary master computer is done by means of TCP/IP protocol managed by a Labview software. Communication between the two computer of the GCS is done via UDP protocol and is always managed by the Labview software

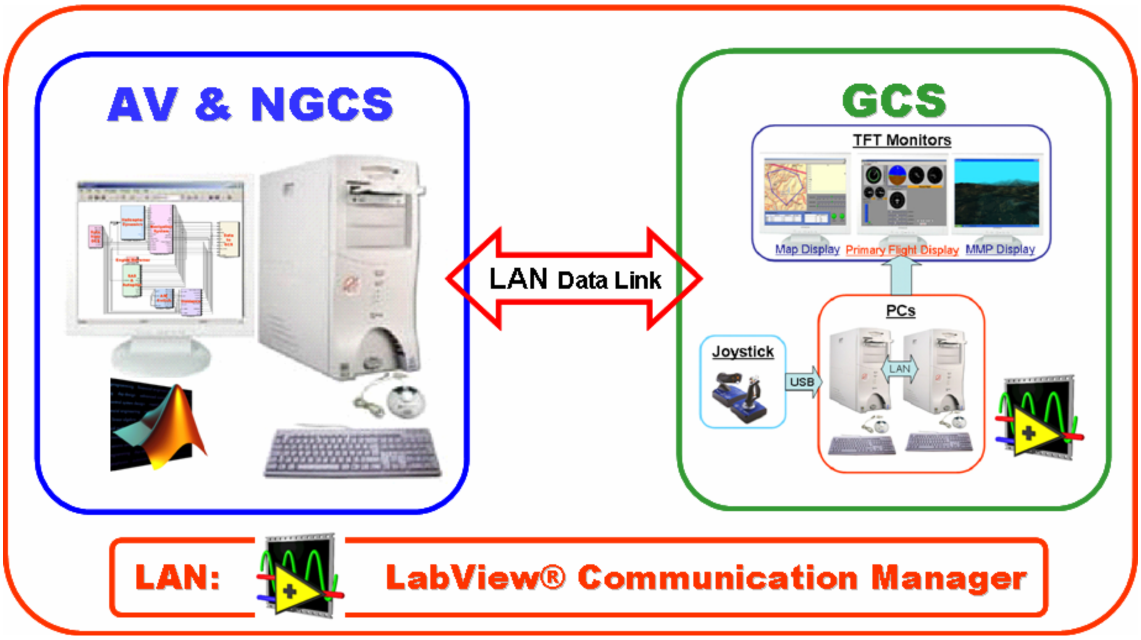


Figure 3: RUAV Mission Simulation Environment Architecture

2.1 AV & NGCS Simulink™ model: Overview

In order to simulate the behavior of an autonomous RUAV, it is necessary to model the air vehicle dynamics and the NGCS.
The complete Simulink model developed by UNIBO is shown in Figure 4.

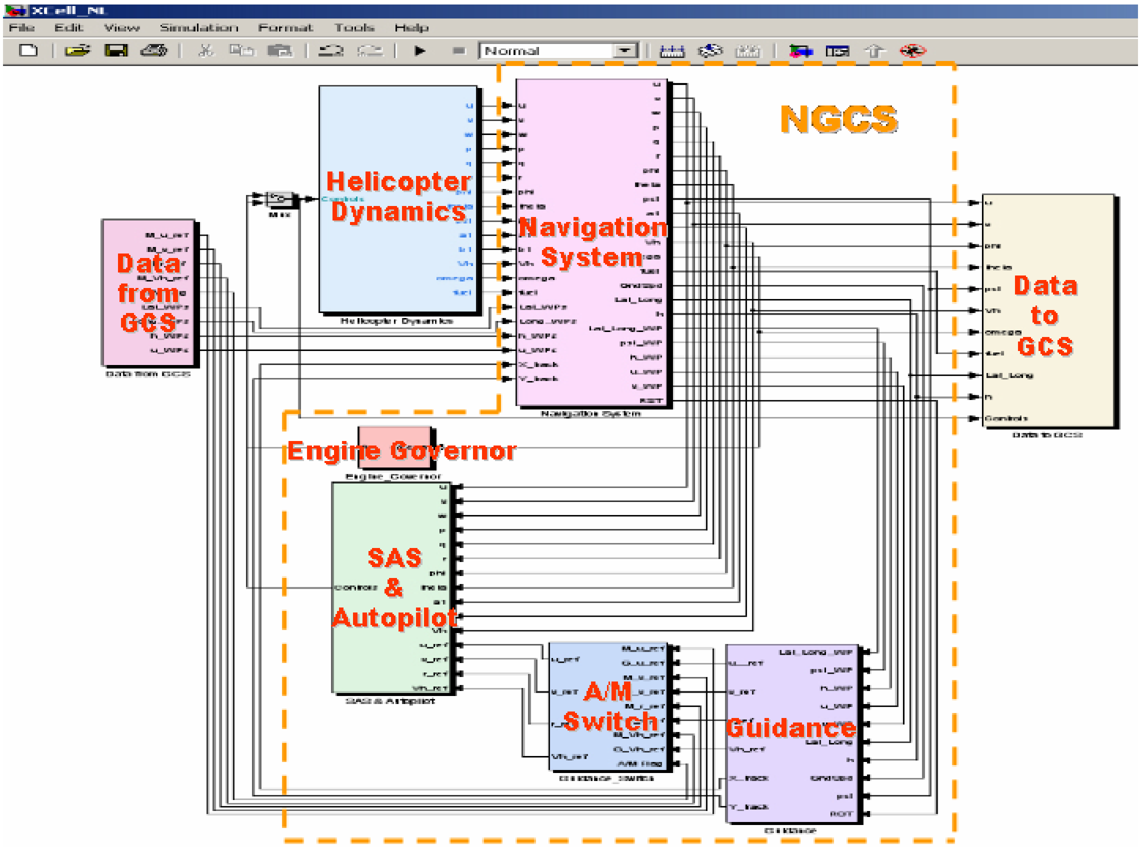


Figure 4: Complete AV & NGCS Simulink model

It is constituted by several different blocks:

- a “data from GCS” block which receives input from the ground control station. Input data are the Joystick commands (in manual flight), the flight plan commands (in autonomous flight) and a flag signal for switching between manual and autonomous flight mode
- an “helicopter dynamics” block which is able to simulate the flight dynamics of a classical main & tail rotor helicopter

The NGCS is able to provide control for the air vehicle stabilization and enables the air vehicle to stably track a set of pre-planned flight segments, starting from any initial condition. It is composed by the following blocks:

- the ”navigation system” block which receives input data from the GCS (the 4D flight plan waypoints) and from the helicopter dynamics block (the state parameters). It is able to select the next waypoint to reach, depending on the helicopter current position. Meanwhile it generates input data for other system blocks (see Figure 4)

- a “Guidance” block which generates reference parameters for the autopilot to track the pre-planned flight segments
- The “Automatic/Manual Switch” block which is able to switch between manual and automatic flight mode, depending on a flag input signal coming from the GCS
- The “SAS & Autopilot” block which works both as stabilization and autopilot system. The autopilot gives controls to the helicopter dynamics block in order to obtain the reference flight parameters generated by the guidance system (in automatic flight) or by the joystick input (in manual flight)
- The “Engine Governor” block which changes the throttle settings in order to maintain constant rotor RPM
- A “Data to GCS” block which passes useful output data to the GCS (see Figure 4)

3 The Air Vehicle

The Air Vehicle Real-Time Simulink™ Dynamic Model developed by UNIBO is valid for Main & Tail Rotor Helicopter Configuration up to advance ratio of 0.15. The goal was to obtain a model with the least amount of complexity, yet accurate across the flight conditions encountered during a typical RUAV survey mission.

3.1 Equation of Motion

The helicopter model has been built by combining the six degrees of freedom rigid body equations of motion (in body axis) with the lateral & longitudinal flapping dynamics and the rotorspeed dynamics [7].

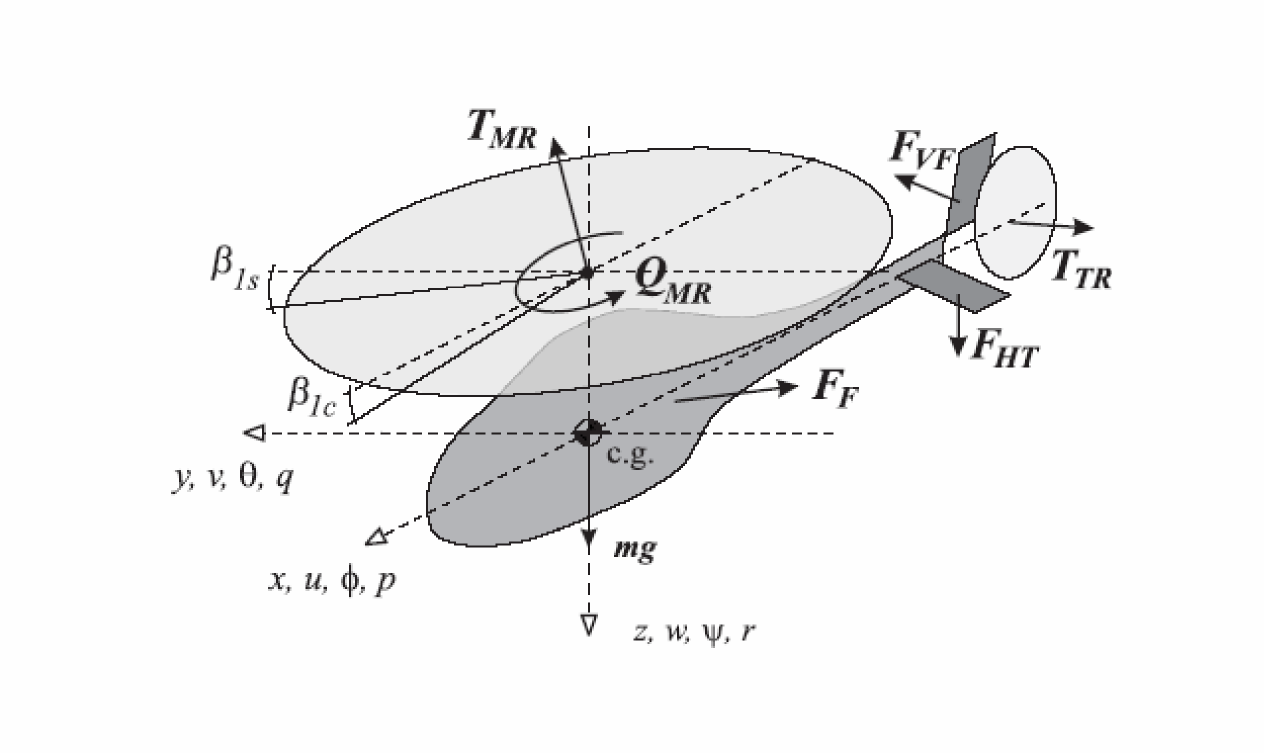


Figure 5: AV reference frame, forces & moments

The rigid body equations of motion for a helicopter are given by the Newton-Euler equations shown below. Here the cross products of inertia are neglected.

$$\dot{u} = vr - wq - g\sin\theta + (X_{mr} + X_{fus})/m$$

$$\dot{v} = wp - ur + g\sin\phi\cos\theta + (Y_{mr} + Y_{fus} + Y_{tr} + Y_{vf})/m$$

$$\dot{w} = uq - vp + g\cos\phi\ \cos\theta + (Z_{mr} + Z_{fus} + Z_{ht})/m$$

$$\dot{p} = qr(I_{yy} - I_{zz})/I_{xx} + (L_{mr} + L_{vf} + L_{tr})/I_{xx}$$

$$\dot{q} = pr(I_{zz} - I_{xx})/I_{yy} + (M_{mr} + M_{ht})/I_{yy}$$

$$\dot{r} = pq(I_{xx} - I_{yy})/I_{zz} + (-Q_e + N_{vf} + N_{tr})/I_{zz}$$

The set of forces and moments acting on the helicopter are organized by components: $()_{mr}$ for the main rotor; $()_{tr}$ for the tail rotor; $()_{fus}$ for the fuselage (including fuselage aerodynamic effects); $()_{vf}$ for the vertical fin and $()_{ht}$ for the horizontal stabilizer. These forces and moments are also shown in Figure 5 along with the main helicopter variables. Q_e is the torque produced by the engine to counteract the aerodynamic torque on the main rotor blades. Q_e is considered ≥ 0 when the helicopter blades rotate clockwise (viewed from above). In the above equations it is assumed that the fuselage center of pressure coincides with the c.g.; therefore, the moments created by the fuselage aerodynamic forces were neglected.

3.2 Main rotor forces and moments

3.2.1 Thrust

For the main rotor thrust we assumed that the inflow is steady and uniform. A momentum theory based iterative scheme given by Padfield [8] was adapted to compute the thrust coefficient and inflow ratio as a function of airspeed, rotor speed and collective setting. We neglect the flapping angles in the computation of the rotor thrust. The blades of the main rotor have no twist. The influence of the cyclics and the roll rate on thrust are of second order for our advance ratio range $\mu < 0.15$, and were neglected as well. We also introduced an empirically determined maximum thrust coefficient, since momentum theory does not take into account the effect of blade stall. The thrust coefficient is given by (omitting the “mr” index):

$$C_T = \frac{T}{\rho(\Omega R)^2 \pi R^2}$$

where T is the main rotor thrust.

Then the following system of equations can be solved iteratively:

$$\lambda_0 = \frac{C_T}{2\eta_w\sqrt{\mu^2 + (\lambda_0 - \mu_z)^2}}$$

$$C_T = \frac{a\sigma}{2}\left(\theta_0\left(\frac{1}{3}+\frac{\mu^2}{2}\right)+\frac{\mu_z - \lambda_0}{2}\right)$$

Here

$\mu = \frac{\sqrt{u^2+v^2}}{\Omega R}$, advance ratio

$\mu_z = \frac{w}{\Omega R}$, normal airflow component

$\sigma = \frac{2c}{\pi R}$, solidity ratio

a, lift curve slope

θ_0, commanded collective angle

η_w, coefficient of non-ideal wake contraction

Based on momentum theory, the rotor wake far downstream contracts by a factor of two [8]. We introduced a coefficient η_w to account for non-ideal wake contraction and the power lost due to the non-uniform velocity and pressure distribution in the wake. We have approximated this coefficient to be η_w=0.9.

Note that at hover the denominator of the equation describing λ_0 is zero when the vertical velocity is equal to the inflow velocity. This condition corresponds to a vortex-ring state, which can not be modeled adequately by the momentum theory. Instead, the denominator is numerically separated from zero. In general, this condition is avoided in flight because it leads to a loss of control. We have to keep in mind that the simulation does not adequately represent the helicopter dynamics when vortex-ring conditions exist on either the main or the tail rotor. Furthermore, strictly speaking the momentum theory applies only to a fully developed steady state flow in ascending flight. Empirical corrections for descending flight, cited by Padfield [8], could be used to make thrust prediction somewhat more accurate.

3.2.2 Torque

The main rotor torque can be approximated as a sum of induced torque due to generated thrust, and torque due to profile drag on the blades [8]:

$$C_Q = \frac{Q}{\rho(\Omega R)^2 \pi R^3} = C_T(\lambda_0 - \mu_z) + \frac{C_{D0}\sigma}{8}\left(1+\frac{7}{3}\mu^2\right)$$

where C_Q is the torque coefficient, C_{D0} is the profile drag coefficient of the main rotor blade. The profile drag is not significantly affected by changes in the collective setting.

Thus, the yawing moment produced by the main rotor is given by:

$$\mathrm{Q_{mr}} = \mathrm{C_Q}\ (\Omega \mathrm{R})^2\ \pi \mathrm{R}^3$$

3.2.3 Main Rotor Moments and Flapping Dynamics

The main rotor flapping angle β can be represented as a Fourier series of the blade azimuth angle ψ, with only the first three coefficients retained [8]:

$$\beta(\psi) = \mathrm{a_0} + \mathrm{a_1} \cos\psi + \mathrm{b_1} \sin\psi$$

Flapping of the teetering stabilizer bar can be represented by a similar equation without the constant term since no coning takes place:

$$\beta_S(\psi) = \mathrm{a_{1S}} \sin\psi + \mathrm{b_{1S}} \cos\psi$$

Stabilizer bar flapping contributes to the change of the main rotor blade pitch angle through a mechanical linkage:

$$\theta(\psi) = \theta_0 + \theta_{lon} \sin\psi + \theta_{\mathrm{lat}} \cos\psi + k_S \beta_{\mathrm{S}}$$

The swashplate deflections change the cyclic pitch angle of both the main rotor and the stabilizer bar. Coupled second-order differential equations can be developed for Fourier coefficients of the main rotor and stabilizer bar flapping. It can be shown [8] that the undamped natural frequency of the flapping motion is close to the rotorspeed Ω_{mr}, and the damping ratio can be approximated by γ/8, where γ is the Lock number of the blades being considered (main rotor or stabilizer bar). The Lock number represents the ratio of aerodynamic to inertial forces and is defined as:

$$\gamma = \frac{\rho c a R^4}{I_\beta}$$

For the main rotor blades the Lock number is relatively high, therefore the flapping motion is well damped. For the stabilizer bar, with its small aerodynamic surfaces, the Lock number is low and the corresponding settling time is much higher than the main rotor one. Earlier work on modeling of small-scale rotorcraft with Bell-Hiller stabilizer bars [9, 10,11] showed that the main rotor and stabilizer bar flapping dynamics can be lumped and represented by tip-path plane (TPP) flapping dynamics with only two states. This result was based on frequency-domain identification and comparison of reduced and full order transfer functions for attitude dynamics. Furthermore, coupling of the lumped flapping dynamics and rigid body pitch and roll motions leads to pronounced second-order characteristics [12,13,10,11]. These modes are lightly damped, and should be explicitly accounted for in designing high-bandwidth attitude or rate control systems [14, 15].

We represented the lateral and longitudinal flapping dynamics by the first-order equations:

$$\dot{b}_1 = -p - \frac{b_1}{\tau_e} - \frac{1}{\tau_e}\frac{\partial b_1}{\partial \mu_v}\frac{v}{\Omega R} + \frac{B_{\delta_{lat}}}{\tau_e}\delta_{lat}$$

$$\dot{a}_1 = -q - \frac{a_1}{\tau_e} + \frac{1}{\tau_e}\left(\frac{\partial a_1}{\partial \mu}\frac{u}{\Omega R} + \frac{\partial a_1}{\partial \mu_z}\frac{w}{\Omega R}\right) + \frac{A_{\delta_{lon}}}{\tau_e}\delta_{lon}$$

where $B_{\delta_{lat}}$ and $A_{\delta_{lon}}$ are effective steady-state lateral and longitudinal gains from the cyclic inputs to the main rotor flap angles; δ_{lat} and δ_{lon} are the lateral and longitudinal cyclic control inputs (pilot stick or control system outputs τ_e is the effective rotor time constant for a rotor with the stabilizer bar:

$$\tau_e = \left({}^{16}\!/_{\gamma_{fb}\Omega} \right)$$

The dominant rotor moments are the control moments produced by the rotor flapping. In the following we describe the moments in the roll direction (resulting from the lateral TPP flapping b_1). Figure 6 shows the rotor moments that are acting on the fuselage. The first contribution results from the restraint in the blade attachment to the rotor head.

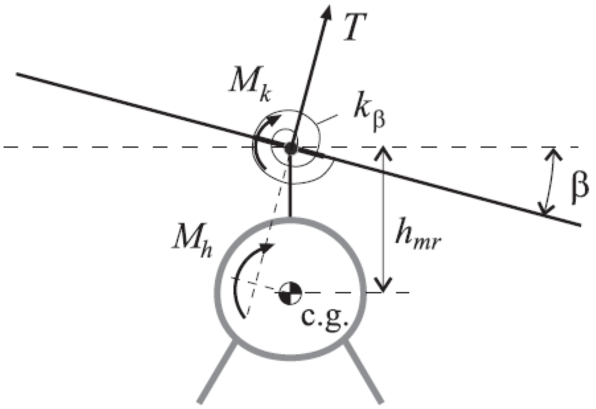


Figure 6: Rotor moments acting on the fuselage

The restraint can be approximated using a linear torsional spring with a constant stiffness coefficient K_β; resulting in a roll moment :

$$M_{K,lat} = K_\beta b_1$$

The second contribution results from the tilting of the thrust vector. Assuming that the thrust vector is perpendicular to the TPP, the thrust vector will tilt proportionally to the rotor flapping angles. The moment arm is the distance h_{mr} between the rotor head and the helicopter center of gravity; resulting in a lateral moment:

$$M_{h,lat} = Th_{mr} b_1$$

The total main rotor rolling moment, entering the rigid body equations of motion, is represented by:

$$L_{mr} = \left(K_\beta + Th_{mr}\right) b_1$$

Similarly, the pitching moment is given by:

$$M_{mr} = \left(K_\beta + Th_{mr}\right) a_1$$

The flapping due to translational velocity is described by the flapping derivatives $\frac{\partial a_1}{\partial \mu}, \frac{\partial b_1}{\partial \mu_v}$. From rotor symmetry we conclude that the longitudinal and lateral dihedral derivatives are equal in magnitude, and in both cases cause the rotor to flap away from the incoming air.

$$\frac{\partial a_1}{\partial \mu} = -\frac{\partial b_1}{\partial \mu_v}$$

A theoretical value for the derivative is given in [16]:

$$\frac{\partial a_1}{\partial \mu} = 2K_\mu \left(\frac{4\delta_{col}}{3} - \lambda_0 \right)$$

Where K_μ is a scaling coefficient that take into account the presence of the fly bar that dramatically reduces the value of $\frac{\partial a_1}{\partial \mu}$. A rough estimate of K_μ is 0.2.

Positive z-axis velocity causes higher lift on advancing blade, which results in a flap-back of the rotor; this effect is captured by the stability derivative $\frac{\partial a_1}{\partial \mu_z}$. An analytical estimate of the derivative is adapted [16] to accommodate backward flight, and scaled by the same coefficient K_μ to reflect the effect of the stabilizer bar:

$$\frac{\partial a_1}{\partial \mu_z} = K_\mu \frac{16\mu^2 4\delta_{col}}{8|\mu| + a\sigma} sign\mu$$

3.2.4 Rotor Forces

For small advance ration flight (μ < 0.15) we can assume that the thrust vector is perpendicular to the TPP. The small flapping angles (below 10 degrees) allow us to use linear approximation for the main rotor force components along the helicopter body axes. As was stated above, the in-plane rotor force was lumped with the fuselage forces, and is not accounted for in the equations below:

$$\begin{aligned} X_{mr} &= -T_{mr} a_1 \\ Y_{mr} &= T_{mr} b_1 \\ Z_{mr} &= -T_{mr} \end{aligned}$$

3.3 Engine and Rotorspeed model

The rotorspeed dynamics is modeled by the following equation:

$$\dot{\Omega} = \dot{r} + \frac{1}{I_{rot}} \left(Q_e - Q_{mr} - \eta_{tr} Q_{tr} \right)$$

where Q_e is the engine torque (positive clockwise), Q_{mr} is the main rotor torque (positive counter-clockwise), Q_{tr} is the tail rotor torque, η_{tr} is the tail rotor gear ratio, I_{rot} is the total rotating inertia referenced to the main rotor speed and Ω is the rotorspeed. The engine torque depends on the throttle setting δ_t and rotorspeed, and is usually represented by engine maps, or look-up tables. The maps for the engine were not available, and a simplified representation of the engine torque is suggested. Assume that engine power is proportional to the throttle setting:

$$P_e = P_e^{\max} \delta_t$$

where 0 < δ_t < 1. Then the torque is:

$$Q_e = \frac{P_e}{\Omega}$$

The engine torque response to throttle changes can be considered instantaneous, since the time lags associated with air intake, fuel flow and combustion are very small compared to vehicle dynamics.
A total kinetic energy of all rotating components is:

$$2 I_{\beta mr} \Omega^2 + I_{es} (n_{es} \Omega)^2 + 2 I_{\beta tr} (n_{tr} \Omega)^2$$

where $I_{\beta mr}$ and $I_{\beta tr}$ are, respectively, the main and the tail rotor blade inertias, I_{es} is the inertia of the engine shaft and all components rotating at the engine speed, n_{tr} is the tail rotor gear ratio, and n_{es} is the engine gear ratio. Therefore the rotating inertia referenced to the main rotor speed can be represented as:

$$I_{rot} = 2I_{\beta mr} + I_{es}n_{es}^{\ 2} + 2I_{\beta tr}n_{tr}^{\ 2}.$$

The most important contribution comes from the main rotor blades. The tail rotor inertia, after scaling with the gear ratio squared, amounts to about 5 percent of the main rotor inertia. The rotating inertia referenced to the engine speed is harder to estimate, but an upper bound can be found by estimating the total mass of rotating components and its effective radius of inertia. We thus obtain an estimate for I_{rot} equal to 2.5 inertias of the main rotor blade.

3.4 Fuselage Forces

For hover flight and forward speeds well below the induced velocity at hover, the rotor downwash is deflected by the forward and side velocity. This deflection creates a force opposing the movement. We can express the x and y drag forces created by the fuselage in this flight regime by:

$$X_f = -\frac{1}{2}S_x^f \rho V_{imr}^{\ 2}\frac{u}{V_{imr}}$$

$$Y_f = -\frac{1}{2}S_y^f \rho V_{imr}^{\ 2}\frac{v}{V_{imr}}$$

where S_x^f and S_y^f are effective drag areas of the fuselage in the x and y directions.

When the forward speed is higher than the rotor induced velocity, the fuselage drag can be modeled as the drag of a flat plate exposed to dynamic pressure. In this case the perturbations to the fuselage forces can be expressed as:

$$X_f = -\frac{1}{2}S_x^f \rho U_e^{\ 2}\frac{u}{U_e}$$

$$Y_f = -\frac{1}{2}S_y^f \rho U_e^{\ 2}\frac{v}{U_e}$$

where U_e is the trim airspeed.

Considering the above equations, fuselage forces can be approximated by [7]:

$$\begin{aligned}
V_\infty &= \sqrt{u^2 + v^2 + \left(w - V_{imr}\right)^2} \\
X_f &= -\frac{1}{2} S_x^f \rho u V_\infty \\
Y_f &= -\frac{1}{2} S_y^f \rho v V_\infty \\
Z_f &= -\frac{1}{2} S_z^f \rho \left(w - V_{imr}\right) V_\infty
\end{aligned}$$

where S_x^f , S_y^f and S_z^f are effective frontal, side and vertical drag areas of the fuselage. We neglect small moments generated by the fuselage, and assume that the fuselage center of pressure coincides with the helicopter center of gravity.

3.5 Vertical Fin Forces and Moments

We approximated the side-force produced by the vertical fin by:

$$Y_{vf} = \frac{1}{2} S_{vf} \rho \left(C_{l\alpha}^{vf} V_\infty^{tr} + \left| v_{vf} \right| \right) v_{vf}$$

where S_{vf} is the vertical fin area, $C_{l\alpha}^{vf}$ is its lift curve slope, $V_\infty^{tr} = \sqrt{u^2 + w_{tr}}$ is the axial velocity at the location of the tail rotor hub. v_{vf} is the side velocity relative to air at the location of the vertical fin, w_{tr} is the vertical velocity (same as for the tail rotor):

$$\begin{aligned}
v_{vf} &= v - \varepsilon_{vf}^{tr} V_{itr} - l_{tr} r \\
w_{tr} &= w + l_{tr} q - K_\lambda V_{imr}
\end{aligned}$$

Here V_{itr} is the induced velocity of the tail rotor, r is the yaw rate, ε_{vf}^{tr} is the fraction of the vertical fin area exposed to full induced velocity from the tail rotor, l_{tr} is the distance between the c.g. and tail rotor hub, which is about the same distance to the center of pressure of the vertical fin, V_{imr} is main rotor induced velocity, K_λ is the wake intensity factor, calculated in the tail rotor section.

To accommodate for stall of the vertical fin [17], the absolute value of the vertical fin side force is limited by:

$$\left| Y_{vf} \right| \le \frac{1}{2} S_{vf} \rho \left(\left(V_\infty^{tr} \right)^2 + v_{vf}^{\;2} \right)$$

The vertical fin sideforce creates a yawing moment and a small rolling moment due to the offsets from the c.g.:

$$N_{vf} = -Y_{vf} l_{tr}$$
$$L_{vf} = Y_{vf} h_{tr}$$

3.6 Horizontal Stabilizer Forces and Moments

The horizontal tail produces lift and a stabilizing pitching moment around the center of gravity. An effective vertical speed at the horizontal tail location is determined, assuming that the stabilizer may be fully or partially submerged in the downwash of the main rotor:

$$w_{ht} = w + l_{ht} q - K_{\lambda} V_{imr}$$

The same wake intensity factor is used for the horizontal fin as for the vertical fin and the tail rotor. Next, the z-force generated by the horizontal stabilizer is determined according to:

$$Z_{ht} = \frac{1}{2} S_{ht} \rho \left(C_{L\alpha}^{ht} |u| w_{ht} + |w_{ht}| w_{ht} \right)$$

where S_{ht} is the horizontal stabilizer area, $C_{L\alpha}^{ht}$ = 3.0 is its lift curve slope. To accommodate for the stall of the horizontal stabilizer [17], the absolute value of the horizontal stabilizer lift is limited by:

$$|Z_{ht}| \le \frac{1}{2} S_{ht} \rho \left(u^2 + w_{ht}{}^2 \right)$$

Finally, the pitching moment generated by the horizontal stabilizer is

$$M_{ht} = Z_{ht} l_{ht}$$

3.7 Tail Rotor Forces and Moments

The tail rotor is subjected to a wide range of flow conditions. We need to determine the normal μ_z^{tr} and the in-plane μ_{tr} tail rotor inflow components. The main rotor wake affects the tail rotor thrust in a complex way; to model this influence accurately an extensive modeling of the wake is required. We have decided to approximate just the increase in an apparent in-plane velocity seen by the tail rotor. For this, we have determined the main rotor wake intensity factor K_λ. The geometry calculations are equivalent to those given in [18], but computationally more efficient since an explicit evaluation of the trigonometric functions is avoided. We have calculated the following variables (tangents of the angles determining the geometry):

$$g_i = \frac{l_{tr} - R_{mr} - R_{tr}}{h_{tr}}$$

$$g_f = \frac{l_{tr} - R_{mr} + R_{tr}}{h_{tr}}$$

First, the tail rotor is out of the downwash if $V_{imr} \leq w_a$, in which case there is an effective upwash. Next, at low enough forward speed with respect to air the tail rotor is out of the wake as well. This can be represented by the condition:

$$\frac{u}{V_{imr} - w} \leq g_i$$

In both of these cases $K_\lambda = 0$. The tail rotor is fully in the wake if:

$$\frac{u}{V_{imr} - w} \geq g_f$$

In the far wake the downwash is twice the value at the rotor. We assume that $K_\lambda = 1.5$ when the tail rotor is fully immersed. In the remaining case, when the tail rotor is partially immersed, we assume a linear growth of the wake intensity factor with the forward speed:

$$K_\lambda = 1.5 \frac{\dfrac{u}{V_{imr} - w} - g_i}{g_f - g_i}$$

The derived expression is used to calculate vertical component of airspeed at the tail rotor location. Next determine the advance ratio for the tail rotor:

$$\mu_{tr} = \frac{\sqrt{u^2 + w_{tr}{}^2}}{\Omega_{tr} R_{tr}}$$

The velocity component normal to the tail rotor is given by:

$$v_{tr} = v - l_{tr} r + h_{tr} p$$

and in non-dimensional form:

$$\mu_{ztr} = \frac{v_{tr}}{\Omega_{tr} R_{tr}}$$

The tail rotor thrust Y_{tr} is computed using the same scheme as for the main rotor [8] with the inflow ratio approximated by:

$$\lambda_0^{tr} = \mu_{ztr} - 2\left[\frac{2C_T^{tr}}{a_{tr}\sigma_{tr}} - \delta_r\left(\frac{1}{3} + \frac{\mu_{tr}{}^2}{2}\right)\right]$$

where C_T^{tr} is the computed tail rotor thrust coefficient, a_{tr} is the tail rotor blade lift curve slope, $\sigma_{tr} = \frac{2c_{tr}}{\pi R_{tr}}$ is the tail rotor solidity ratio.

The yawing and rolling moments due to the offset of the tail rotor thrust from the center of gravity are computed as follow:

$$N_{tr} = -Y_{tr} l_{tr}$$
$$L_{tr} = Y_{tr} h_{tr}$$

Finally, the tail rotor torque Q$_{tr}$ is computed as for the main rotor using the tail rotor parameters in place of the main rotor parameters.

3.8 Atmospheric Conditions

To take into account the changes of forces and moments due to altitude and air temperature; the following ISA equations have been considered:

$$T_{z,ISA} = T_0 - 6.5 * 10^{-3} \, {}^{K}\!/_{m} * z \, , \; T_0 = 288{,}15 \text{ K}$$

$$P_{z.ISA} = P_0 \left(\frac{T_{z,ISA}}{T_0} \right)^{5.256} \, , \; P_0 = 101325 \text{ Pa}$$

$$\rho_{z,ISA} = \frac{P_{z,ISA}}{RT_{z,ISA}} \, , \; \mathrm{R} = 287{,}26 \, {}^{\mathrm{J}}\!/_{\mathrm{kgK}}$$

3.9 Non-Linear Model Implementation and Verification

The equations presented in the last section have been implemented in Simulink™ for testing and simulation purposes. The structure of this Simulink™ model follows the schematic shown in Figure 7. This structure divides the entire non-linear model into 8 parts: Main Rotor Forces and Moment, Fuselage Forces, Vertical Fin Forces and Moment, Horizontal Tail Forces and Moment, Engine and Rotorspeed Dynamics, Flapping Dynamics, Rigid Body Dynamics and Standard ISA Atmosphere calculation.

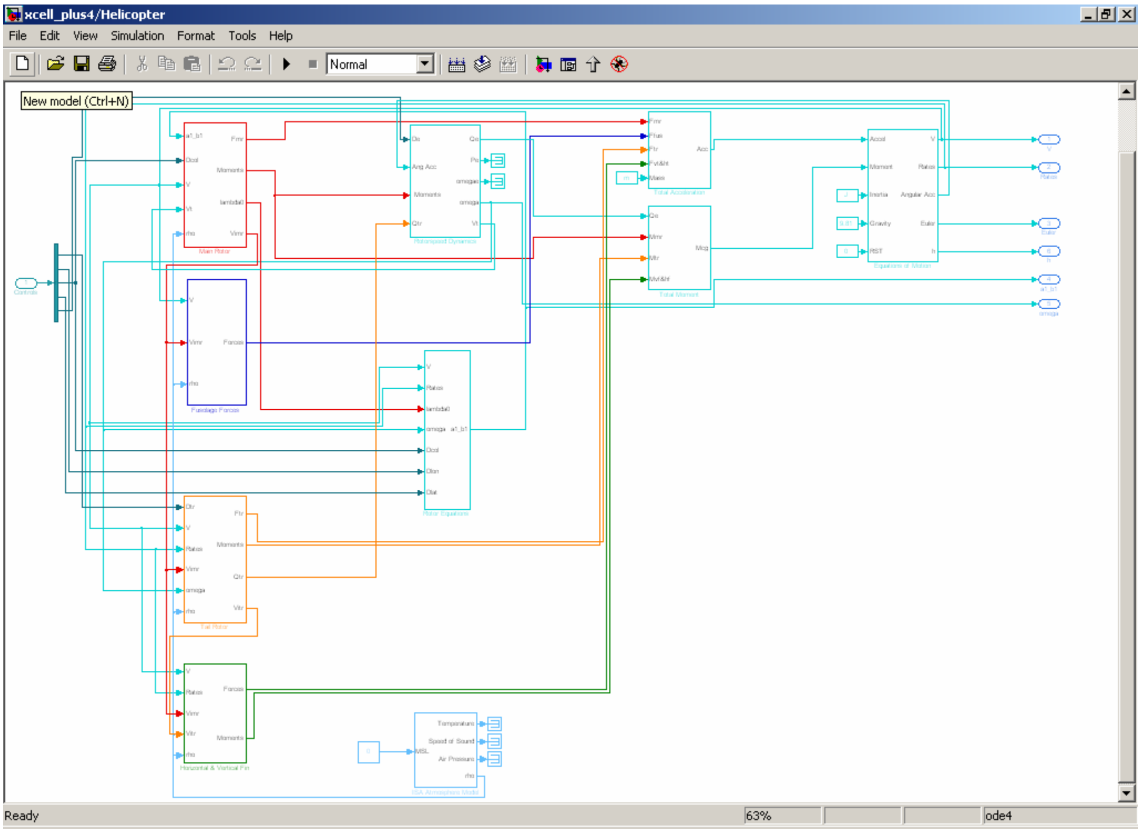


Figure 7: AV Simulink™ Schematic

The whole air vehicle model can be considered as a “black box” with five inputs:

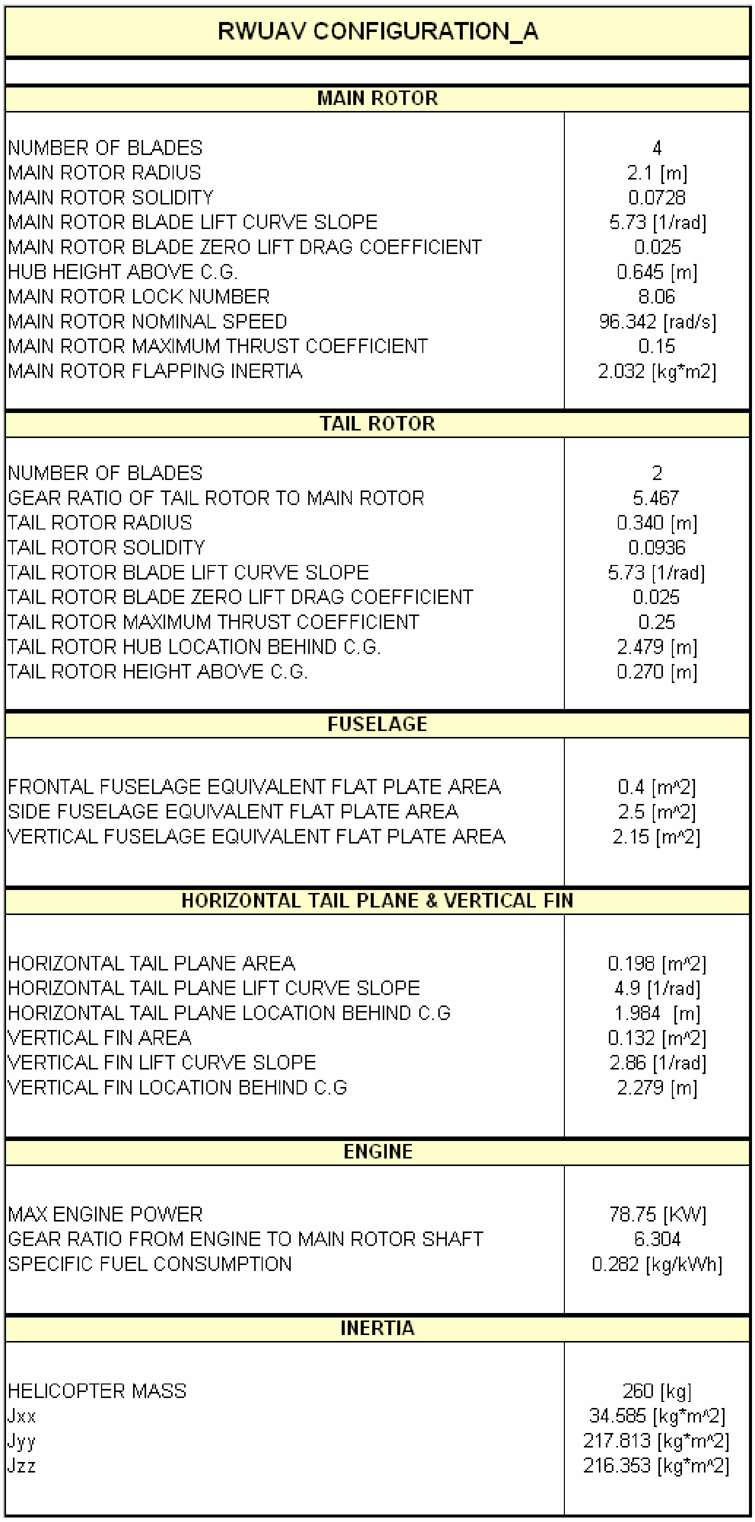
- Tail Rotor Collective;
- Longitudinal Cyclic;
- Main Rotor Collective;
- Lateral Cyclic;
- Throttle;

and 16 main outputs:

- 3 body velocity (u,v,w);
- 3 body angular rates (p,q,r);
- 3 Euler angles (φ,θ,ψ);
- 2 flapping angles (a_1,b_1);
- vertical inertial velocity (Vh);

- main rotor, tail rotor & engine RPM;
- fuel consumption.

The helicopter geometrical, inertia and aerodynamic data, necessary to solve the dynamic model, are stored inside a Matlab® configuration file and are automatically loaded when the simulation starts. The data describing the helicopter configuration developed by Agusta [19] are reported in Table 1



RWUAV CONFIGURATION_A	
MAIN ROTOR	
NUMBER OF BLADES	4
MAIN ROTOR RADIUS	2.1 [m]
MAIN ROTOR SOLIDITY	0.0728
MAIN ROTOR BLADE LIFT CURVE SLOPE	5.73 [1/rad]
MAIN ROTOR BLADE ZERO LIFT DRAG COEFFICIENT	0.025
HUB HEIGHT ABOVE C.G.	0.645 [m]
MAIN ROTOR LOCK NUMBER	8.06
MAIN ROTOR NOMINAL SPEED	96.342 [rad/s]
MAIN ROTOR MAXIMUM THRUST COEFFICIENT	0.15
MAIN ROTOR FLAPPING INERTIA	2.032 [kg*m2]
TAIL ROTOR	
NUMBER OF BLADES	2
GEAR RATIO OF TAIL ROTOR TO MAIN ROTOR	5.467
TAIL ROTOR RADIUS	0.340 [m]
TAIL ROTOR SOLIDITY	0.0936
TAIL ROTOR BLADE LIFT CURVE SLOPE	5.73 [1/rad]
TAIL ROTOR BLADE ZERO LIFT DRAG COEFFICIENT	0.025
TAIL ROTOR MAXIMUM THRUST COEFFICIENT	0.25
TAIL ROTOR HUB LOCATION BEHIND C.G.	2.479 [m]
TAIL ROTOR HEIGHT ABOVE C.G.	0.270 [m]
FUSELAGE	
FRONTAL FUSELAGE EQUIVALENT FLAT PLATE AREA	0.4 [m^2]
SIDE FUSELAGE EQUIVALENT FLAT PLATE AREA	2.5 [m^2]
VERTICAL FUSELAGE EQUIVALENT FLAT PLATE AREA	2.15 [m^2]
HORIZONTAL TAIL PLANE & VERTICAL FIN	
HORIZONTAL TAIL PLANE AREA	0.198 [m^2]
HORIZONTAL TAIL PLANE LIFT CURVE SLOPE	4.9 [1/rad]
HORIZONTAL TAIL PLANE LOCATION BEHIND C.G	1.984 [m]
VERTICAL FIN AREA	0.132 [m^2]
VERTICAL FIN LIFT CURVE SLOPE	2.86 [1/rad]
VERTICAL FIN LOCATION BEHIND C.G	2.279 [m]
ENGINE	
MAX ENGINE POWER	78.75 [KW]
GEAR RATIO FROM ENGINE TO MAIN ROTOR SHAFT	6.304
SPECIFIC FUEL CONSUMPTION	0.282 [kg/kWh]
INERTIA	
HELICOPTER MASS	260 [kg]
Jxx	34.585 [kg*m^2]
Jyy	217.813 [kg*m^2]
Jzz	216.353 [kg*m^2]

Table 1: Helicopter configuration data

Figure 8: CAPECON Agusta configuration

To verify the non-linear model, due to the lack of flight test data for this helicopter size, an analysis of the expected movement of an helicopter in hover and in forward flight ($\mu \leq 0.15$) has been carried out [20].

In the next sections it is tested how the non-linear model reacts to the inputs, and it is shown how the states of the non-linear model reacts correctly to the different inputs.

3.9.1 Lateral cyclic positive step

The models will now be tested with positive step on the lateral input δ_{lat}. The expected outcome of this is that the translatory velocity along the body y-axis and the rotational velocity around the body x-axis both become positive. Figure 9 shows that p and v becomes positive as expected when giving positive lateral input.

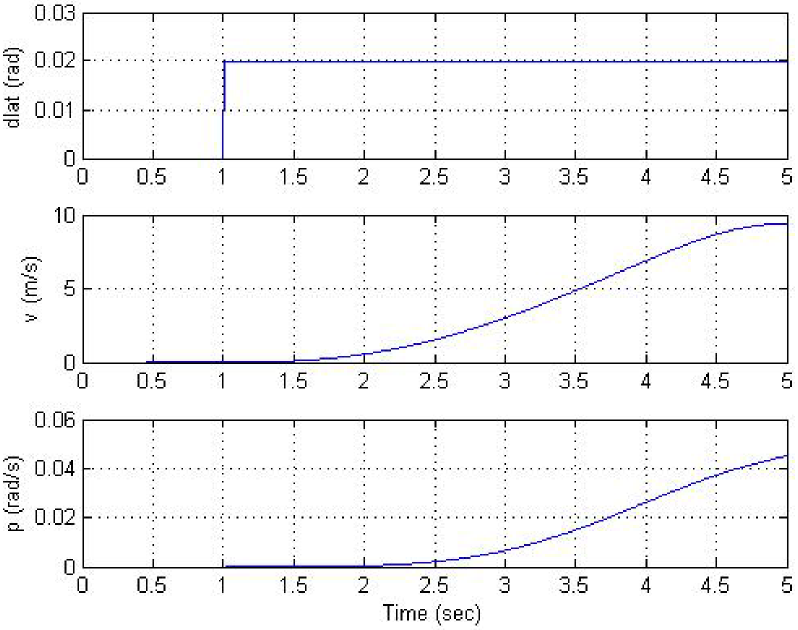


Figure 9: Lateral cyclic positive step

3.9.2 Longitudinal cyclic positive step

The models will now be tested with positive step on the longitudinal input δ_{lon}. The expected outcome of this is that the translatory velocity along the body x-axis becomes negative and the rotational velocity around the body y axis becomes positive. Figure 10 shows that q becomes positive and u becomes negative as expected when giving positive longitudinal input.

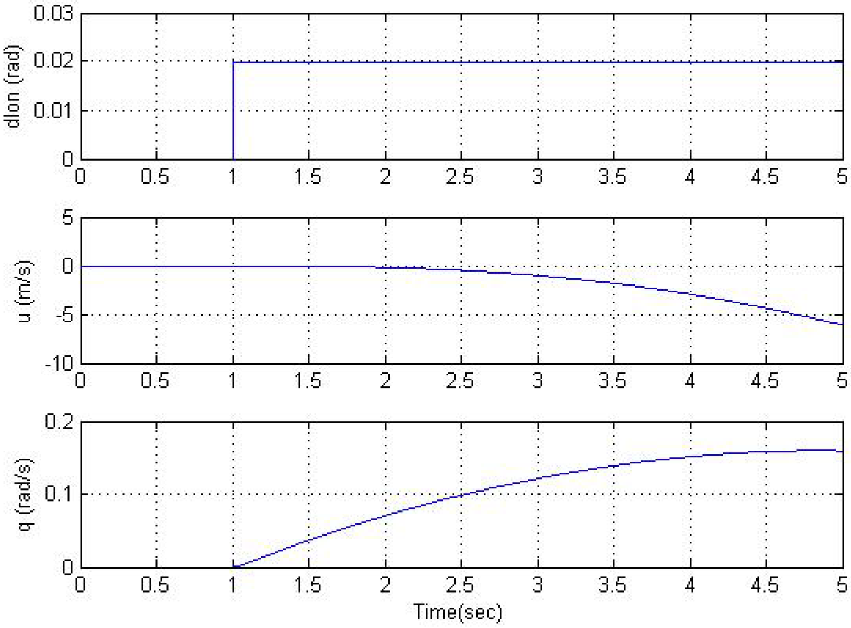


Figure 10: Longitudinal cyclic positive step

3.9.3 Collective positive step

The models will now be tested with positive step on the collective input δ_{col}. The expected outcome of this is that the translatory velocity along the body z axis negative and the rotational velocity around the body z-axis becomes positive. Figure 11 shows that r becomes positive and w becomes negative as expected when giving positive collective input.

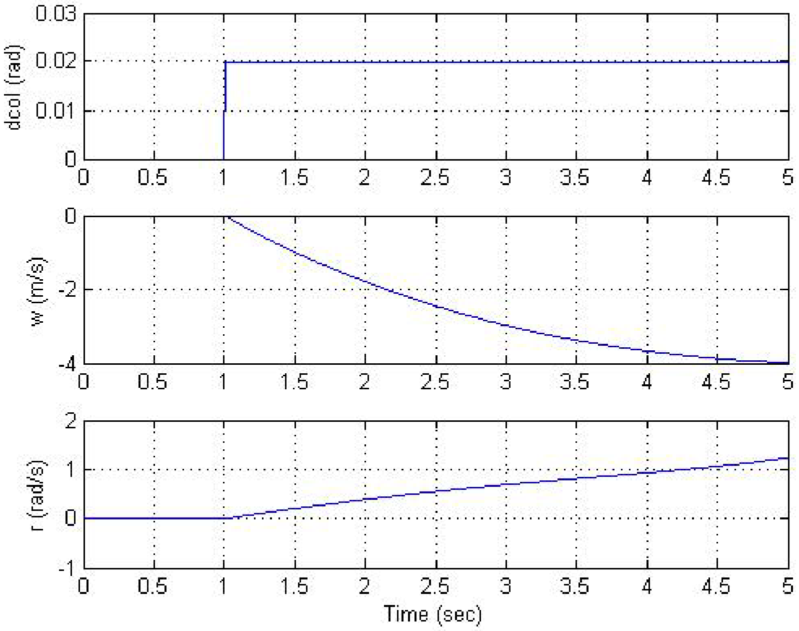


Figure 11: Collective positive step

3.9.4 Pedals positive step

The models will now be tested with positive step on the pedals input δ_{tail}. The expected outcome of this is that the rotational velocity around the body z-axis becomes negative. Figure 12 shows that r becomes negative as expected when giving positive pedals input.

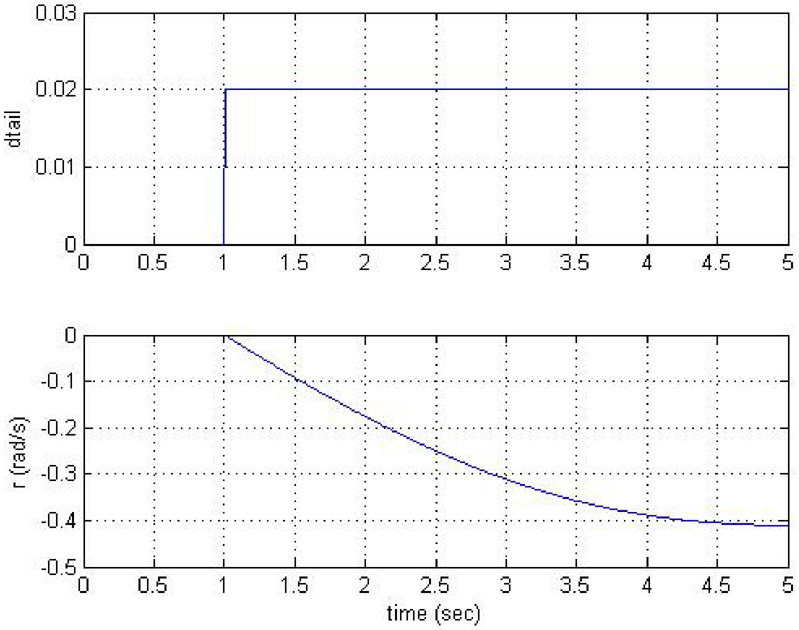


Figure 12: Pedals positive step

3.9.5 Cross-Coupling

To test cross-coupled movements, an analysis based on causes and effect behaviour of the states in the non-linear model has been carried out. As an example, we report the behaviour of the helicopter, applying a negative collective step, starting from trimmed hover conditions.

Reducing the main rotor collective, the main rotor thrust and torque decrease so the helicopter would start to descend and to yaw. So we expect positive w and negative r. With the same amount of lateral cyclic, if the thrust decreases, the helicopter would start to move laterally and bank. In this case we expect positive v and positive p.

In the Figure 13 simulation results are presented for a negative step of 2 degrees collective.

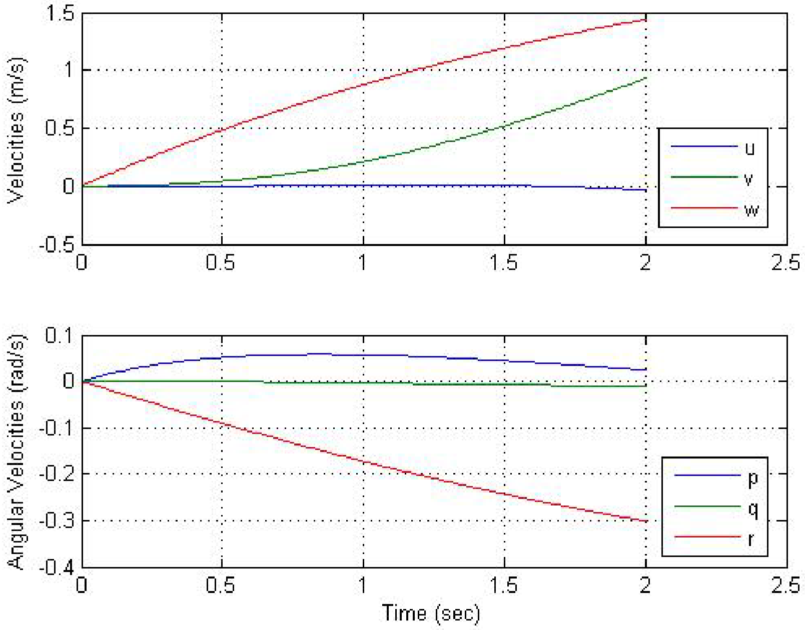


Figure 13: Cross-Coupling (collective negative step)

The movements reported in Figure 13, and all the other tests done, matches the desired helicopter behaviours and thereby the qualitative movement of the helicopter is considered to be verified.

3.10 Helicopter non-Linear model Trim and Linearization

In this section, the helicopter non-linear model has been trimmed and linearized in order to apply linear model control techniques.

3.10.1 Helicopter Trim

A trim point, also known as an equilibrium point, is a point in the parameter space of a dynamic system at which the system is in a steady state. For example, a trim point of a helicopter is a setting of its controls that causes the helicopter to stay in stabilized hover, if no perturbation occurs. Mathematically, a trim point is a point where the system's state derivatives equal zero.

Helicopter trim conditions have been calculated for different x body axis velocities starting from hover up to 30 m/s by step of 5 m/s.

The calculation has been carried out through the TRIM Matlab® command [21]. TRIM starts from an initial point and searches, using a sequential quadratic programming algorithm, until it finds the nearest trim point.

```
[x,u,y] = trim('sys', x0, u0, y0, ix, iu, iy)
```

finds the trim point (states x, input u and output y) of the system 'sys' closest to state x0, input u0 and output y0. The integer vectors ix, iu, and iy select the values in x0, u0, and y0 that must be satisfied.

Figures 14,15,16,17 and 18 show the control trim positions with increasing x body axis velocity. For the whole range of speeds the control positions are as expected.

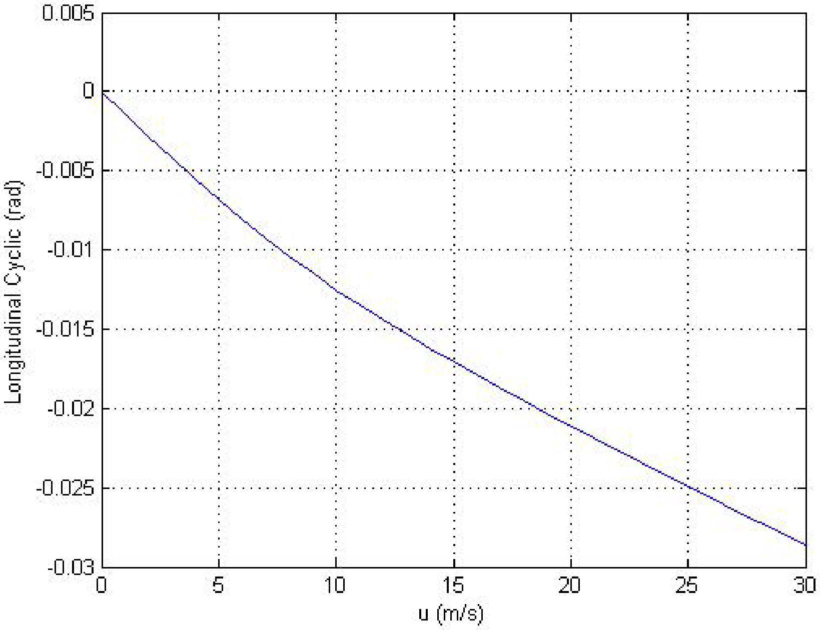


Figure 14: Longitudinal cyclic trim position

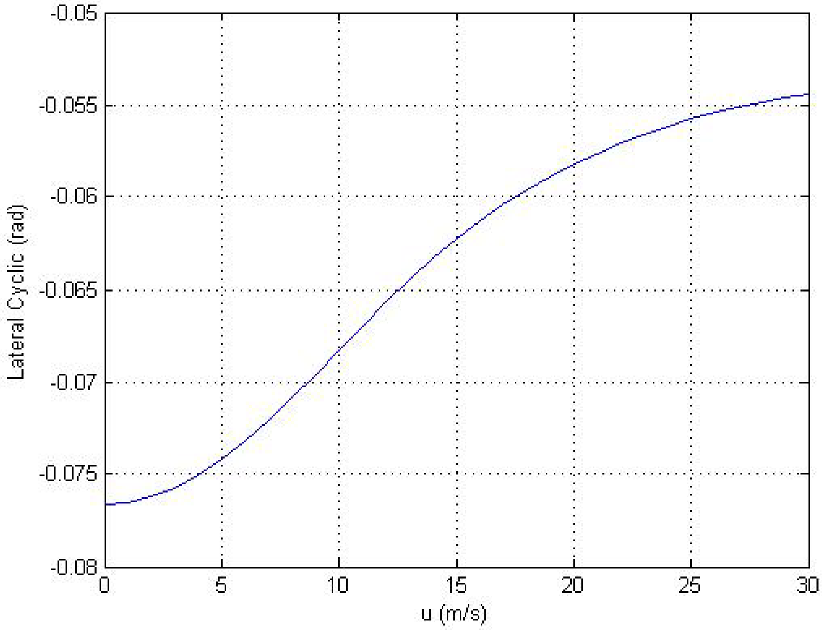


Figure 15: Lateral cyclic trim position

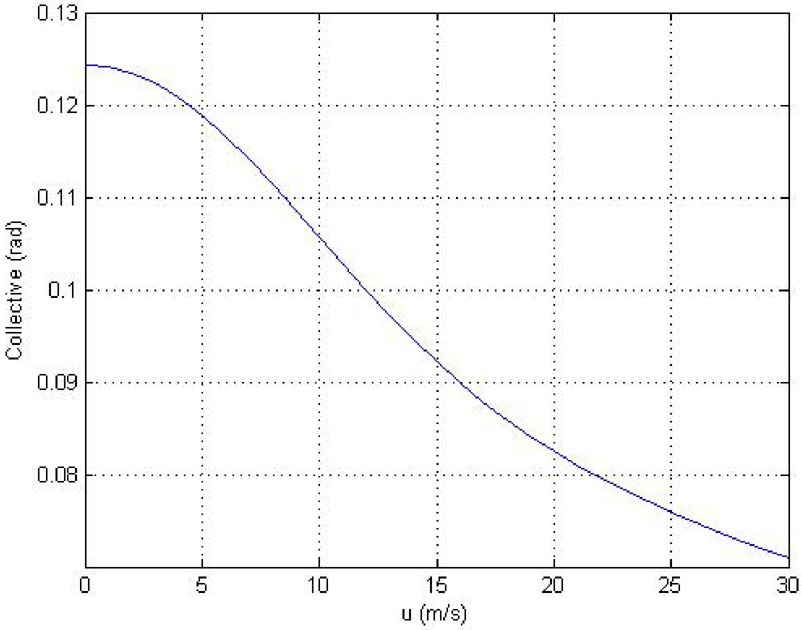


Figure 16: Collective trim position

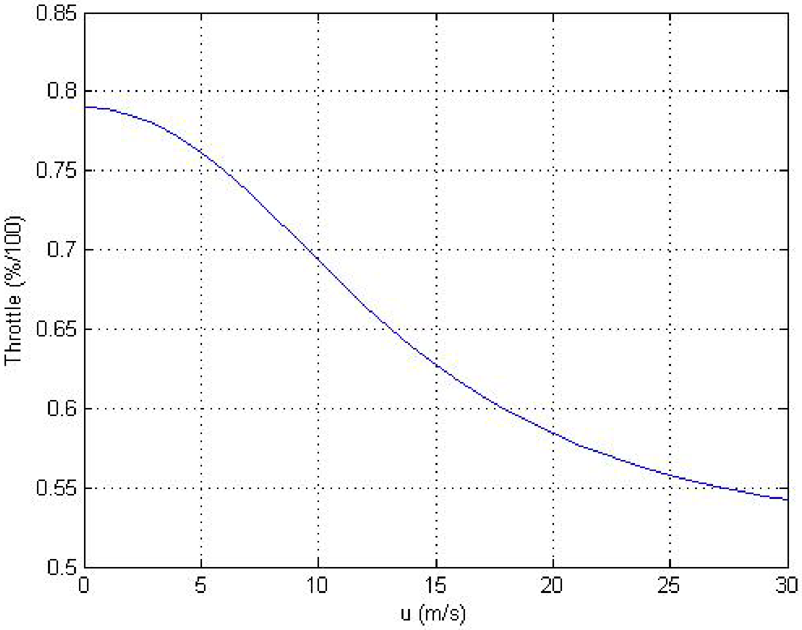


Figure 17: Throttle trim position

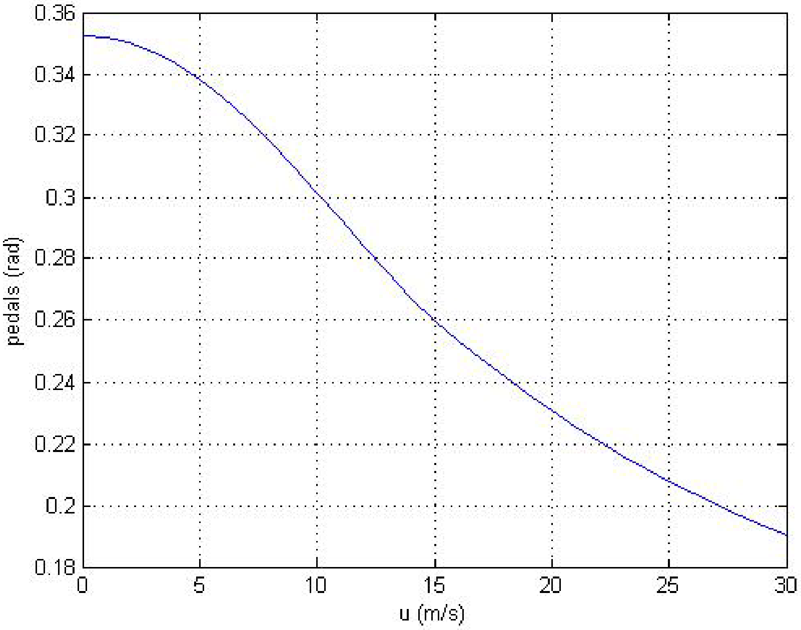


Figure 18: Pedals trim position

Figures 19 and 20 show the vehicle trimmed pitch and roll attitude with increasing x body axis velocity. The vehicle pitch attitude shows the expected trend of an increasing nose-down attitude with increasing speed. The roll attitude shows a roll to the left which is decreasing with increasing speed. The trim conditions were calculated using zero-side slip angle.

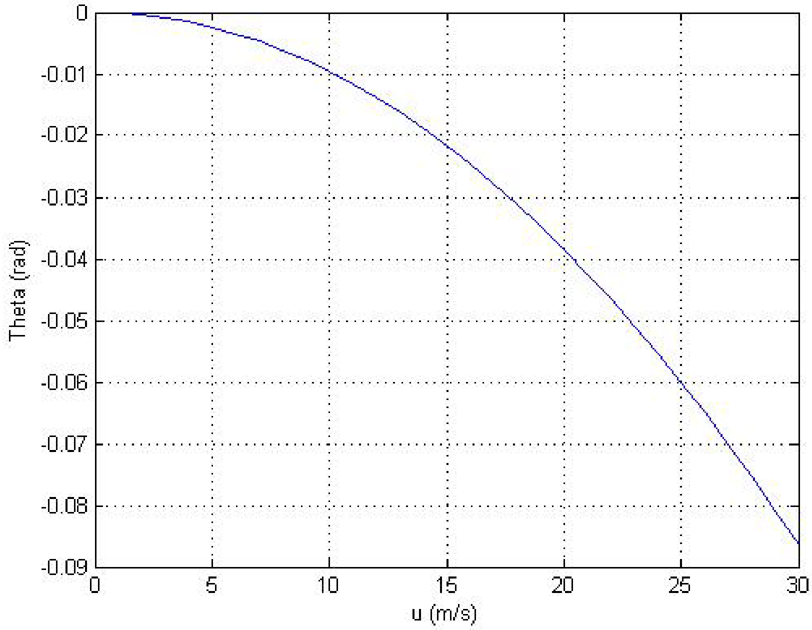


Figure 19: Theta trim

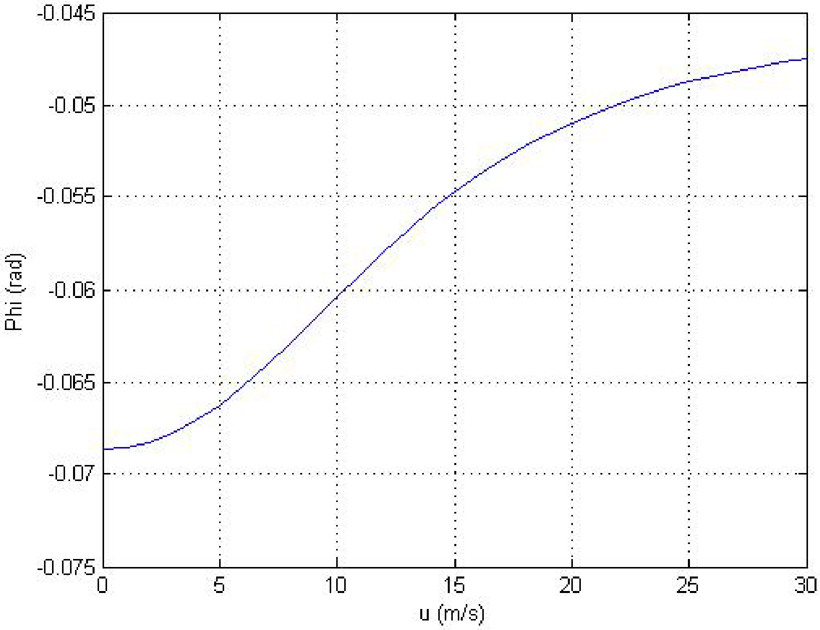


Figure 20: Phi trim

3.10.2 Helicopter Model Linearization

For each calculated trim condition, the non-linear helicopter math model has been linearized to obtain a linear state-space model of the system.
The linear state-space model has the form:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

where A, B, C, D are the system matrices, x(t) the state vector

$$x(t) = [u \quad v \quad w \quad p \quad q \quad r \quad \varphi \quad \theta \quad \psi \quad a_1 \quad b_1\,]^T$$

u(t) the input vector

$$u(t) = [\delta_r \; \delta_{lon} \; \delta_{col} \; \delta_{lat} \;]^T$$

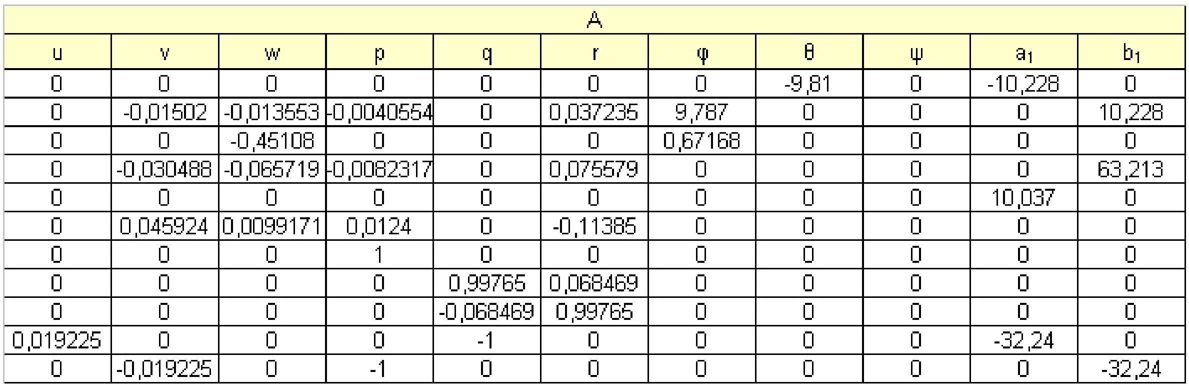
and y(t) the output vector

$$y(t) = [u \quad v \quad w \quad p \quad q \quad r \quad \varphi \quad \theta \quad \psi \quad a_1 \quad b_1\,]^T$$

Since $y(t) \equiv x(t)$ then C=I and D=[0].
The linearization process has been carried out through the usage of the LINMOD Matlab® command [21].

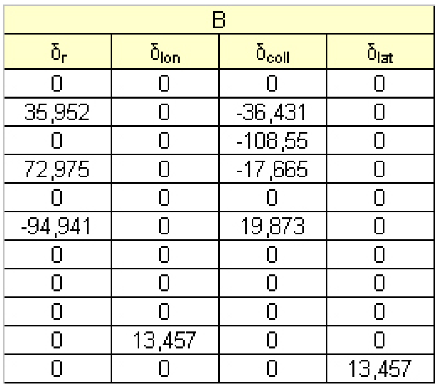
[A,B,C,D] = linmod('sys', x, u)

obtains the linearized model of sys around an operating point with the specified state variables x and the input u. LINMOD compute the linear state space model by linearizing each block in a model individually.
As an example, system matrices for hover condition are reported below.



A										
u	v	w	p	q	r	φ	θ	ψ	a_1	b_1
0	0	0	0	0	0	0	-9,81	0	-10,228	0
0	-0,01502	-0,013553	-0,0040554	0	0,037235	9,787	0	0	0	10,228
0	0	-0,45108	0	0	0	0,67168	0	0	0	0
0	-0,030488	-0,065719	-0,0082317	0	0,075579	0	0	0	0	63,213
0	0	0	0	0	0	0	0	0	10,037	0
0	0,045924	0,0099171	0,0124	0	-0,11385	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0,99765	0,068469	0	0	0	0	0
0	0	0	0	-0,068469	0,99765	0	0	0	0	0
0,019225	0	0	0	-1	0	0	0	0	-32,24	0
0	-0,019225	0	-1	0	0	0	0	0	0	-32,24

Table 2: A system matrix in Hover



B			
δ_r	δ_{lon}	δ_{coll}	δ_{lat}
0	0	0	0
35,952	0	-36,431	0
0	0	-108,55	0
72,975	0	-17,665	0
0	0	0	0
-94,941	0	19,873	0
0	0	0	0
0	0	0	0
0	0	0	0
0	13,457	0	0
0	0	0	13,457

Table 3: B system matrix in Hover

3.11 Verification of the Linear Model

As the non-linear model is linearized, it would be desirable to verify that the linear and the non-linear model act in the same way to a given input [20]. This has been done by applying a step on all of the four inputs, one by one, and comparing the outputs from the non-linear and the linear models with the expected output. First the expectations of the trend of the linear and non-linear states are presented, after a simulation of the models with the given input is performed. This simulation is presented as plots of the states directly affected by the input.

3.11.1 Lateral cyclic positive step

We expect:

- v → positive
- p→ positive

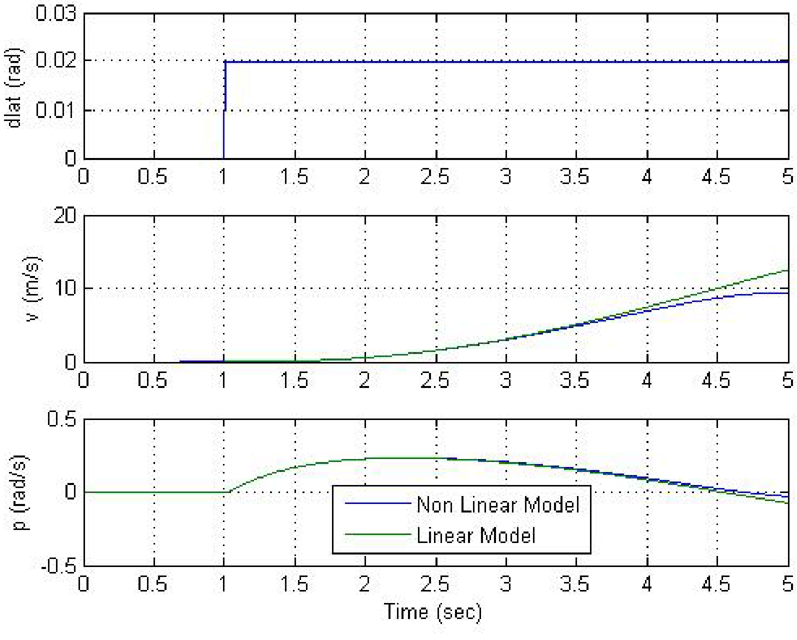


Figure 21: Linear model verification: lateral cyclic

3.11.2 Longitudinal cyclic positive step

We expect:

- u → negative
- q→ positive

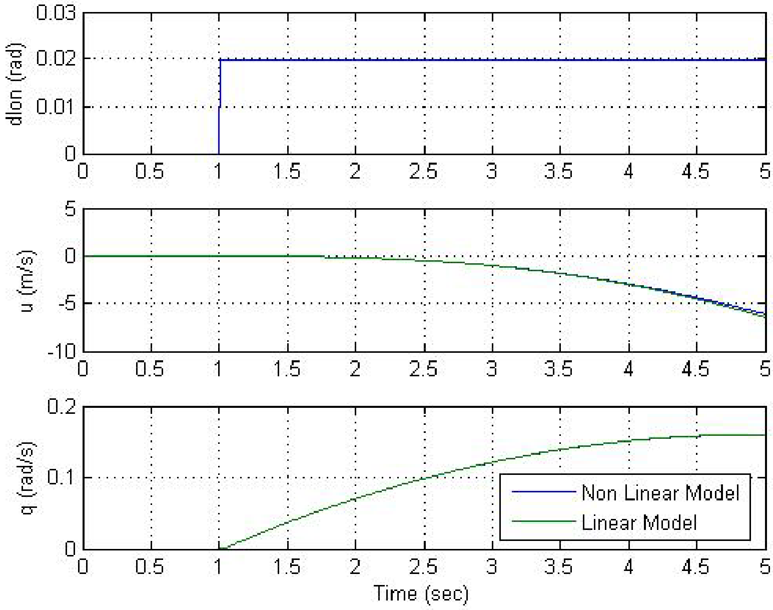


Figure 22: Linear model verification: longitudinal cyclic

3.11.3 Collective positive step

We expect:

- w → negative
- r → positive

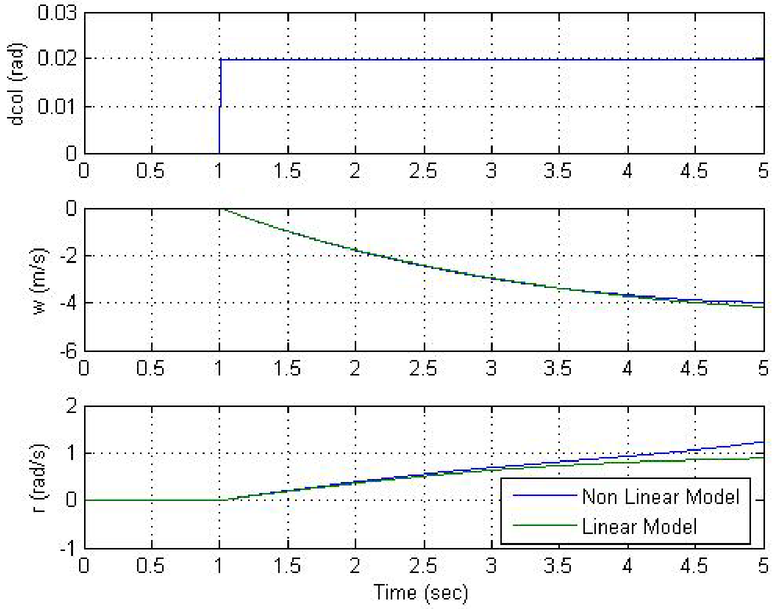


Figure 23: Linear model verification: collective

3.11.4 Pedals positive step

We expect:

- r → negative

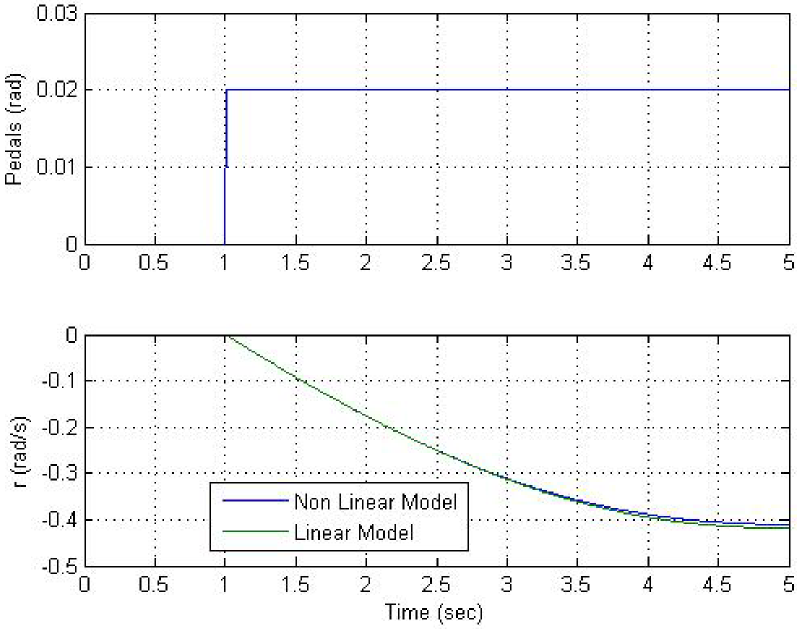


Figure 24: Linear model verification: Pedals

In all the tests, the changes in the states are the same for the two models thereby it is concluded that the linear model is describing the non-linear model adequately in the operating point.

4 Navigation Guidance and Control System

The following sections describe in details the NGCS main subsystems.

4.1 Stability Augmentation System and Autopilot

The Stability and Autopilot system design is based on a LQR (Linear Quadratic Regulator) approach [22] which will be described in the next sections.

4.1.1 LQR Overview

The purpose of this section is to give a short description of the linear quadratic approach to optimal control.
When controlling a dynamic system the aim is to bring the states from an initial position to a reference state, and keep them at this reference. Often it is desirable to drive the states to a steady state value as fast as possible, but this task will always be bounded by the amount of actuator power available. In continuous-time state-space model

$$\dot{x} = Ax + Bu \,,$$

by minimizing a quadratic performance index of the type:

$$J = \int_0^{\infty} \left\{ x^T Q_x x + u^T R u \right\} dt$$

an optimal sequence of inputs can be found for bringing the state x to the state reference. That implies to find the negative control feedback of the form:

$$u = -Lx$$

that minimizes the quadratic performance index described above.
The well-known solution is found in several textbooks as:

$$L = R^{-1} B^T K$$

where the gain matrix K is the Ricatti matrix, found from solving the steady-state Ricatti equation:

$$KA + A^T K - KBR^{-1} B^T K + Q_x = 0$$

The weighting matrices Q$_x$ and R were defined as diagonal matrices, with the elements equal to the inverse of the square of the maximum allowable deviations as follows:

$$Q_x(i,i) = \left(\frac{1}{\Delta x_{\max}(i)}\right)^2 ; \ R(i,i) = \left(\frac{1}{\Delta u_{\max}(i)}\right)^2$$

The performance of the controller after calculation of L is only optimal in relation to the chosen performance index. Therefore, the design parameters of the performance index, Q$_x$ and R, has to be chosen wisely to give the controller the right properties suiting the given control task. This is a trade-off between good control and good economy.

4.1.2 LQR Design

This section describes the design of a LQR based controller that stabilizes the helicopter and tracks the reference states. The command variables are the yaw rate, the body-axis forward, lateral and inertial vertical velocities.
For the linear model, the inertial vertical velocity has been calculated as:

$$V_h = u\sin(\theta_e) - w\cos(\theta_e)\cos(\varphi_e)$$

Where θ_e and φ_e are respectively the trim pitch and bank angle.
Furthermore to guarantee zero steady state error in tracking reference states the state vector has been augmented with integrators [23], which resulted in a new 15-dimensional state vector and corresponding state and control matrices.
The equations for the integral errors are:

$$\dot{u}_i = u^{cmd} - u$$

$$\dot{v}_i = v^{cmd} - v$$

$$\dot{r}_i = r^{cmd} - r$$

$$\dot{V}_{hi} = V_h^{cmd} - V_h$$

The resultant augmented state vector is:

$$x(t) = [u \ \ v \ \ w \ \ p \ \ q \ \ r \ \ \varphi \ \ \theta \ \ a_1 \ \ b_1 \ \ V_h \ \ u_i \ \ v_i \ \ r_i \ \ V_{hi}]^T$$

Note that ψ has been eliminated from the state vector and matrices since it doesn't affect any state (see Table 2).

The state feedback gain L was calculated using the LQR Matlab® command [21].

$$[L,K,e] = lqr(A,B,Q_x,R)$$

where e is the vector containing the closed loop eigenvalues.
For the problem at hand, to calculate Q_x , the following assumptions were made for the maximum allowable deviations in the outputs [24]:

$$\begin{aligned}
&\Delta u_{\max} = \Delta v_{\max} = \Delta w_{\max} = \Delta V_{h\max} = 0.1m/s;\\
&\Delta p_{\max} = \Delta q_{\max} = \Delta r_{\max} = 0.05rad/s;\\
&\Delta \theta_{\max} = \Delta \varphi_{\max} = 0.05rad;\\
&\Delta a_{1\max} = \Delta b_{1\max} = 0.05rad\\
&\Delta u_{i\max} = \Delta v_{i\max} = 0.1m/s\\
&\Delta r_{i\max} = 0.05rad/s
\end{aligned}$$

Some of the above requirements may be very demanding, with, as result, high gains and much control activity to achieve this.
For the maximum allowable control deflection deviations, to calculate R, the following was assumed [24]:

$$\begin{aligned}
&\delta_{r\max} = 0.15rad\\
&\delta_{lon\max} = \delta_{lat\max} = 0.05rad\\
&\delta_{col\max} = 0.09rad
\end{aligned}$$

With these assumptions the solution (L matrix) was computed for each trim condition (from hover to 30 m/s by step of 5 m/s). The feedback resulted in a stable system. As an example, the open-loop and closed-loop system eigenvalues at hover and at 30 m/s, that have a real part equal to -3 or more, are given in Figures 25 and 26.

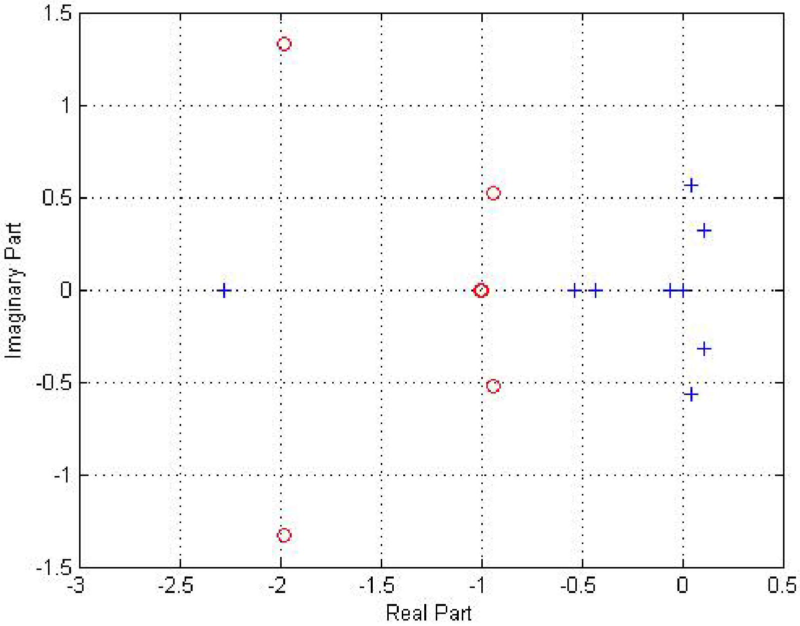


Figure 25: Open-loop (+) and Closed-loop (O) hover eigenvalues

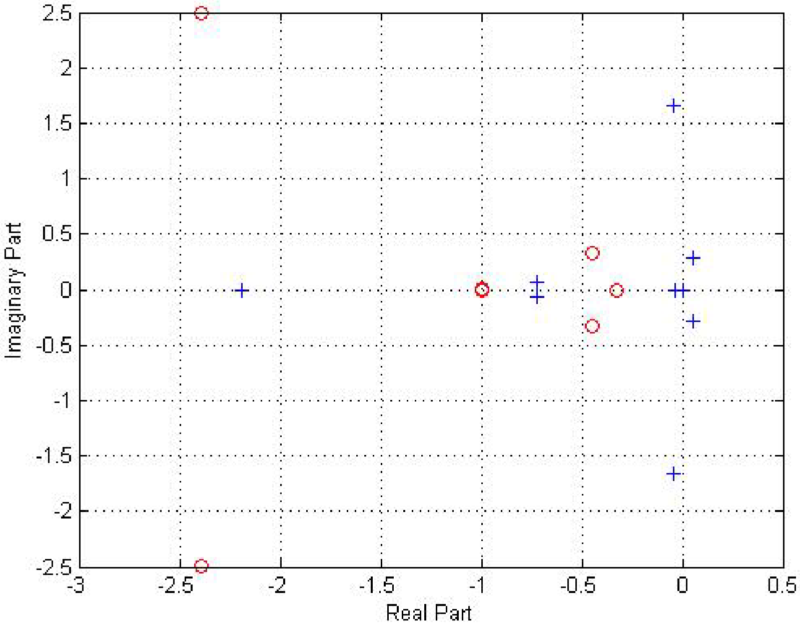


Figure 26: Open-loop (+) and Closed-loop (O) 30 m/s forward speed eigenvalues

4.1.3 LQR Verification on the Linear Model

First an offset initial state-value is chosen as test scenario [20]. This equates to the situation, where the controller is initialized when the helicopter is not in a trim condition. Then a tracking test has been carried out for each commanded variables [20]. Both tests have been performed for all the 7 linear models computed between hover and 30 m/s forward speed. For simplicity only the results for the hover model are reported here.
Figure 27 shows hover offset closed-loop simulation performed with initial values of:

- u=v=V_h=2 m/s
- $\varphi = \theta$ =0.35 rad

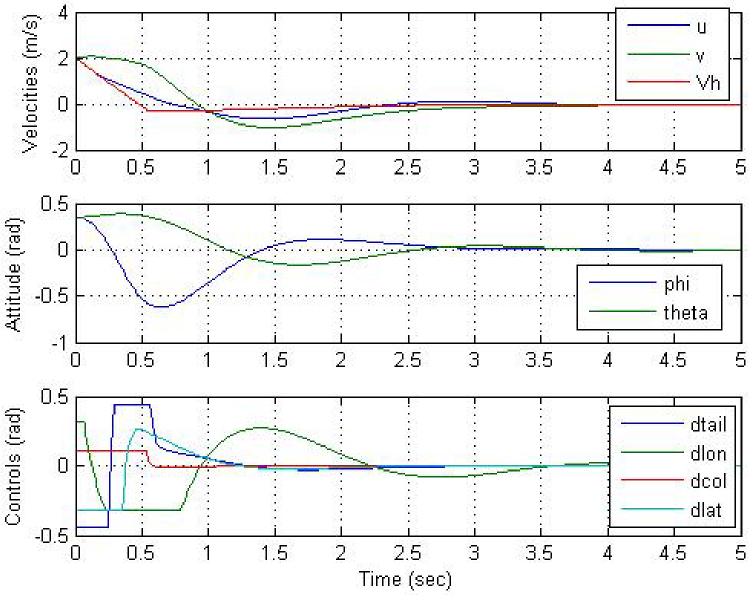


Figure 27: Hover closed-loop offset simulation

The results give an indication of how well the controller is able to stabilize the system. The controller proved able to handle simultaneous initial values at periphery of operating range. When considering speed performance, expressed as the time it takes for the controller to bring the states sufficiently near the equilibrium, it is more difficult to reach a final conclusion. The results state that the helicopter reaches hover within 5 second. These results cannot be properly evaluated because no requirements were set up in this project to the maximum allowable value of this time. However, the values seem reasonable.

Figure 28 shows controller tracking performance near hover conditions. The commanded variables have been set to:

- $u_{cmd}=v_{cmd}=V_{hcmd}=1$ m/s
- $r_{cmd}=0.1$ rad/s

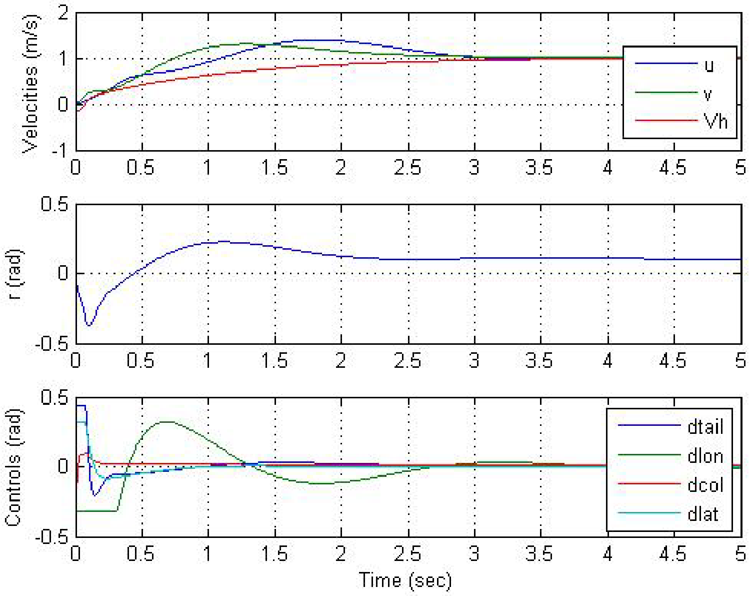


Figure 28: Hover closed-loop tracking simulation

Based on the test results the controllers seem to perform satisfactory on the linear models. However, the conducted tests can not be used to make any general conclusions regarding the controller's performance when it will be applied on the non-linear model. If any conclusion is to be made, it is that the controller displayed satisfactory performance within the scope of the conducted tests.

4.1.4 LQR Implementation and Verification on the non-Linear model

The Stability and Autopilot system for the non-linear model was obtained by interpolating the 7 gain matrices L (Section 3.10.2) with body axis forward velocity. This procedure was described and verified in [23].

In Figure 29, as an example, is reported the trend of the gains, L(1,6), L(2,1), L(3,3) and L(4,2), with forward velocity. From the physics point of view:

- L(1,6) directly correlate the yaw rate error with the pedals
- L(2,1) directly correlate the x body axis velocity error with the longitudinal cyclic
- L(3,3) directly correlate the z body axis velocity error with the collective
- L(4,2) directly correlate the y body axis velocity error with the lateral

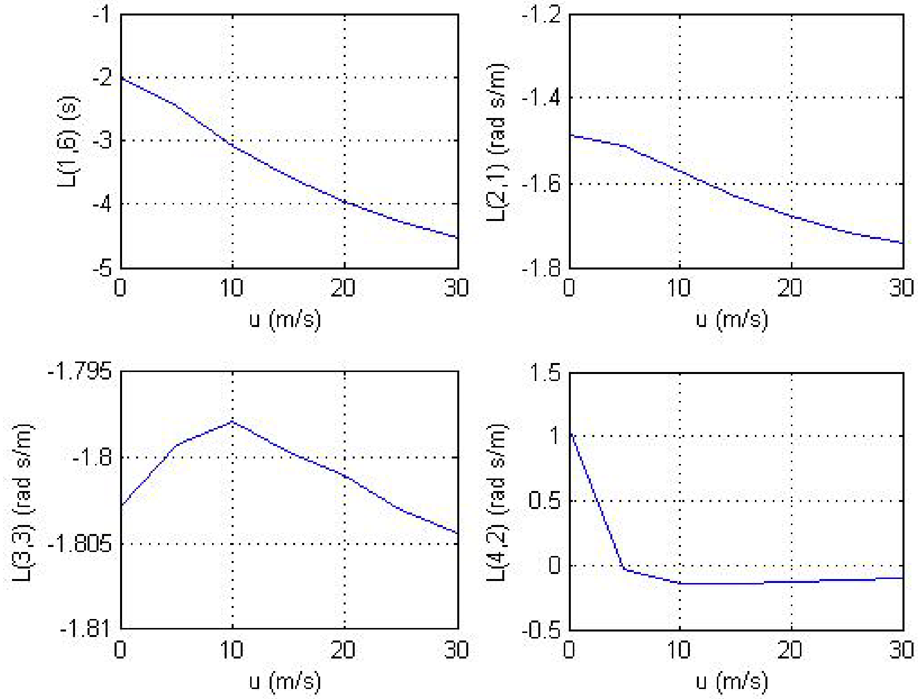


Figure 29: Gains trend with forward speed

The Simulink™ final implementation of the Stability and Autopilot system is shown in Figure 30.

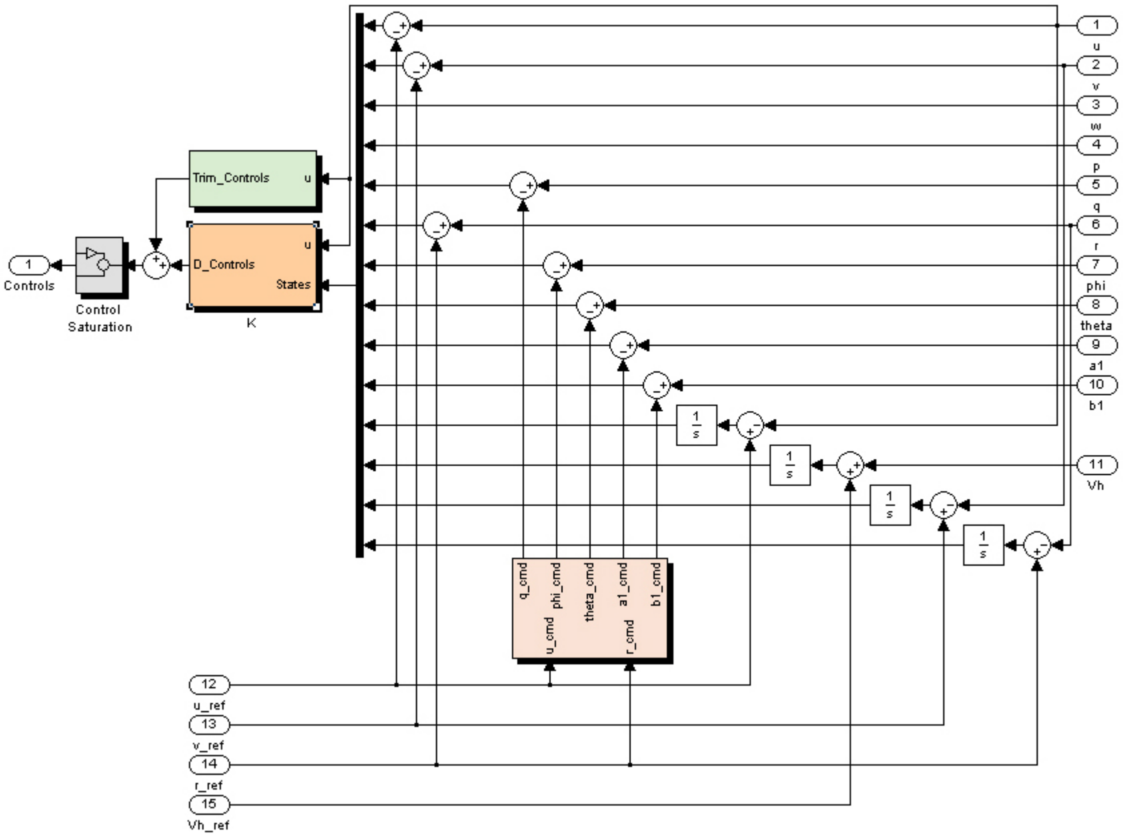


Figure 30: Simulink™ SAS and Autopilot implementation

In particularly, the controls at trim, calculated in Section 3.10.1, have been added to the commands elaborated by the controller. The controls are then saturated to the following maximum admitted travels [19]:

- lateral and longitudinal cyclic → ±18°
- collective → -3°/+15°
- tail rotor →±25°

Offset initial state-values and state tracking tests has been extensively conducted within the whole flight envelope to verify the performance of the stabilization and autopilot system. The results of the tests showed that the controller has satisfactory performance.
As an example, Figure 31 shows the comparison of tracking performance between the linear and the non-liner model at near hover condition. The commanded variables have been set to:

- $u_{cmd}=v_{cmd}=V_{hcmd}=1$ m/s
- $r_{cmd}=0.1$ rad/s

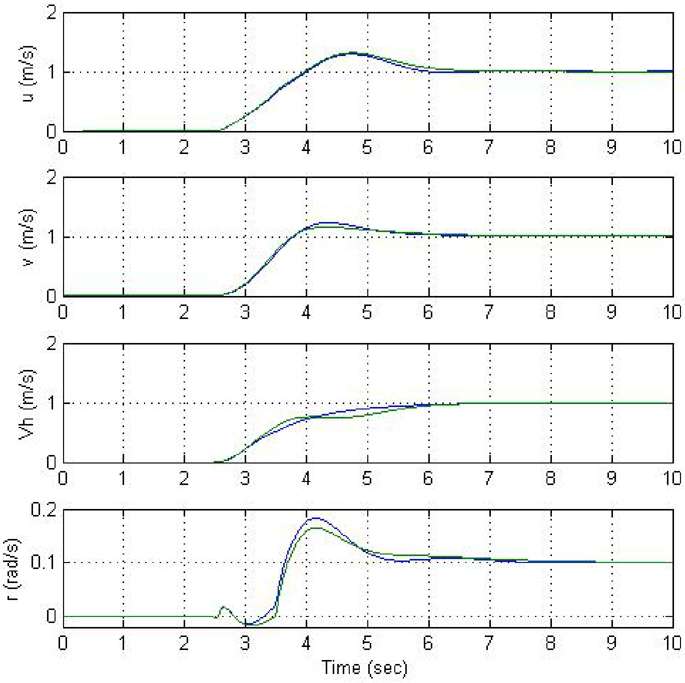


Figure 31: Closed-loop Linear vs. non-Linear model

Figure 32 shows the good system stability during acceleration at 1 m/sec^2 from hover to 30 m/s.

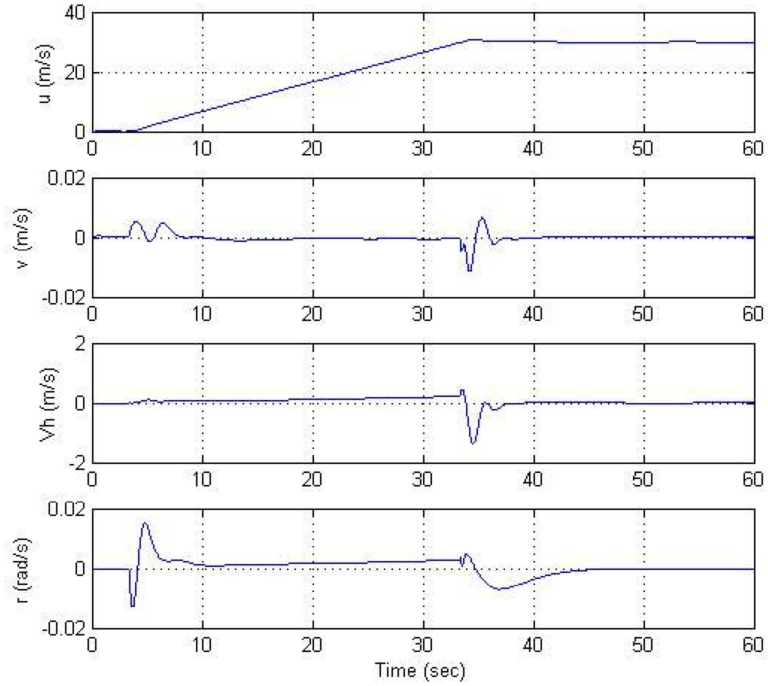


Figure 32: Closed-loop Non-linear model acceleration

4.2 Engine Governor

The engine governor manages helicopter throttle in order to maintain constant rotor RPM.

In the absence of manufacturer data, the governor can be modeled as a proportional-integral feedback controller [7], maintaining commanded rotorspeed by changing the throttle:

$$\delta_t = K_p(\Omega_c - \Omega) + K_i\omega_i$$

$$\dot{\omega}_i = \Omega_c - \Omega$$

where Ω_c is the commanded rotorspeed, K_p and K_i are proportional and integral feedback gains. Throttle servo dynamics is much faster than the rotorspeed dynamics, and was neglected in the model. Using Ziegler and Nichols method [25] and fine tuning we have found:

- K_p=0.1
- K_i=0.02

Figure 33 shows the throttle response and the rotorspeed tracking (96.3 rad/s) after a commanded climb at 2 m/s.

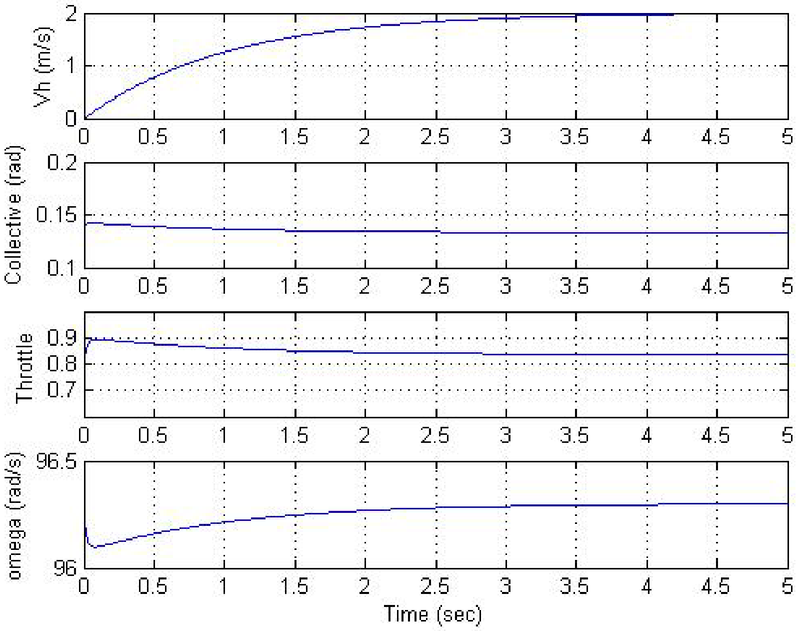


Figure 33: Engine governor tracking performance

4.3 Guidance

The guidance system is composed by two main parts: the lateral track control and the altitude controller.

4.3.1 Altitude Controller

The altitude-hold is a simple proportional integral controller. It takes as input the destination waypoint altitude and the current vehicle altitude and gives as output the vertical velocity to maintain or reach the reference altitude. Using Ziegler and Nichols method [25] and fine tuning we have found:

- K_p=1.2
- K_i=0.05

Figure 34 shows an example of altitude tracking.

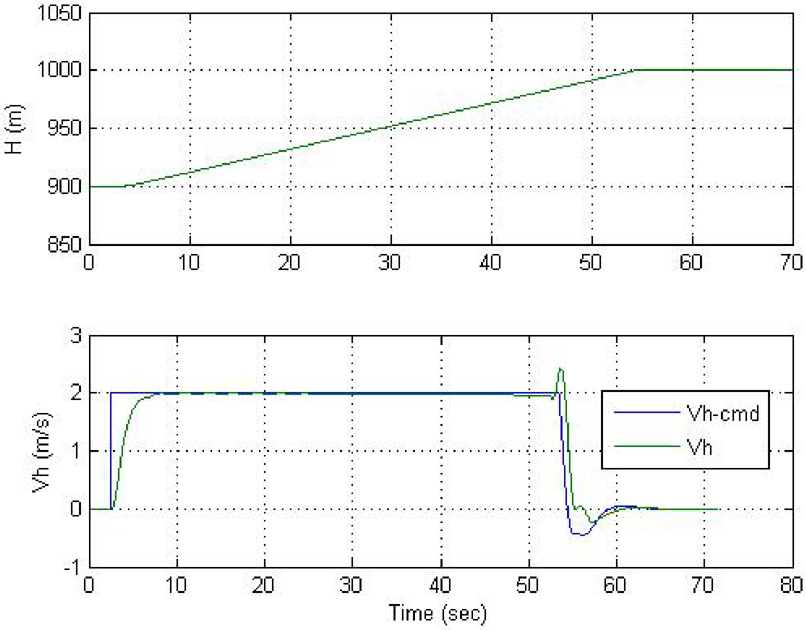


Figure 34: Altitude controller tracking performance

The initial helicopter altitude is 900 m while the commanded altitude is 1000 m. Therefore the altitude controller commands a vertical velocity (saturated at 2 m/s) until the reference altitude is captured.

4.3.2 Lateral Track Controller

The primary design objective of the lateral controller is to intercept and track a specified flight plan segment by means of a yaw rate command [26].
Let's consider the helicopter in level flight at an arbitrary position relative to the track line between way points W_{p1} and W_{p2}, and flying on an arbitrary heading Ψ. We are interested to obtain the position and velocity components in the X_{track}, Y_{track} reference frame (see Figure 35). The transformation will be a rotation of an angle $(\Psi_{12} - \pi/2)$ and the associated rotation matrix is given by:

$$T_{\psi} = \begin{bmatrix} \cos(\psi_{12} - \pi/2) & -\sin(\psi_{12} - \pi/2) \\ \sin(\psi_{12} - \pi/2) & \cos(\psi_{12} - \pi/2) \end{bmatrix}$$

where Ψ_{12} is the track line heading, in radians, between W_{p1} and $W_{p2.}$

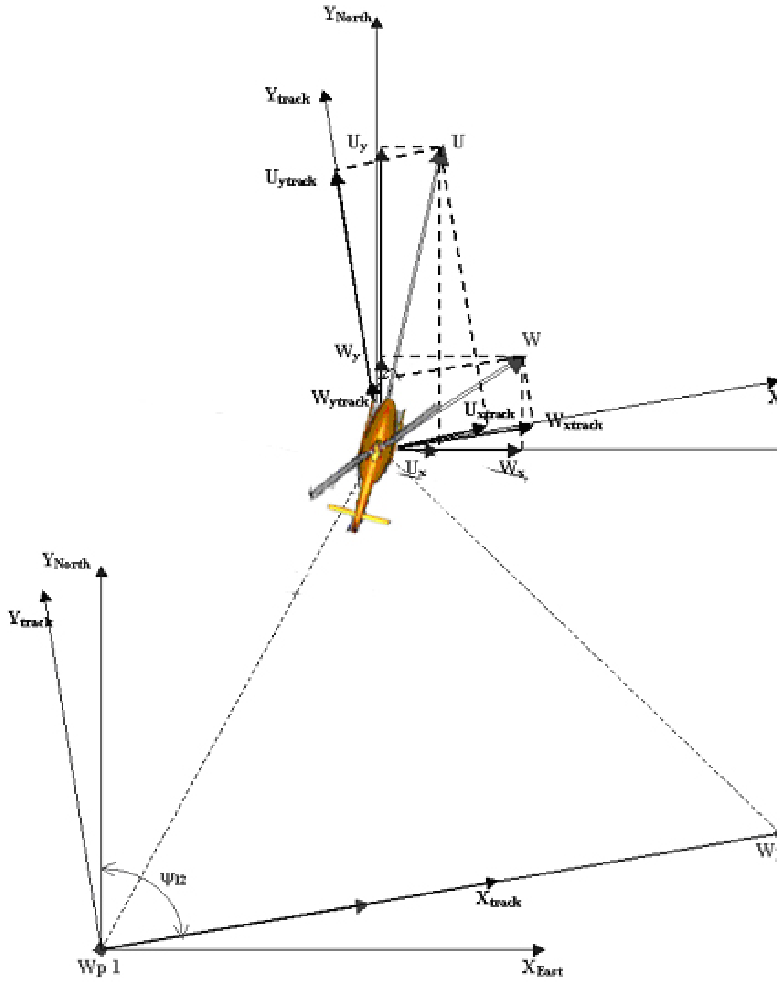


Figure 35: Track reference frame

Applying the above rotation to the North and East helicopter ground speed vector (V_N,V_E) we obtain:

$$\begin{bmatrix} \dot{X}_{track} \\ \dot{Y}_{tracx} \end{bmatrix} = T_{\psi} \begin{bmatrix} V_N \\ V_E \end{bmatrix}$$

The X_{track} and Y_{track} helicopter position can be found solving the differential equation system reported above with initial conditions:

$$\begin{bmatrix} X_{track0} \\ Y_{tracx0} \end{bmatrix}$$

Knowing the current track position (X_{track}, Y_{track}) of the helicopter from the destination way point W_{p2}, the control strategy is to point the vehicle ground speed vector in the direction of the track intercepting the track-line at point C. The intercept point C is determined by a design parameter k where the distance on the track line from the intercept point C to the way point W_{p2} is at any instant of time equal to (1−k) X_{track}. From the geometry of the similar triangles OAB and OCD (Figure 36), a new control strategy is proposed based on establishing the helicopter position and velocity according the following relationship:

$$\frac{\dot{X}_{track}}{kX_{track}} = \frac{\dot{Y}_{track}}{Y_{track}}$$

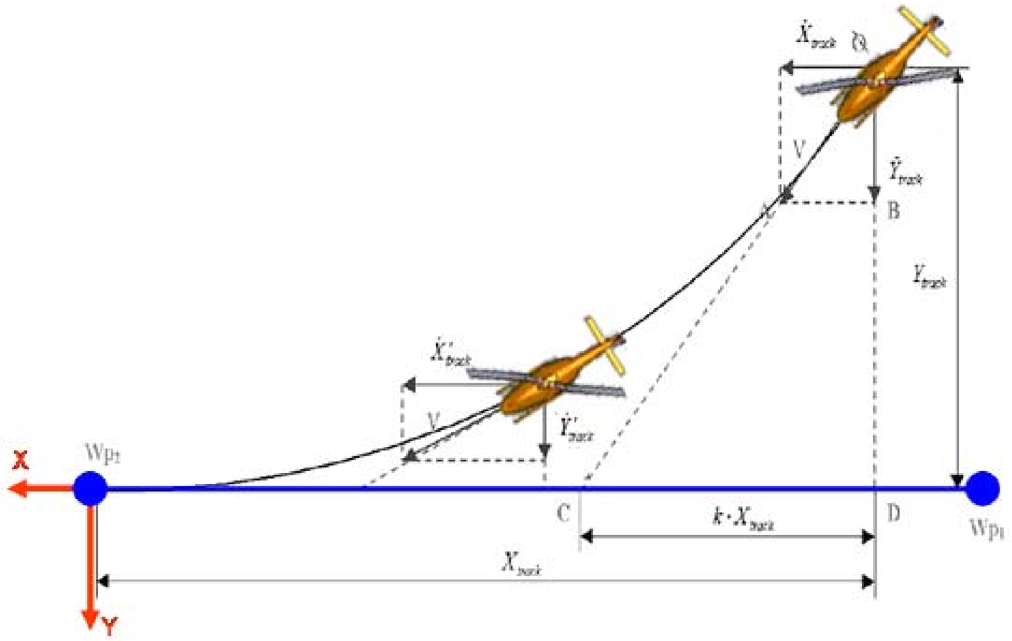


Figure 36: Guide control strategy

To achieve this objective, the error E given by:

$$E = kX_{track}\dot{Y}_{track} - \dot{X}_{track}Y_{track}$$

is to be driven to zero, using the proportional feedback control law that expresses yaw-rate commands as:

$$r_{cmd} = K_r E = K_r\left(kX_{track}\dot{Y}_{track} - \dot{X}_{track}Y_{track}\right)$$

The proportional gain K_r is determined iteratively through simulation until good tracking is achieved with virtually no overshoot. A value of K_r = -0.0005 and a value of k=0.1 were found to be satisfactory in our lateral track control law. The yaw rate command has been saturated at ±0.2 rad/s and filtered to eliminate the limit-cycle problem.
A Simulink™ block diagram that realizes the control scheme described is shown in Figure 37

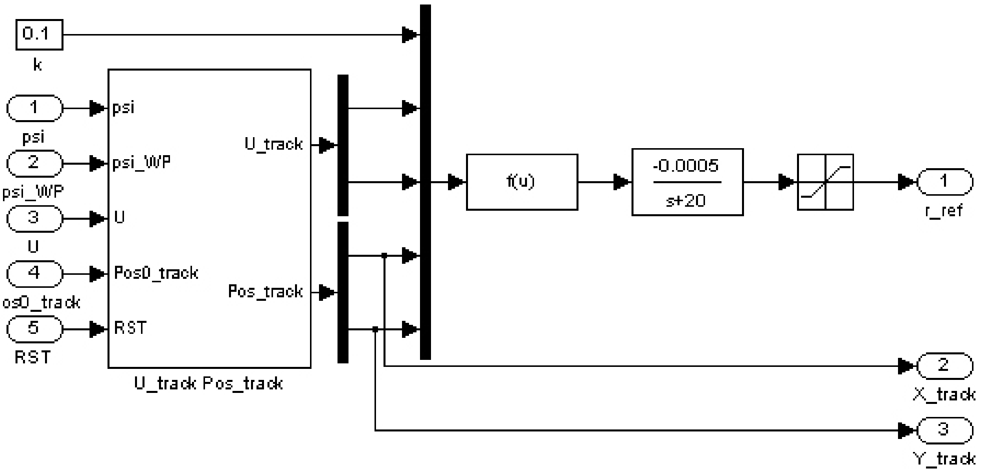


Figure 37: Simulink™ guide implementation

The simulation reported in Figures 38, 39 and 40 show an example of guide tracking performance.
The helicopter is supposed to be in hovering at 1000 m altitude and 44.01 N and 12.01 E position with heading North. Then the RUAV is asked to fly at 20 m/s, constant altitude and to follow the track described by:

- W_{p1} 44 N 12 E
- W_{p2} 44.03 N 12.05 E

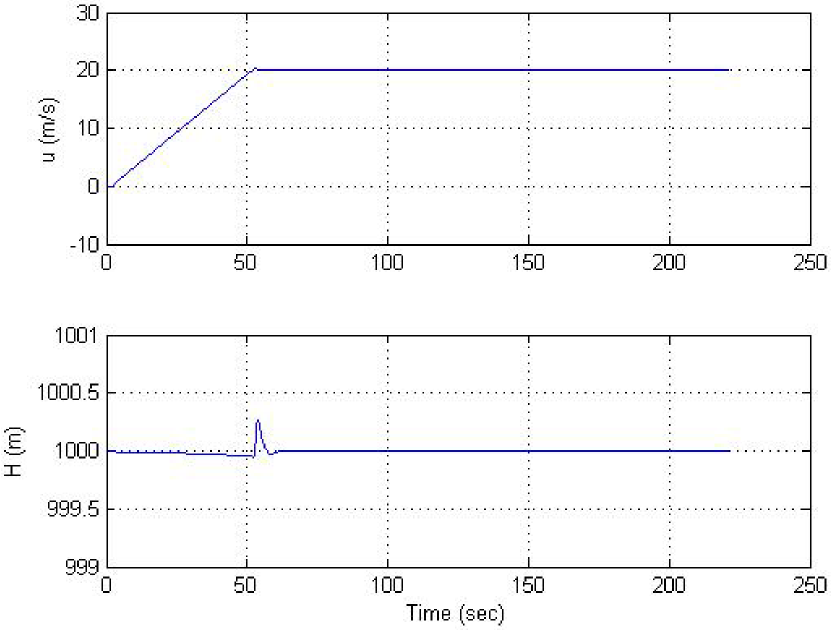


Figure 38: Guide tracking performance – Speed, Altitude

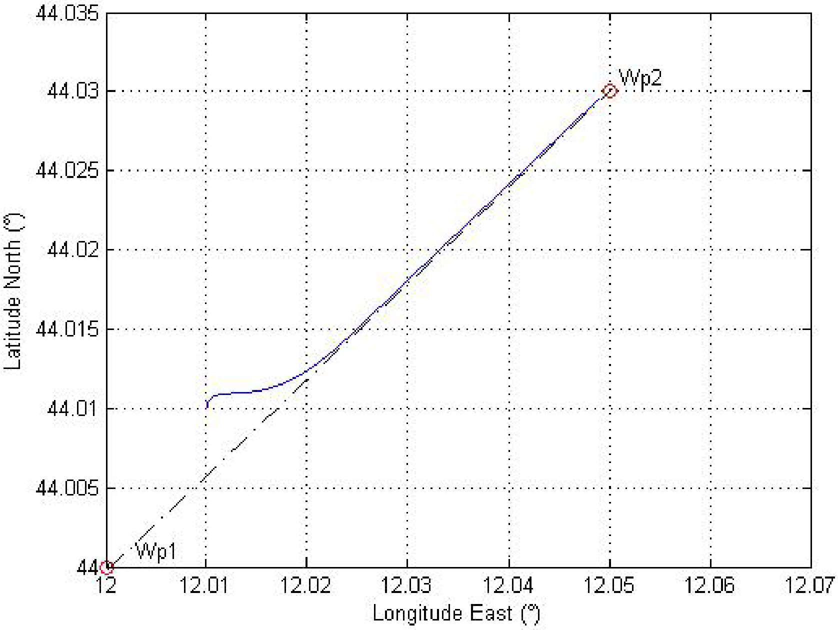


Figure 39: Guide tracking performance – Latitude, Longitude

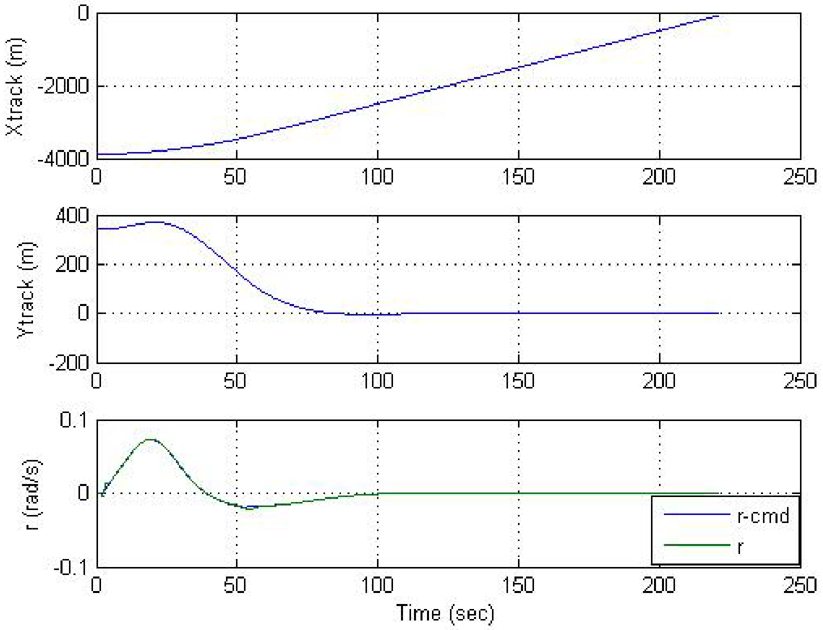


Figure 40: Guide tracking performance – X_{track}, Ytrack, Yaw Rate

The proposed design lateral track control handles also wind cases in a simple manner and ensures track stability over a wide set of initial conditions [26].

In manual flight control mode, the guidance system is disabled: the operator at the ground controls body axis forward speed, vertical velocity, side sleep velocity and yaw rate through joystick input commands. In that case the joystick commands are sent directly to the autopilot.

5 Ground Control Station

The work involved in this part of the project has been the design and development of a ground control station for real time control and display of RUAV flight test data [5]. The GCS is the hub of an unmanned air vehicle. It is what processes the incoming data and sends control instructions to the air vehicle. Typically a GCS will envelope three main functions: mission planning, mission control and data manipulation. The level to which each of these functions are implemented in a particular system, depends on the level of system autonomy. Generally the level of autonomy and the system mission dictate the GCS architecture. For CAPECON, UNIBO has implemented a simplified GCS which is able to operate the RUAV in both autonomous or remote piloted flight. The GCS has been design to be easily modified for controlling and monitoring of a real RUAV.

Therefore, it includes a visual system which induces a sense of presence in the engagement area, provides a multi-modal input interface, including head tracker and joystick, which enables efficient interactions and could be also transportable to the location of the test flights.

Key problems to be solved have been:

- the interfacing of the different hardware and software components of the system
- the development of the graphic interface for mission and flight control
- the development of a visual system for modular mission payload simulation and for a data-driven “virtual view” of the flight vehicle, displaying its current position

The basic software has been developed through the Labview data acquisition, control and visualization software. The Labview software has been chosen due to its quick and flexible applications. The Labview code has been implement on the primary master computer of the GCS and is able to manage:

- communication between the Simulink model of the air vehicle and the master computer of the GCS
- communication between the visual system, developed in C++ code and the primary master computer of the GCS
- the graphic interface for mission and flight control

It is constituted by different blocks:

- a “read loop” which receives data from the RUAV simulator via TCP/IP communication using the Labview Simulation Interface Toolkit 2 Blockset
- a “data selection block” which is able to split the data, received at the GCS primary master computer, into three main cluster of data to be displayed on the GCS graphic interface: a “cluster to visual” data, a “cluster to map” data and a “cluster to virtual cockpit” data
- two graphic blocks have been created for generating real-time plots of various flight parameters, animated map display, flight plan window and virtual cockpit
- a “joystick manager” block for receiving joystick signals from the USB port when the remote piloted flight mode is active
- a “send loop” to the visual system for displaying data on a 3D graphical interface which uses an UDP communication protocol
- a “send loop” to the air vehicle computer for sending real time control signal. In remote piloted flight mode the control signals comes from the joystick interface

while in autonomous flight mode the control-navigation signals depend on the flight plan entered through the flight plan graphical interface.

The GCS actual layout is shown in Figure 41.

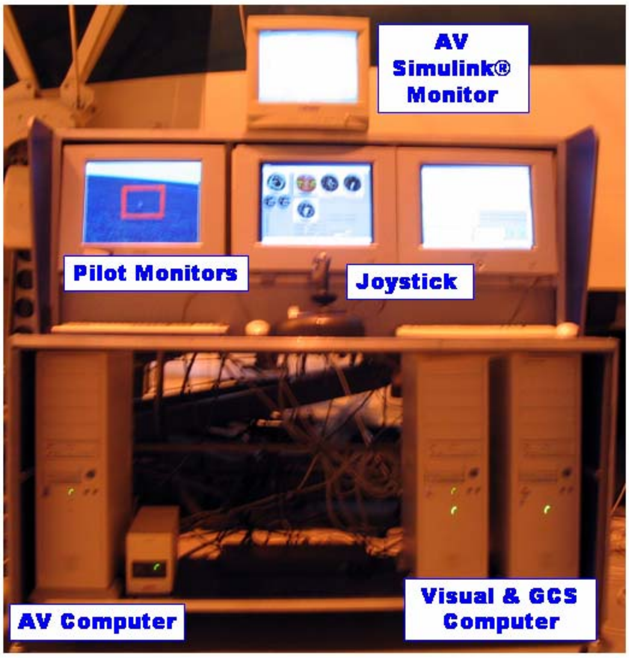


Figure 41: GCS actual layout

6 Mission Simulation Evaluation

The mission simulation evaluation has been performed for the CAPECON Agusta helicopter configuration [19], based on the Simulink™ dynamics model and NGCS developed. The configuration has been tested for a standard "ship" search mission within an area of 50 km.

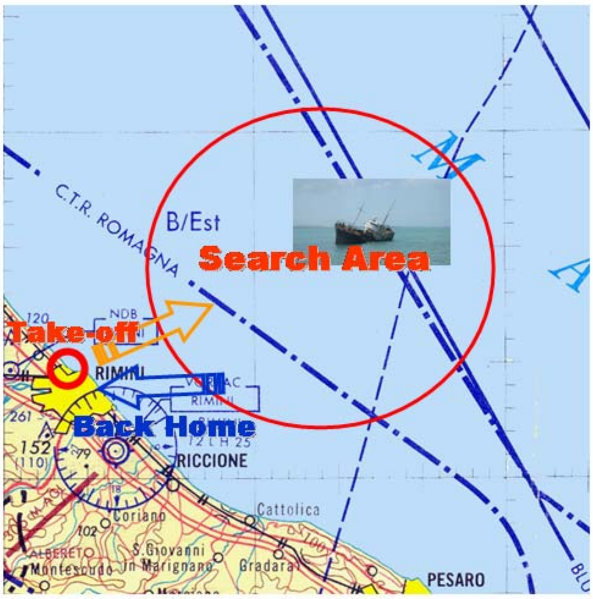
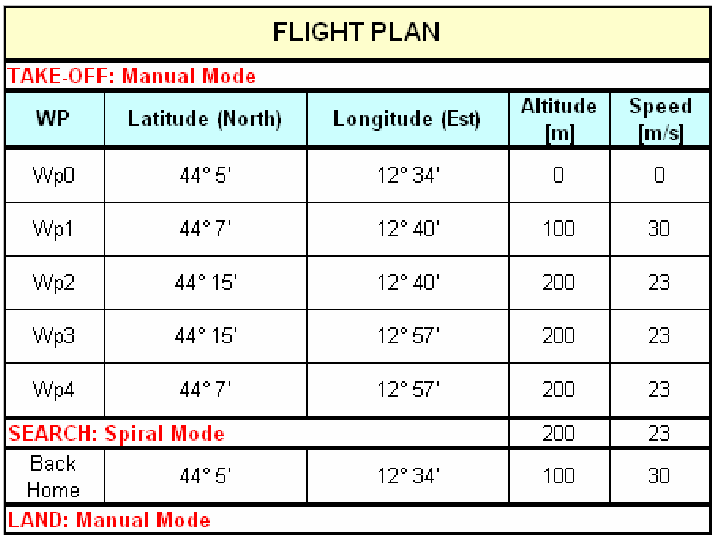


Figure 42: Mission scenario

The air vehicle is supposed to take-off in manual mode and then follow in autonomous mode the flight path and the flight plan as described in Figure 43 and in Table 4.



FLIGHT PLAN				
TAKE-OFF: Manual Mode				
WP	Latitude (North)	Longitude (Est)	Altitude [m]	Speed [m/s]
Wp0	44° 5'	12° 34'	0	0
Wp1	44° 7'	12° 40'	100	30
Wp2	44° 15'	12° 40'	200	23
Wp3	44° 15'	12° 57'	200	23
Wp4	44° 7'	12° 57'	200	23
SEARCH: Spiral Mode			200	23
Back Home	44° 5'	12° 34'	100	30
LAND: Manual Mode				

Table 4: Flight Plan data

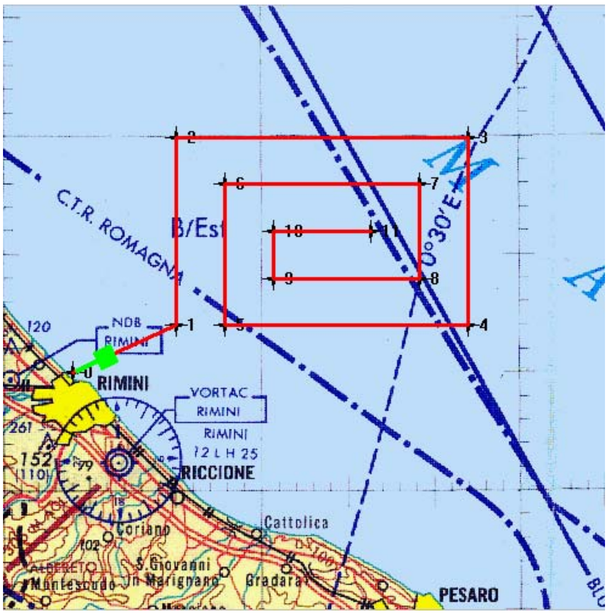


Figure 43: Search Mission flight path

The operator at the ground can monitor the search area by means of a simulated gyro-ball camera. If something is found the operator can switch in manual mode or modify the flight plan in order to survey and better control the situation.

In order to evaluate the mission operational capabilities of the RUAV, it was supposed to find a ship at a certain point along the mission path. In that case the operator at the ground has taken the air vehicle control switching in manual mode for monitoring the situation. The actual flight path together with the mission vertical profile is shown in Figure 44, 45.

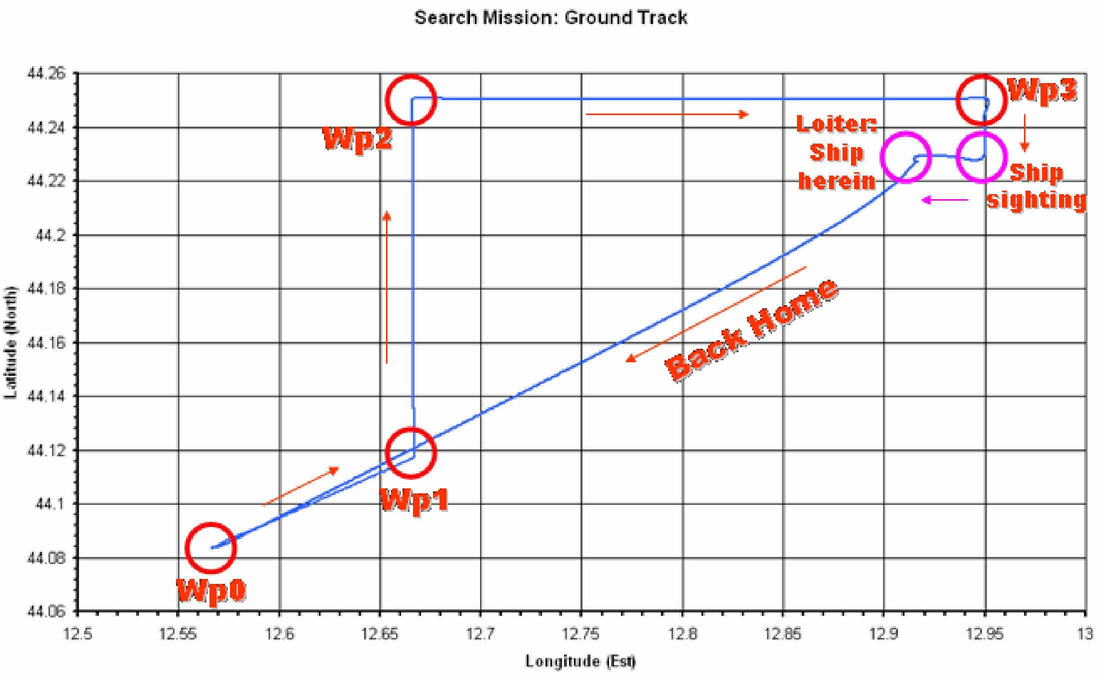


Figure 44: Search Mission actual flight path

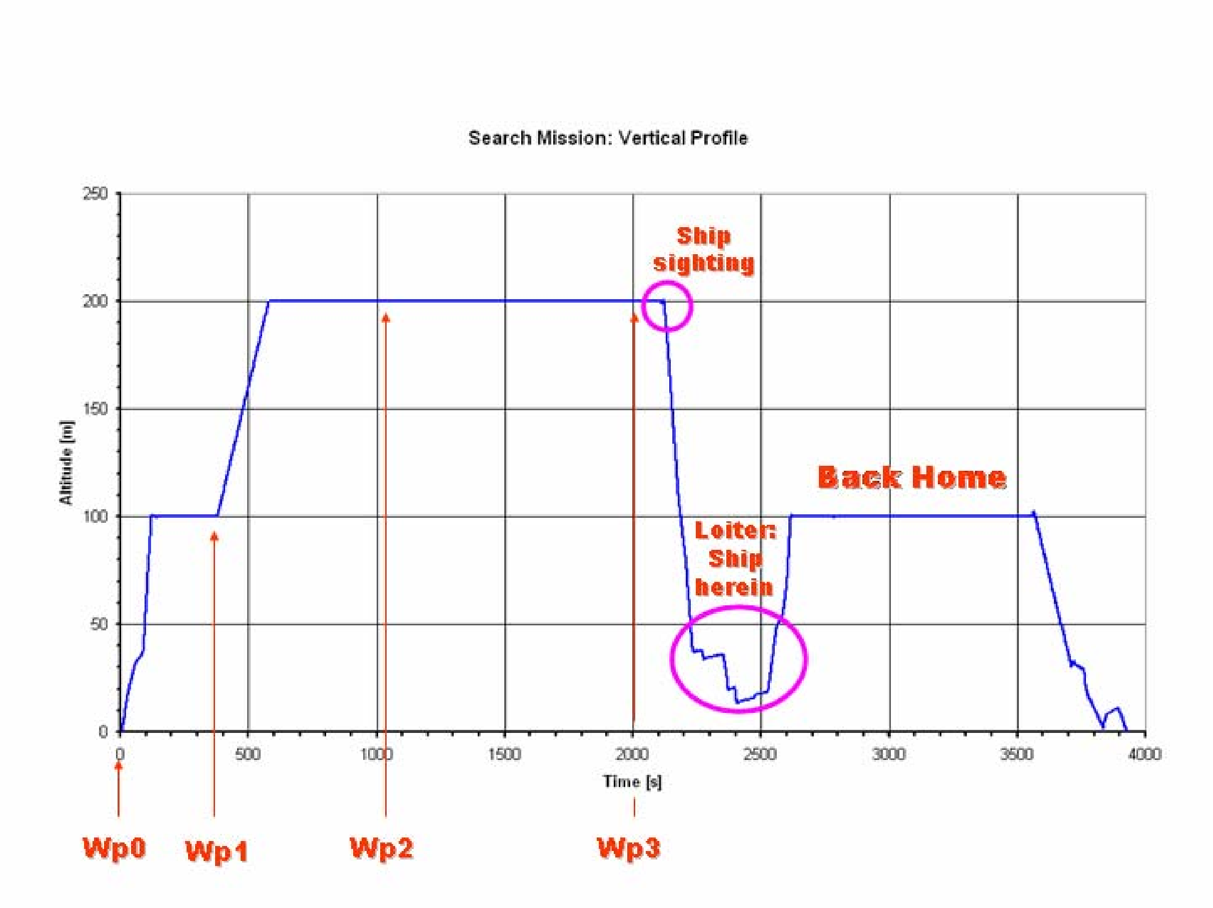


Figure 45: Search Mission vertical profile

Once the air vehicle was nearby the ship, a manual descent and loiter have been performed in order to have a better situation overview and send ship position and video images at the ground control station. After a detailed survey, the “back home” key on the GCS control panel allows the operator to directly drive the air vehicle to base.
If nothing is found the RUAV is supposed to cover the pre-planned spiral flight path (Figure 43) and then come back to base. The search area is reached at high speed while the spiral path is performed at the best endurance speed. At every instant of time the operator at the ground can always fly back the air vehicle by means of the “back home “ key. The landing maneuver is always done in manual flight mode.
Other post-processed data are reported in the following figures.

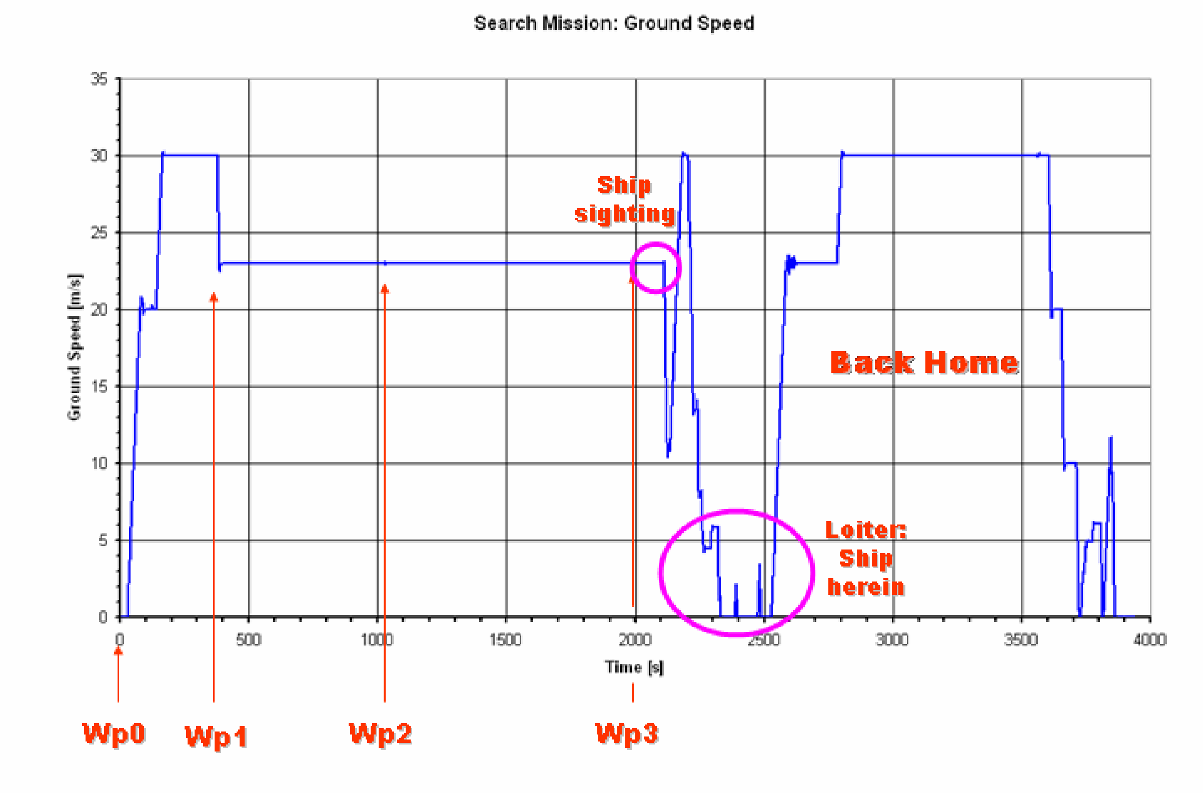


Figure 46: Search Mission ground speed

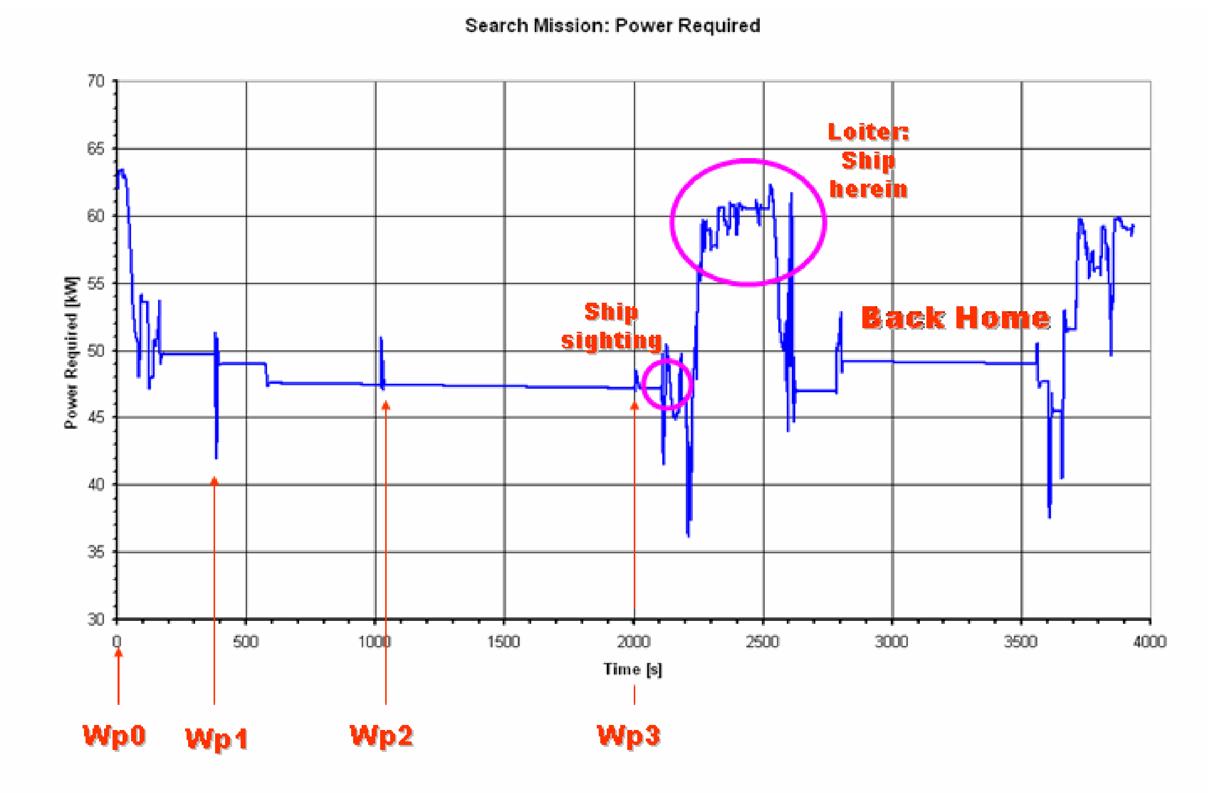


Figure 47: Search Mission power required

Figure 47 shows the power required during each mission phase.
The total fuel consumption is about 22 lt (Figure 48).

The total fuel consumption has been also evaluated in the worst case (ship at the center of the spiral, ten minutes hover and flight back home) as shown in Figure 49: A fuel consumption of 39 lt. (over the 50 lt. available) is found to be necessary to cover the whole mission.

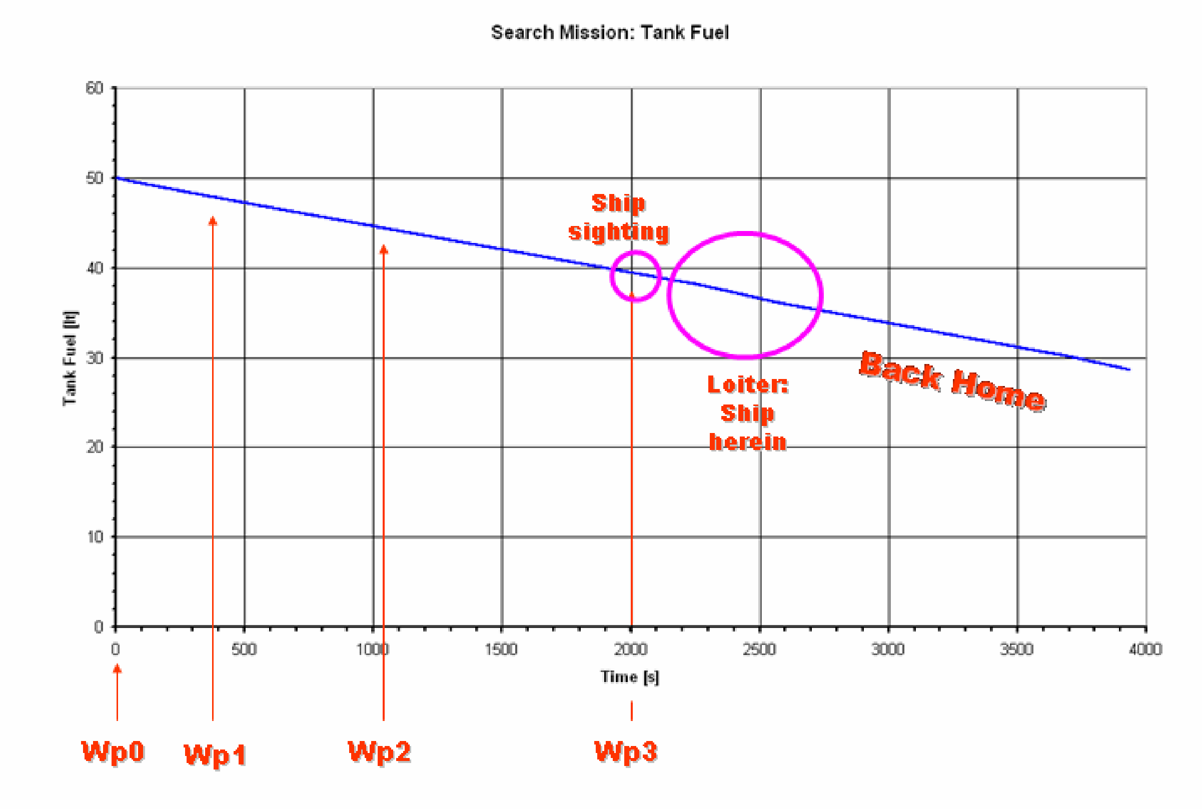


Figure 48: Search Mission fuel consumption

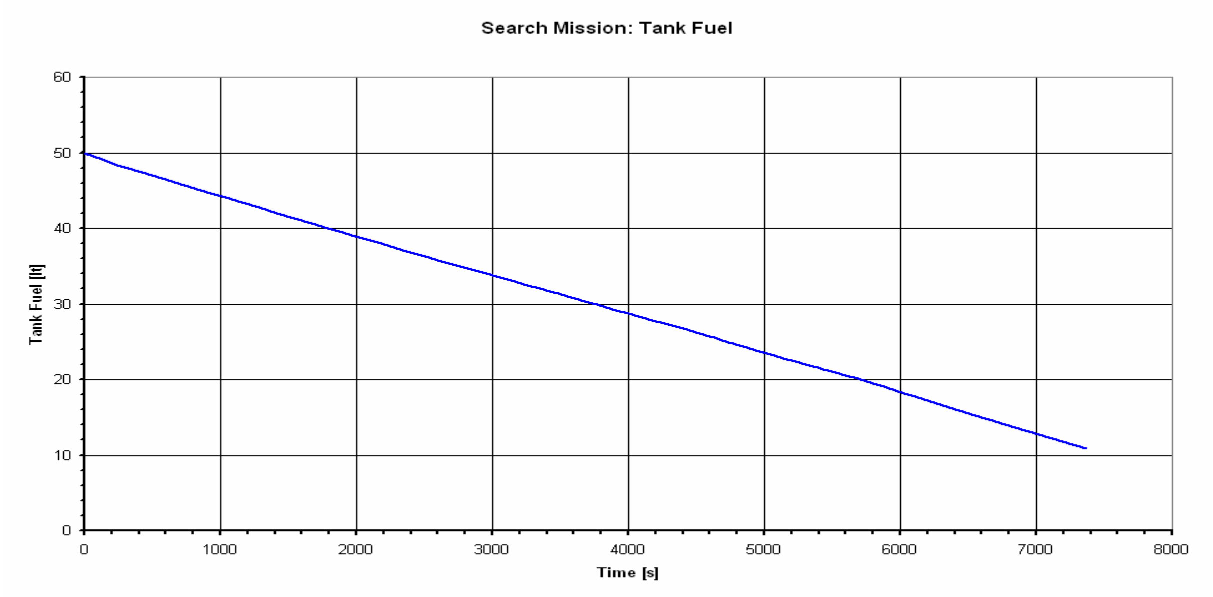


Figure 49: Pre-planned Search Mission fuel consumption

7 Mission Simulation Environment: Conclusion

A RUAV mission simulation environment has been designed and developed at UNIBO laboratories including the ground control station and the development of the conventional helicopter flight dynamics simulator and the related NGCS system. It demonstrated to be a reliable tool for testing mission feasibility. The CAPECON Agusta configuration has been tested for a search mission. The simulation results showed that the helicopter is able to perform the pre-planned mission by means of the NGCS designed.

Useful experience was gained in the design of helicopter math model and NGCS for UAVs/RUAVs.

A mission simulation environment, such as the one developed at UNIBO laboratories, can be an extremely useful tool in UAV researcher activities for minimizing, through the use of modeling and simulation, the time between UAV concept and operational evaluation. In this way, researchers are free to develop different concepts, controllers, intelligence or deployment strategies and test them in cooperative simulation.

The mission environment system makes it possible to integrate complex models, real components and ideas into real-time simulations. Connected to HIL (Hardware In the Loop), the system becomes a reliable test bench for researches in control laws, man-machine interfaces and system integration.

It can provide a tool to go from UAV concepts to reality with minimum time, costs and efforts.

.

Part II: UNIBO RUAV Project

8 UNIBO RUAV Project: Introduction

At University of Bologna (II Faculty of Engineering – Forlì Laboratories), a rotorcraft UAV has been developed, which can be used as flying test bed for researches in Unmanned Air Vehicle control and navigation laws; meanwhile it should be proposed as a technological prototype for industries interested in UAV development and manufacturing.
The goal of UNIBO RUAV project is to develop a helicopter platform capable of autonomous flight which could be used inside the Universities for researches in control and navigation laws, man-machine interfaces and system integration; meanwhile it should be proposed as a technological prototype for industries interested in UAV development and manufacturing. In order to develop such kind of platform, avionic systems are required that enable the helicopter to maintain a stable attitude and follow desired trajectories. This avionics package is comprised of sensors, computer and data link hardware as well as software to guide, navigate and control the air vehicle. These aspects are particularly critical for helicopters, which are well known to be inherently unstable systems, and place numerous requirements on the avionic system design.
The main requirements taken into account for the avionics package design were both operational requirements and physical constraints. From this point of view the instrumented platform should:

- provide accurate flight data acquisition for dynamic model development and validation
- allow onboard implementation of feedback control laws and demonstrate good control capability
- be endowed with an onboard safety system in event of computer failure
- be versatile enough to enable fast and easy integration of different input/output hardware and sensors
- be as light as possible in order to lower the total platform weight and maintain good maneuver capabilities. Flight test demonstrated that the helicopter still has good maneuverability with 6 kg payload mass
- be able to withstand the high vibration load typical of small scale helicopters. The primary sources of vibrations are the engine, the main rotor (spinning at roughly 22 Hz), the tail rotor and the tailboom bending resonance. These vibrations must be reduced to fit the operational vibration range of the onboard sensor and to provide accurate flight data measurements. Experimental tests performed with commercially manufactured elastomeric dampers showed that vibrations can be effectively reduced to the desired level
- be protected against the electromagnetic and RF interference: common shielding precautions were used to isolate the onboard electronics from EM interference

Since I have been one of the founders of this project I have worked in any field ranging from helicopter assembly to hardware integration from software development to flight tests. During the last year I have been involved in the development of the autopilot to be implemented on the avionics system. The aim was to demonstrate the possibility to achieve autonomous flights capabilities using exclusively off-the-shelf equipments.

9 Air Vehicle and Avionics Description

The test vehicle, shown in figure 1, is a Hirobo Eagle II 60 hobby helicopter which was modified to accommodate the avionics hardware. A more powerful engine, longer fiberglass blades, longer tail boom and tail blades were mounted in order to increase the helicopter payload carrying capabilities [27]. The rotor diameter is 1.84 m and the platform total mass is about 11.2 kg. The assembly also includes a Bell-Hiller stabilizer bar, which augments servo torque with aerodynamic moment to change the blades cyclic pitch and adds lagged rate feedback to improve the helicopter handling qualities.

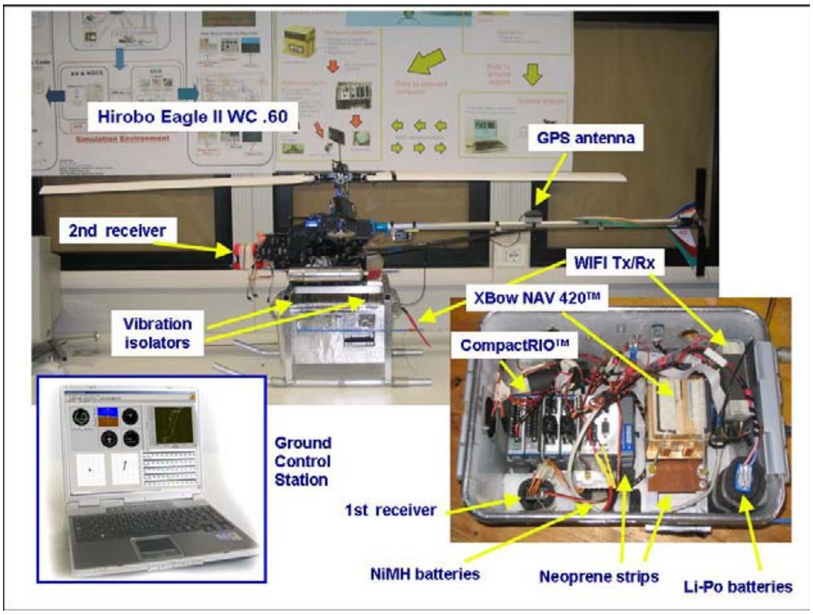
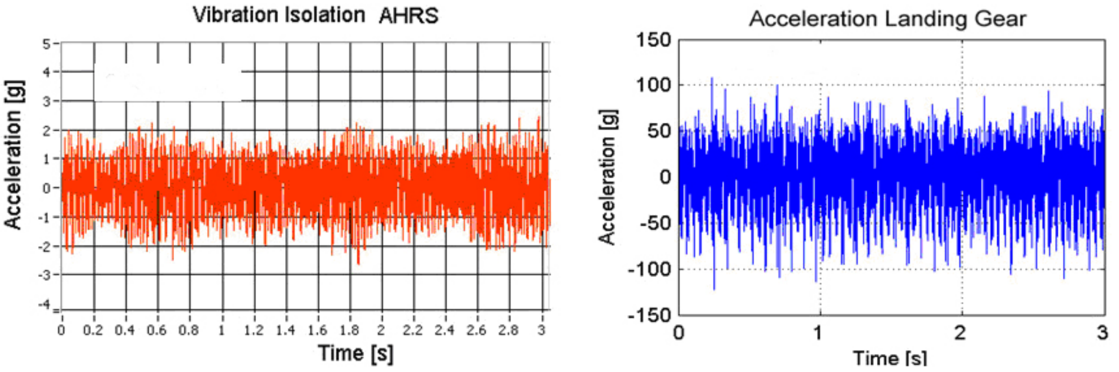


Figure 50: RUAV hardware

The design constraints were met with a 4.5 kg aluminum shielded avionics box (Figure 50) mounted on a customized landing gear and suspended with elastomeric isolators. The suspension system effectively attenuates vibration inputs from the main rotor and the engine to a level well within the operational vibration range of the avionics package (Figure 51).



CRIO Operating Vibration Range : < 5 g rms 10Hz-500Hz

NAV 420 Operating Vibration Range: < 6 g rms 20Hz-2kHz

Figure 51: RUAV operating vibration range

The complete avionics architecture is shown in Figure 52 and better described in [27].

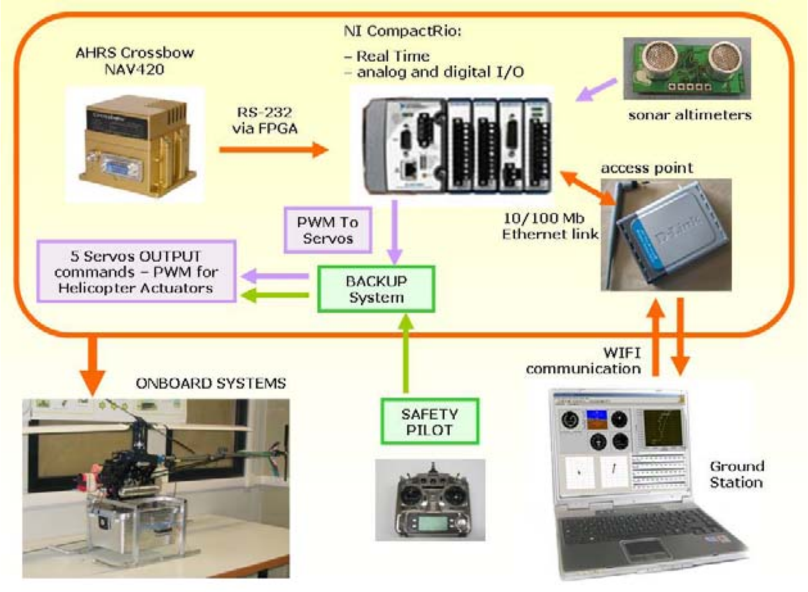


Figure 52: RUAV system architecture

The NI CompactRIO system was chosen as flight computer in order to manage flight data acquisition and helicopter control. It is a complete reconfigurable embedded system which consists of a real-time processor and a reconfigurable chassis containing user-programmable FPGA modules and other swappable industrial I/O modules. The CRIO real time core is an industrial 200 MHz Pentium processor which contains 64 MB of DRAM memory and 512 MB of nonvolatile Compact Flash memory for flight data storage. It includes also 1 serial port and a 10/100 Mb network port for connection to a wireless access point. The I/O modules contain built-in signal conditioning, isolation and I/O connectors to plug-in sensors and actuators. The sensor package, installed on the UNIBO RUAV, includes a Crossbow NAV420 GPS-aided AHRS (Attitude Heading Reference System) and ultrasonic sensors which provide accurate altitude measurement with a resolution of 2 cm. Two separated radio receivers (one inside the avionics box and one mounted on the helicopter frame) and electronic switches are also installed in order to give back helicopter control to the R/C pilot in event of computer failure.

The onboard software has been developed in Labview code and then compiled into the CRIO FPGA and real time modules. Particularly, the FPGA code:

- reads PWM commands from the radio receiver inside the avionics box
- acquires helicopter attitude, angular rates, velocities and position, provided by the Crossbow NAV 420 with 100 Hz updates, using an RS232 protocol. The RS232 protocol has been managed using the FPGA Digital Input to guarantee deterministic data acquisition
- acquires altitude measurement from the ultrasonic sensor using an I2C protocol
- manages a PID based control loops for helicopter control closed at 50 Hz. In manual mode, the original pilot commands go directly to the servos, in automatic mode, the controller generates commands based on the commanded variables. The two flight modes are chosen via radio switch. A second radio switch is used to disable the onboard computer in event of electronics failure: in this safety mode, commands are sent to the servo by means of the second radio receiver mounted on the helicopter airframe

The CompactRIO real time processor receives sensor information from the FPGA and records all the flight data; meanwhile it manages also wireless Ethernet communication with the ground control station. The ground control station software is also developed in Labview and runs on a laptop computer. The remote graphical user interface is constituted by two windows (the virtual cockpit window and the telemetry window) for real time display of flight data information (Figure 52). Additional information is available such as GPS and inertial measurement unit status and system warnings. The ground operator can initiate and terminate the flight software or interact with the program starting and stopping the onboard data logging.

10 Avionics Package Validation

The validation of the avionic system was based on a 6 step procedure that is shown in Figure 53.

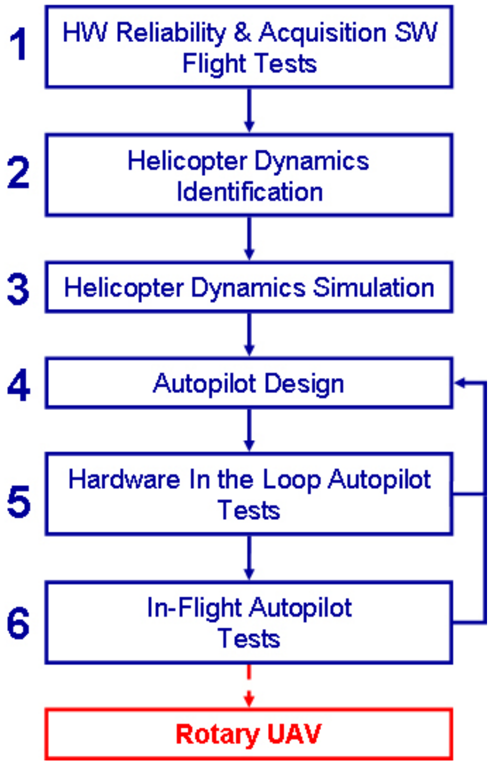

1
HW Reliability & Acquisition SW Flight Tests
2
Helicopter Dynamics Identification
3
Helicopter Dynamics Simulation
4
Autopilot Design
5
Hardware In the Loop Autopilot Tests
6
In-Flight Autopilot Tests
Rotary UAV

Figure 53: Avionics package validation process

First a series of flight tests was performed in order to validate the avionics hardware and flight data acquisition software. The collected data were then analyzed for evaluating the dynamic characteristics of the small scale helicopter: roll/pitch rate and velocities transfer functions were identified in nearly hover flight conditions. Afterwards, a classical PI nested loop controller was designed for pitch, roll, yaw and linear velocities neglecting helicopter cross-coupling dynamics. The controller was tested before in a HIL simulator and then in flight. Experimental results showed that the avionic system is able to satisfy design requirements providing reliable sensor measurements and good control capabilities. The results of the validation procedure are detailed in the next sections.

10.1 Onboard Sensor Measurements Tests

The UNIBO RUAV avionics hardware was successfully tested in flight. Flight data were transferred from the air vehicle back to the GCS via wireless data link.
All onboard electronics worked properly while sensor data was recorded at 100 Hz. AHRS raw data (Figure 54) show vibration disturbances.

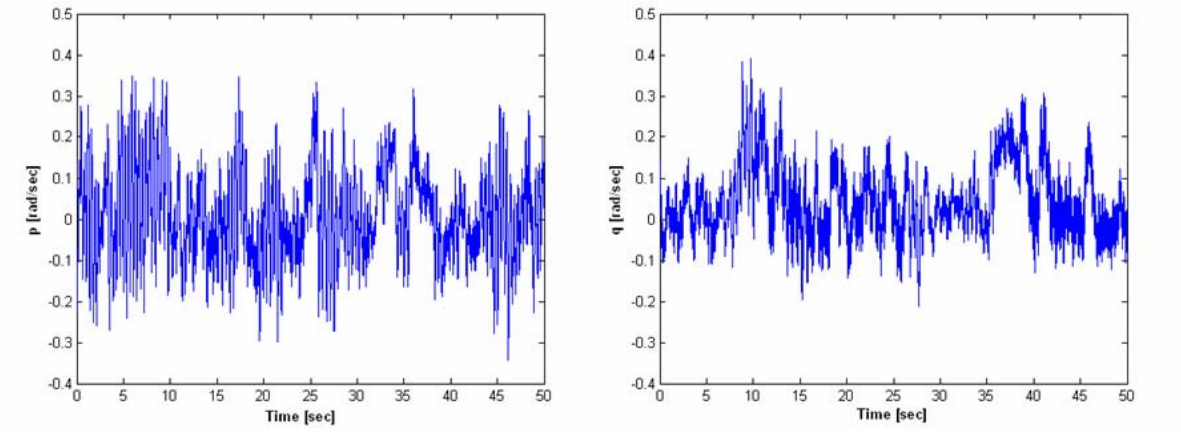


Figure 54: Example of pitch and roll rate AHRS raw data

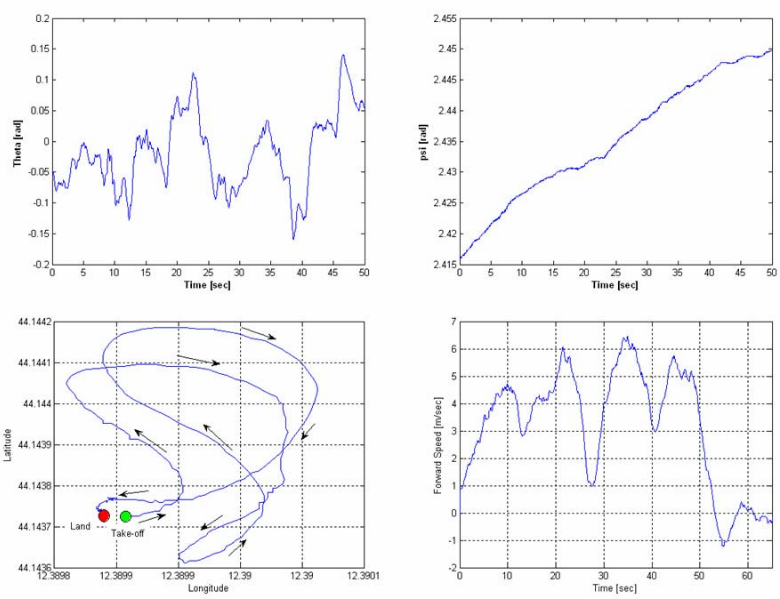


Figure 55: AHRS filtered flight data

However, thanks to the XBow NAV420 integrated Kalman filter, smooth and stable GPS position information, velocity and attitude measurements were available, which can be used for system identification control and navigation laws implementation. Figure 55 shows examples of sensor data measurements taken while the helicopter was overflying the test field at low speed conditions.
Ultrasonic sensors were also tested. First they were calibrated at ground and then mounted on the avionics box, using neoprene strips for vibration isolation. Recorded flight tests showed good experimental results although they can provide reliable altitude measurements only up to 5.5 m (Figure 56). More details are given in [27].

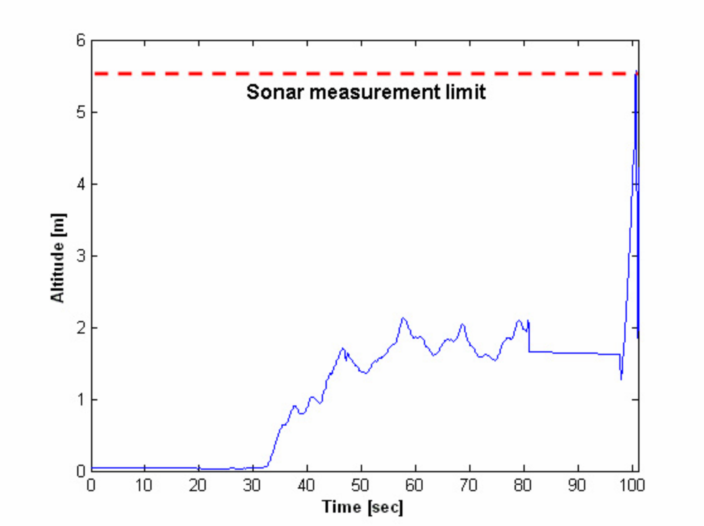


Figure 56: Sonar altimeter indication

10.2 Helicopter Dynamics Identification and Simulation

The goal of system identification is to achieve the best possible fit of the flight data with a model that is consistent with the physical knowledge of the vehicle dynamics. The first part of the problem consists on the collection of flight data. The second step is to develop a model structure with unknown parameters. Once this is accomplished, the parameters of the model can be identified. Based on the results obtained, the model structure could be refined until satisfactory agreement between flight test and system's time responses is achieved. Our aim is to identify the helicopter and to build a parametric dynamics model for near hover conditions.

10.2.1 Collection of Flight Data

A series of flight experiments have been organized for hover-flight operating point. For each flight in a series, the pilot has applied a frequency sweep control sequence (Figure 57) to one of the four control inputs via the R/C (radio control) unit. While doing so he has used the other control inputs to hold the helicopter at the selected operating point. In order to gather enough data, the same experiment has been repeated several times. The experiments have been conducted open-loop, except for an active yaw damping system, and the stabilizer bar which can be regarded as a dynamic augmentation. If as input a frequency sweep is used, the flight experiments for the hover condition are unproblematic: the helicopter is in the proximity of the pilot and it is relatively easy to hold the operating point and at the same time all the frequency of interest are excited [28].
High quality flight data is essential to a successful identification. The principal concerns are the accuracy of the state estimates (i.e., unbiased, disturbance free, no drop outs), the information content of the flight data (i.e., whether the measurements contain evidence of all the relevant flight-dynamic effects), and the compatibility of the flight data with the postulate of linear dynamics used for the modeling. While the accuracy of the state estimates depends on the instrumentation, the information content and compatibility depends on the execution of the flight experiments.

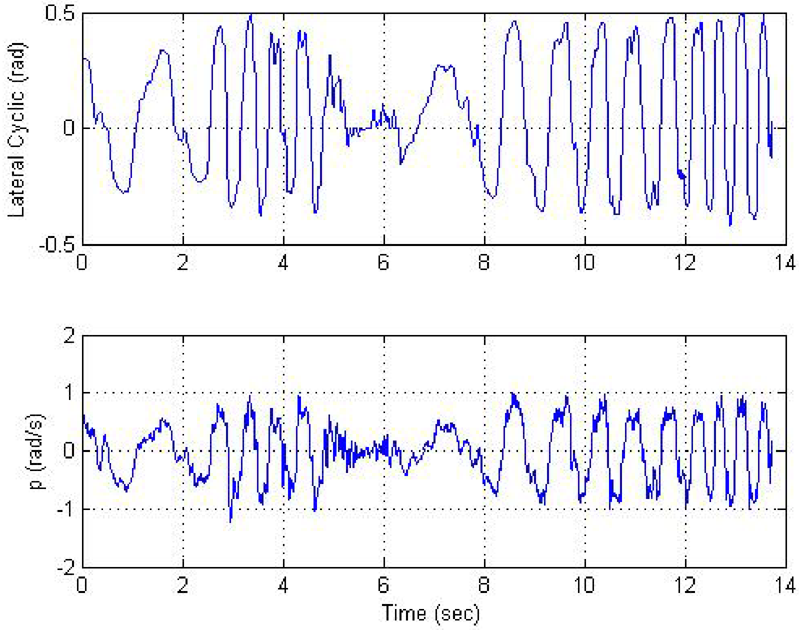


Figure 57: Sample flight data for lateral frequency sweep

10.2.2 Angular Rates and Attitudes Model Structure

In his work, Mettler [28] showed that small scale helicopters exhibit characteristics which, for the attitude rate dynamics, are very close to those of a second order system. The pitching and roll rates dynamics, for low-speed flight only, can be approximated through the following transfer functions:

$$\frac{q}{\delta_{long}} = \frac{A_{long}}{\tau_e} \frac{\omega_{nq}^2}{s^2 + 1/\tau_e s + \omega_{nq}^2}$$

$$\frac{p}{\delta_{lat}} = \frac{B_{lat}}{\tau_e} \frac{\omega_{np}^2}{s^2 + 1/\tau_e s + \omega_{np}^2}$$

In the equations, ω_{nq} and ω_{np} are the natural frequencies of the longitudinal and lateral fuselage-rotor modes τ_e is the effective rotor time constant for the flapping motion taking into account the effect of the stabilizer bar. A_{long} and B_{lat} are the effective cyclic control derivatives taking into account the effect of the stabilizer bar. All these parameter must be identified for our flight vehicle.

The model structure defined before assumes that the longitudinal and lateral modes are decoupled. These assumptions were confirmed from experimental flight data records.

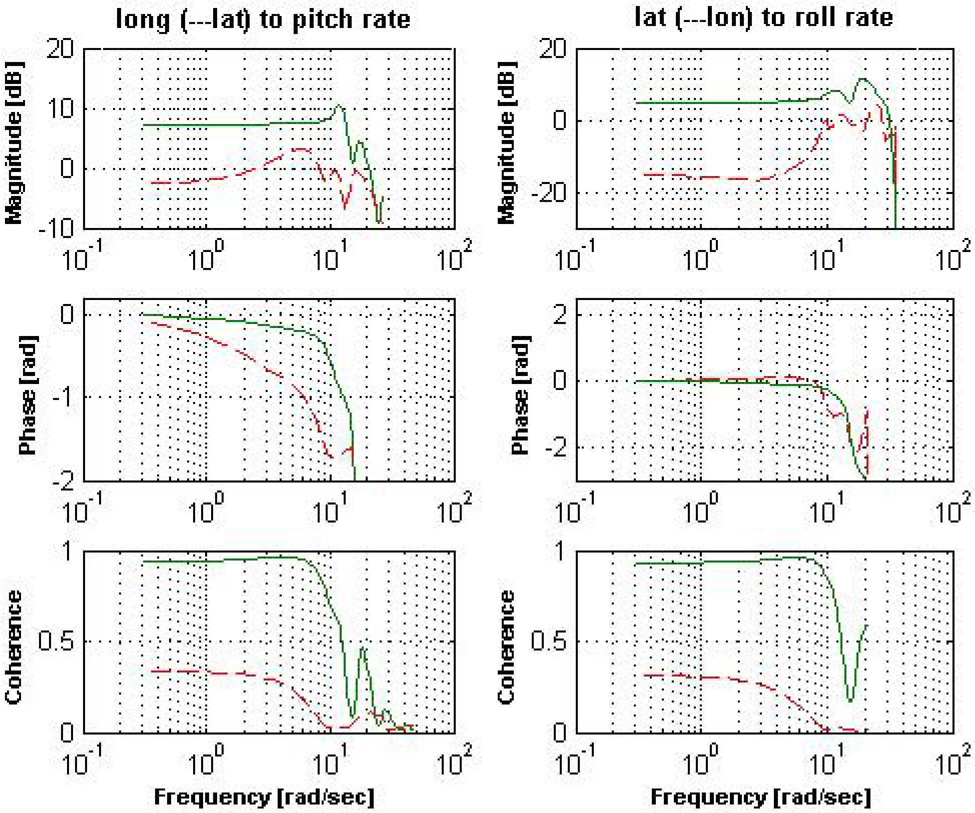


Figure 58: Pitch and roll rate estimated frequency responses to longitudinal and lateral cyclics

Figure 58 shows the estimated experimental frequency response for the on and off-axes roll and pitch angular rates vs. frequency sweep longitudinal and lateral cyclic inputs. Both on-axes pitch and roll rate responses *q/δ_{long}* and *p/δ_{lat}* exhibit well defined characteristics of a second-order system. Furthermore the coherence

$$\gamma_{xy} = \frac{\left|G_{xy}\right|^2}{G_{xx}G_{yy}} \leq 1$$

where G_{xy}, G_{xx} and G_{yy} are respectively the cross-spectrum and the auto-spectrum, demonstrate how the off-axis command is low correlated with the considered output and hence the hypothesis of decoupled system could be assumed.

The transfer function parameters were identified using the Simulink™ Optimization Toolbox [21]. The parameter initial values, to be used in the optimization algorithm, were estimated from the vehicle mass-geometry characteristics reported in Table 1.

Particularly, referring to Table 1, an initial value of ω_{nq} and ω_{np} is given by:

$$\omega_{nq} = \sqrt{\frac{T_{mr}h_{mr} + k_{\beta}}{I_{yy}}}$$

$$\omega_{np} = \sqrt{\frac{T_{mr}h_{mr} + k_{\beta}}{I_{xx}}}$$

For nearly hover conditions $T_{mr} \cong mg$. The distance between the main rotor hub and the helicopter center of gravity h_{mr} was measured. The moments of inertia I_{xx} and I_{yy} were determined with the torsional pendulum test. The hub torsional stiffness k_{β} was evaluated from static measurements. The values, estimated for the natural frequencies ω_{nq} and ω_{np}, were also confirmed from flight tests: in Figure 58, the frequency response magnitude peaks at around 12 rad/sec for the pitch rate and 18 rad/sec for the roll rate. An approximated value of τ_e is given by τ_e= 16/($\gamma_{fb}\Omega_{mr}$) [7], where γ_{fb} is the Lock number of the stabilizer bar and Ω_{mr} is the main rotor speed. Therefore a starting value for τ_e was estimated from geometry and vehicle physical characteristics (see Table 5), a reasonable initial value for A_{long} and B_{lat} was estimated from [7].

Parameter	Description	Source
$m = 11.2$ kg	Helicopter mass	Measured
$I_{xx} = 0.30\ \mathrm{kg \cdot m^2}$	Rolling moment of inertia	Torsional Pendulum Test
$I_{yy} = 0.79\ \mathrm{kg \cdot m^2}$	Pitching moment of inertia	Torsional Pendulum Test
$I_{zz} = 0.57\ \mathrm{kg \cdot m^2}$	Yawing moment of inertia	Torsional Pendulum Test
$K_{\beta} = 80$ N·m/rad	Hub torsional stiffness	Static Measurement
$\gamma_{fb} = 0.8$	Stabilizer bar Lock number	Calculated
$c_{fb\ ext} = 0.35$ m	Stabilizer bar external radius	Measured
$c_{fb\ int} = 0.235$ m	Stabilizer bar internal radius	Measured
$c_{fb} = 0.06$ m	Stabilizer bar chord	Measured
$a_{fb} = 2.67\ \mathrm{rad^{-1}}$	Stabilizer bar lift curve slope	Estimated [7]
$I_{\beta fb} = 0.003\ \mathrm{kg \cdot m^2}$	Stabilizer bar flapping inertia	Estimated [7]
$\Omega_{nom} = 138$ rad/s	Nominal main rotor speed	Measured
$R_{mr} = 0.92$ m	Main rotor radius	Measured
$c_{mr} = 0.07$ m	Main rotor chord	Measured
$a_{mr} = 5.3\ \mathrm{rad^{-1}}$	Main rotor blade lift curve slope	Estimated [7]
$I_{\beta mr} = 0.071\ \mathrm{kg \cdot m^2}$	Main rotor blade flapping inertia	Torsional Pendulum Test

Table 5: UNIBO RUAV parameters I

The identified parameter values are reported in Table 6.

Identified Parameters				
$\boldsymbol{A_{long}}$ [rad/rad]	ω_q [rad/sec]	$\boldsymbol{B_{lat}}$ [rad/rad]	ω_q [rad/sec]	τ_e [sec]
0.30	12.1	0.22	18.1	0.132

Table 6: Identified transfer functions parameters

The pitch and roll rate response q/δ_{long} and p/δ_{lat} were simply integrated to obtain two third order transfer functions for the roll and pitch angle. Figure 59 shows good time domain response agreement between experimental data and the model-predicted responses to pilot input. Note that time domain comparison was made using control inputs and experimental data different from the one used in the identification process.

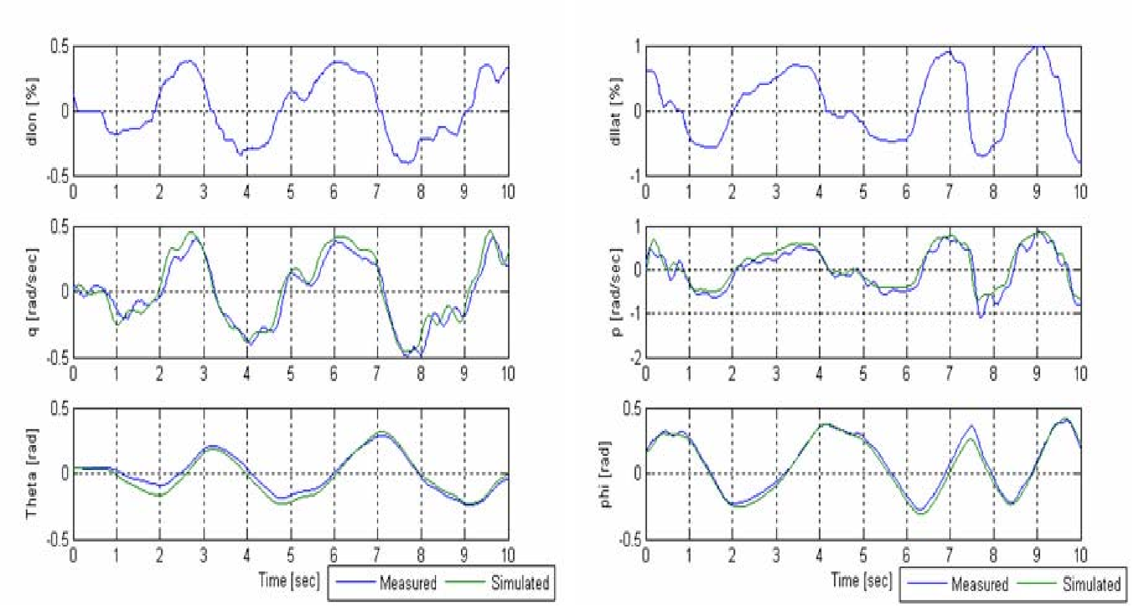


Figure 59: Simulated vs. experimental lateral and longitudinal responses

The Matlab® *BEST FIT* parameter was used as index to evaluate the agreement between simulation and experimental results. By definition, it is given by the following equation:

$$BEST\ \ FIT = \left[\frac{1 - norm(y_{meas} - y_{sim})}{norm(y_{meas} - mean(y_{meas}))}\right]$$

The computed values were 80% for the pitch angle and 76 % for the roll angle (usually a *BEST FIT* index greater than 70% is considered a good agreement level).

For the yaw rate a different approach was taken. Small helicopters come equipped with a so called Heading Lock electronic system, which is effectively a yaw rate controller. Helicopters of this size are almost impossible to fly without these systems. For safety reasons, we have decided to leave the existing yaw rate controller in place. This aids in a smooth transition for the backup pilot, if the controller fail during testing. Hence the bare helicopter yaw dynamics identification is not essential for our purposes and it is not yet identified.

10.2.3 Forward and Lateral Speed Model Structure

In near hover condition, a very simplified helicopter forward and lateral ground speed model is given by [7,8]:

$$\dot{V}_x = -g\theta + X_{Vx}V_x$$

$$\dot{V}_y = g\varphi + X_{Vy}V_y$$

$$X_{Vx} = -\frac{1}{2m}S_x^f \rho V_{imr}$$

$$X_{Vy} = -\frac{1}{2m}S_y^f \rho V_{imr}$$

where the parameters are described in Table 7.

Parameter	Description	Source
m = 11.2 kg	Helicopter mass	Measured
S_x^f = 0.3 m^2	Frontal fuselage drag area	Estimated [7]
S_y^f = 0.2 m^2	Side fuselage drag area	Estimated [7]
ρ = 1.225 kg · m^3	Air density	---------------
V_{imr} = 4.2 m/s	Yawing moment of inertia	Calculated
g = 9.81 m/s^2	Gravity acceleration	---------------

Table 7: UNIBO RUAV parameters II

From the previous equations we can obtain the following transfer functions:

$$\frac{V_x}{\theta} = \frac{-g}{s - X_{Vx}}$$

$$\frac{V_y}{\varphi} = \frac{g}{s - X_{Vx}}$$

with X_{Vx} and X_{Vy} initial guess respectively -0.07 1/s and 0.047 1/s

The transfer function parameters were identified using the Simulink™ Optimization Toolbox. We have found:

- X_{Vx} = -0.39 1/s
- X_{Vy} = -0.05 1/s

Figures 60 and 61 show good time domain response agreement between experimental data and the model-predicted responses to pilot input. Note that time domain comparison was made using control inputs and experimental data different from the one used in the identification process.

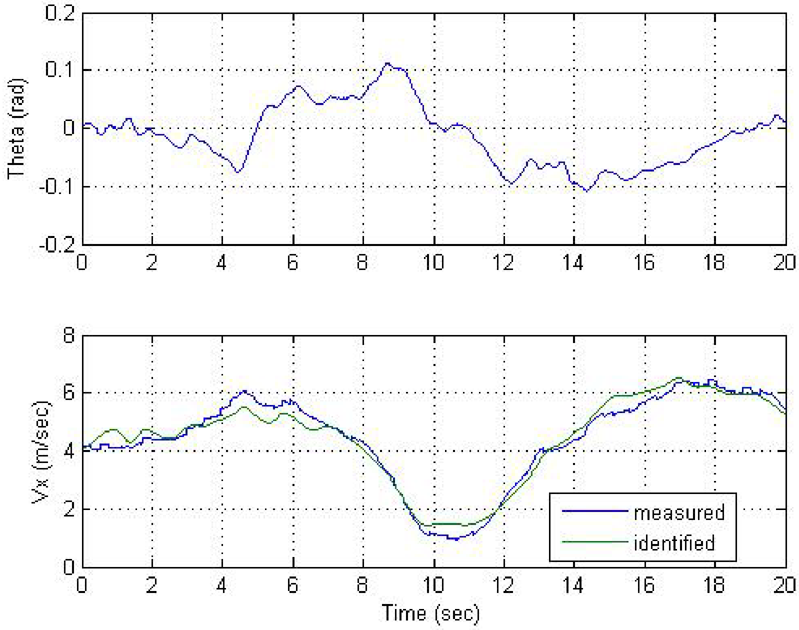


Figure 60: Simulated vs. experimental forward velocity response

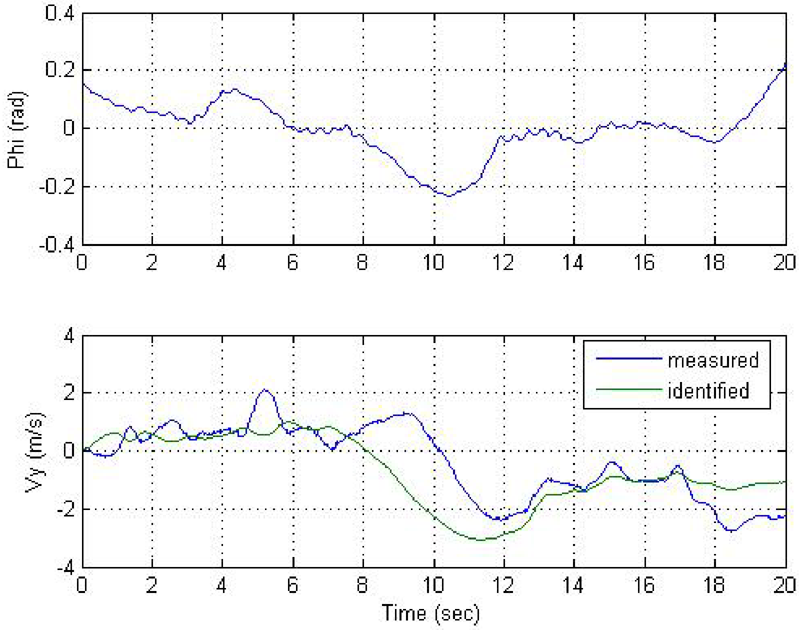


Figure 61: Simulated vs. experimental lateral velocity response

10.2.4 Vertical Down Velocity Model Structure

A very simplified model that describes the heave helicopter dynamic can be written as [7]:

$$\frac{V_D}{\delta_{col}} = \frac{Z_{col}}{s - Z_{Vd}}$$

where Z_{col} and Z_{Vd} are respectively the vertical speed damping derivative and the collective pitch control derivative. An estimate of the two derivatives can be obtained analytically by linearization of the momentum theory equations [8]:

$$Z_{Vd} = -\frac{2a\sigma\lambda_0\rho\pi\Omega R^3}{m(16\lambda_0 + a\sigma)}$$

$$Z_{col} = -\frac{4a\sigma\lambda_0\rho\pi\Omega^2 R^4}{3m(16\lambda_0 + a\sigma)}$$

A starting value for Z_{col} and Z_{Vd} was estimated from geometry and vehicle physical characteristics reported in Table 8.

Parameter	Description	Source
m = 11.2 kg	Helicopter mass	Measured
ρ = 1.225 kg · m^3	Air density	---------------
a =5 1/rad	M.R. blade lift curve slope	Estimated [7]
σ =0.035	Main rotor solidity	Calculated
λ_0=0.033	Main rotor inflow ratio	Calculated
Ω = 138 rad/s	Nominal main rotor speed	Measured

Table 8: UNIBO RUAV parameters III

The resulting optimization starting values are:

- Z_{col} = -42 m/(rad s^2)
- Z_{Vd} = -0.33 1/s

The transfer function parameters were identified using the Simulink™ Optimization Toolbox. We have found:

- Z_{col} = -30 m/(rad s^2)
- Z_{Vd} = -1.1 1/s

Figure 62 shows good time domain response agreement between experimental data and the model-predicted responses to pilot input. Note that time domain comparison was made using control inputs and experimental data different from the one used in the identification process.

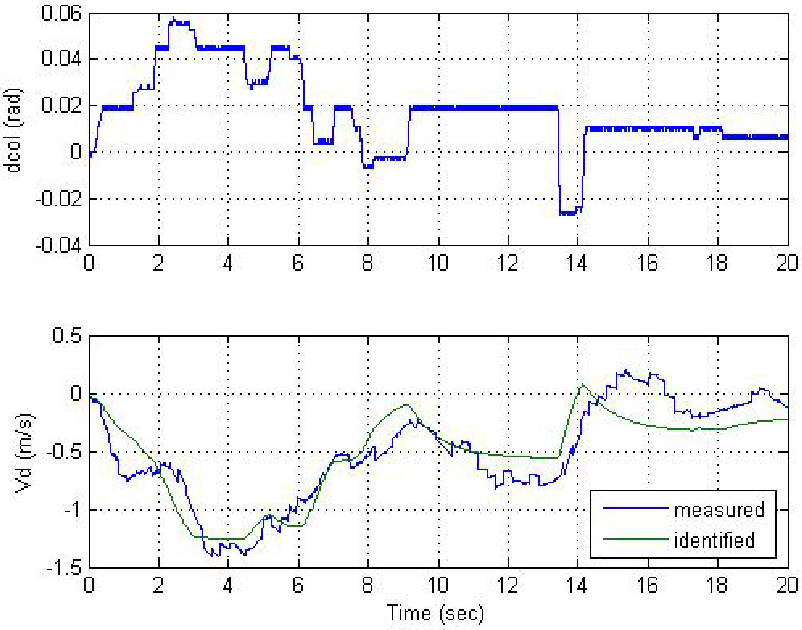


Figure 62: Simulated vs. experimental vertical down velocity response

The identified attitude and velocity transfer functions are then used to design the helicopter controller. The development of the autopilot is described in the next section.

10.3 Autopilot Design

The autopilot is based on a classical PID 3 level nested loop structure [25,29]. A schematic of the controller is reported in Figure 63

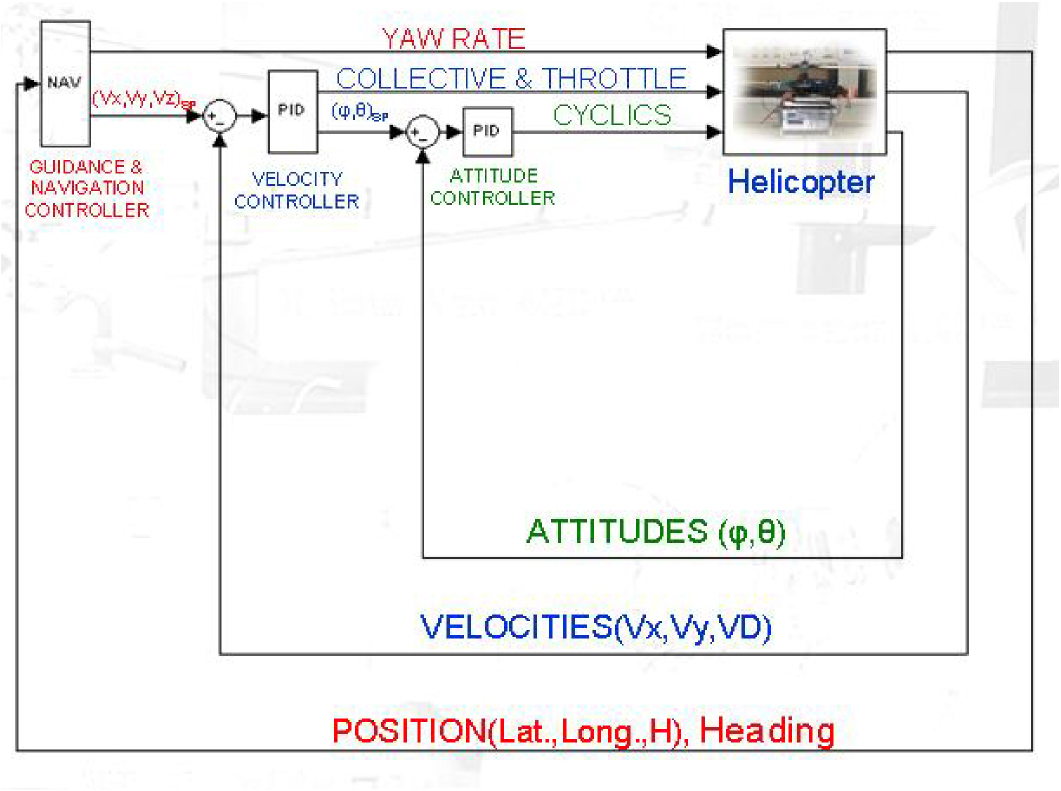


Figure 63: UNIBO RUAV autopilot schematic

The outer-loop guidance and navigation loop is not yet fully implemented. By now, only the yaw rate is controlled by the onboard computer. However, the logic that will be implemented is the one already described and tested in Section 4.3.2.

The ground velocity control is implemented using the two levels, nested-loop structure shown in Figure 63. Lateral velocity (Vy) errors are used to generate roll demands for the roll (φ) control module, while longitudinal velocity (Vx) errors are used to generate pitch demands for the pitch (θ) control module while vertical down velocity errors are used to generate directly collective and throttle commands. Finally pitch and roll errors are used in the attitude controller to generate cyclics.

10.3.1 Attitude Controller

Roll (ϕ) and pitch (θ) control is implemented using simple PI modules; their generic structure is shown in Figure 64.

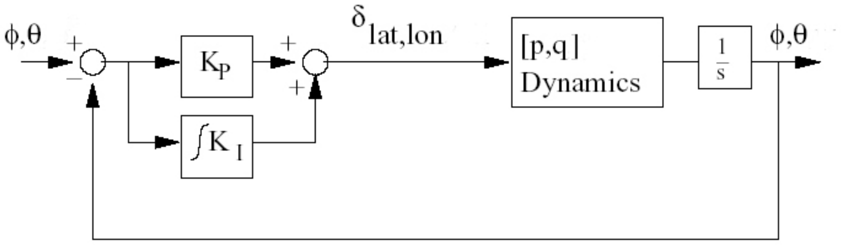


Figure 64: Attitude control modules

The PI gains have been calculated using the well known Ziegler-Nichols rules [25].
We have found:

- $K_{p\theta}$ = -0.77 rad/rad
- $K_{i\theta}$ = -0.08 rad/(rad s)
- $K_{p\varphi}$ = -1.04 rad/rad
- $K_{i\varphi}$ = -0.11 rad/(rad s)

The integral compensates for two types of variance. The first and most simple source is that associated with aircraft maintenance resulting in control linkage length changes, and hence servo positions for trimmed flight; this variation occurs between flights. The second source of variation is wind and all the other disturbances and coupling effects that were not taken into account by the identified helicopter math model.
The pitch and roll modules tracking performance are shown in Figures 65 and 66

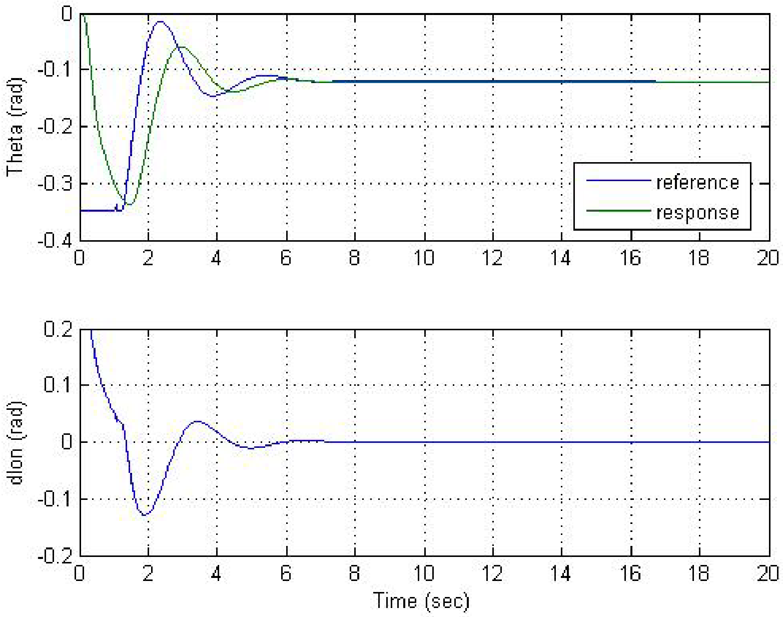


Figure 65: Pitch controller tracking performance

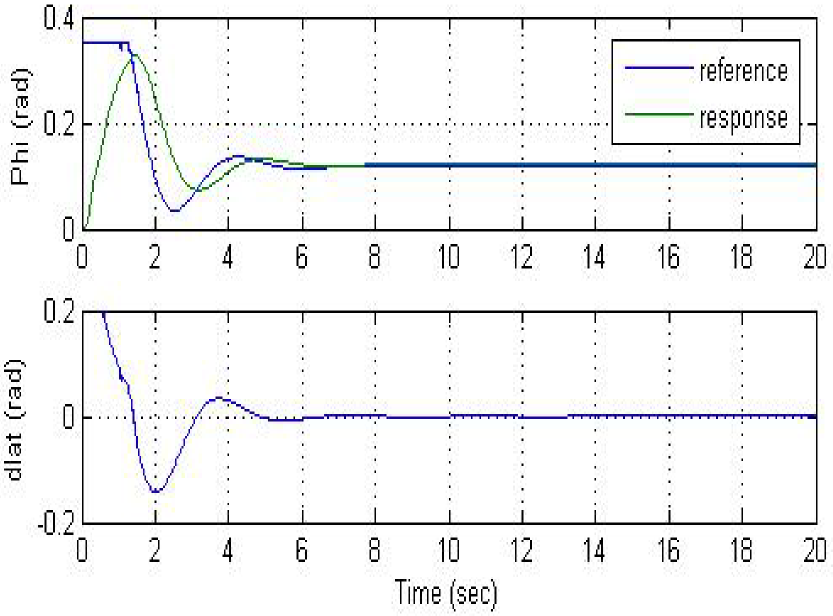


Figure 66: Roll controller tracking performance

10.3.2 Velocity Controller

Forward ground velocity (Vx) and lateral ground velocity (Vy) control is implemented using simple PI modules; their generic structure is shown in Figure 67.

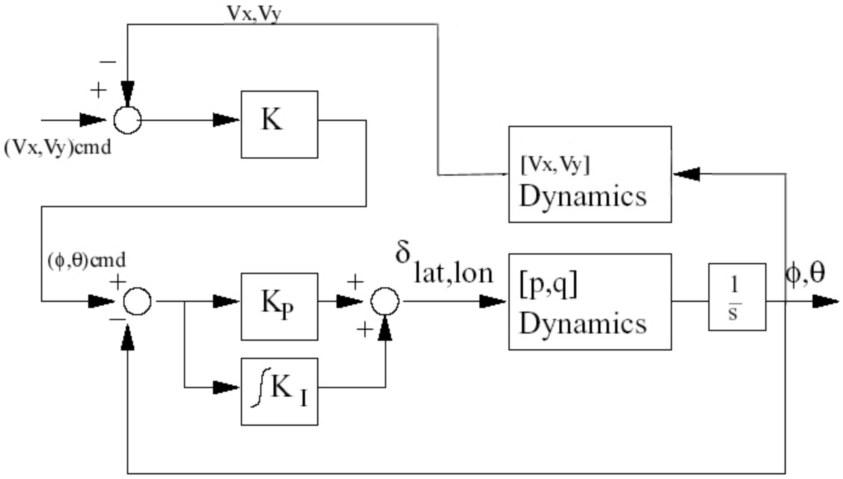


Figure 67: Velocity control modules

The PI gains have been calculated using the well known Ziegler-Nichols rules.
We have found:

- K_{pVx} = -0.23 rad/(m/s)
- K_{iVx} = -0.07 rad/m
- K_{pVy} = -0.20 rad/(m/s)

- K_{iVy} = -0.062 rad/m

The Vx and Vy modules tracking performance are shown in Figures 68 and 69

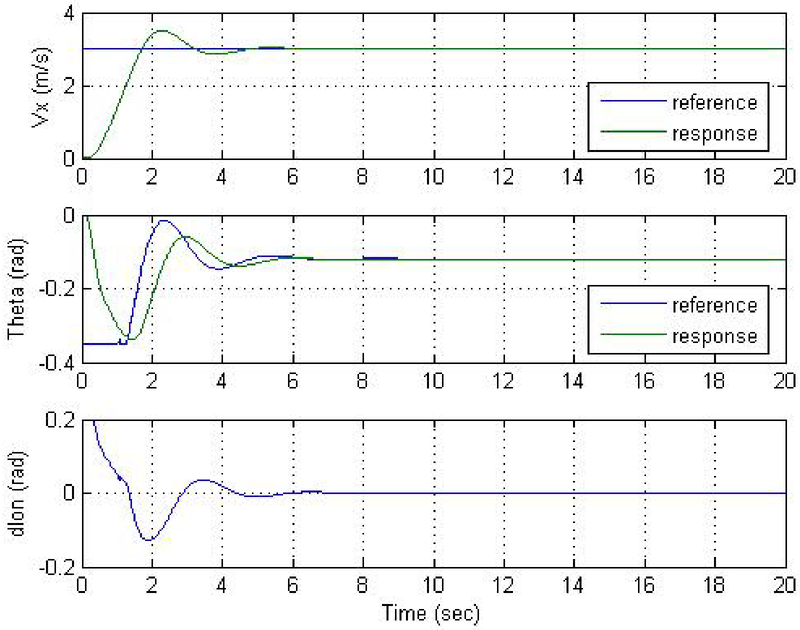


Figure 68: Forward velocity controller tracking performance

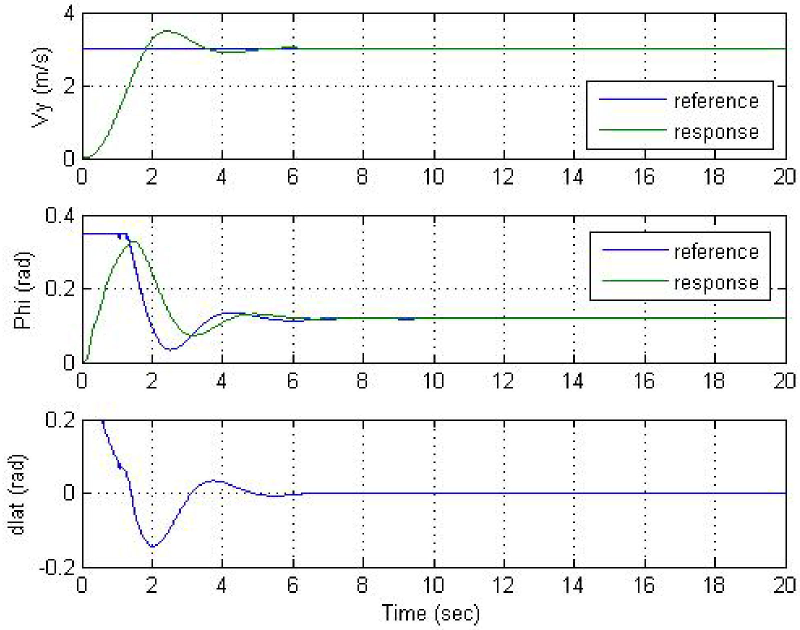


Figure 69: Lateral velocity controller tracking performance

10.3.3 Vertical Down Velocity Controller

Vertical down velocity (Vd) control is implemented using a simple PI module; its generic structure is shown in Figure 70.

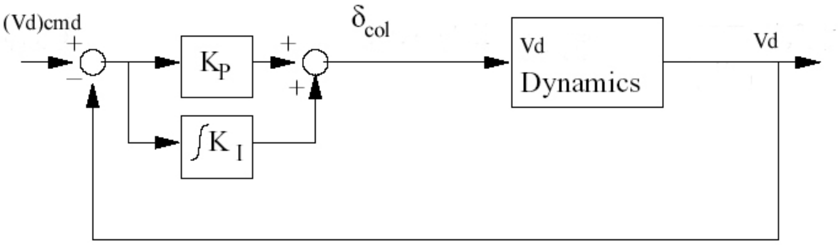


Figure 70: Vertical down velocity control module

The PI gains have been calculated using the well known Ziegler-Nichols rules and fine tunings.
We have found:

- K_{pVd} = -0.063 rad/(m/s)
- K_{iVd} = -0.09 rad/m

The Vd module tracking performance are shown in Figure 71

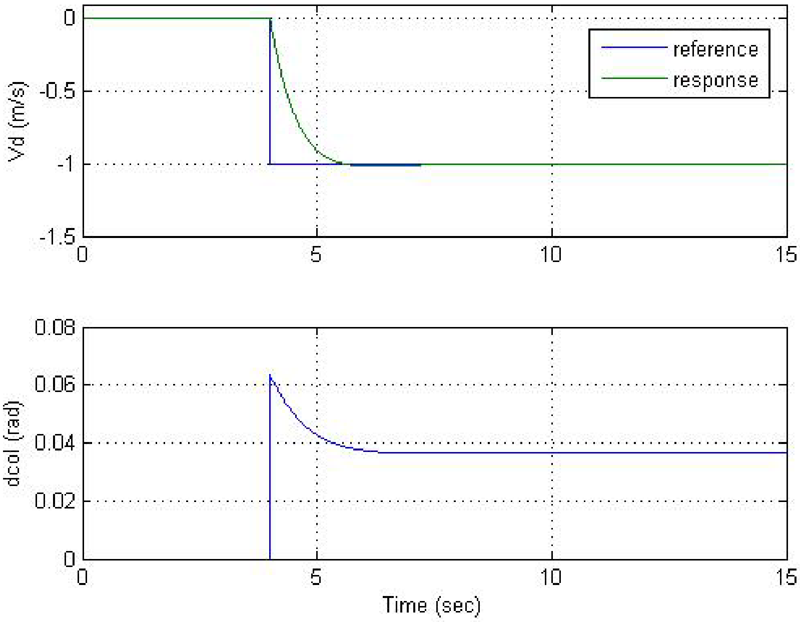


Figure 71: Vertical down velocity controller tracking performance

10.3.4 Heading Controller

Heading control is achieved using the onboard gyro AVCS (Angular Velocity Control System). Therefore, the implemented algorithm [27] gives a reference yaw rate to the gyro AVCS, based on the heading error, calculated with respect to the reference heading set point.

The heading tracking performance during a flight test are shown in Figure 72

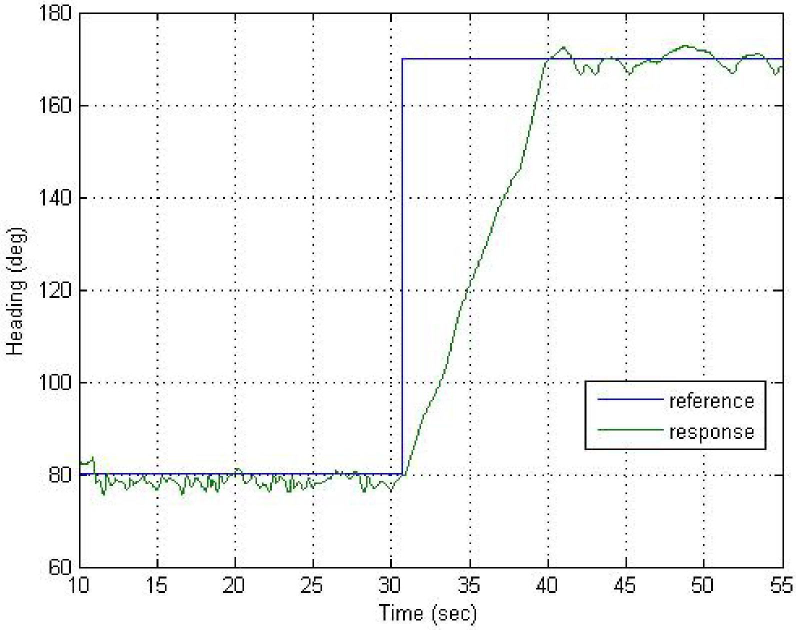


Figure 72: Heading controller tracking performance

10.4 Hardware In the Loop Autopilot Tests

The autopilot software was translated and implemented on the onboard computer [27]. To allow safe, risk-free testing, the PID controllers were first tested in a HIL (Hardware In the Loop) simulator [27] which is shown in Figure 73.

The HIL simulator is constituted by:

- an exact duplicate of the flight computer (the CRIO System) and of the onboard software including the autopilot. Reference value to the controller are given by means of the R/C transmitter and then acquired by the CRIO software from the R/C receiver
- a computer which simulates the helicopter dynamics through the identified transfer functions
- an OpenGL visual system for rendering the helicopter as it moves around in a virtual scenery.

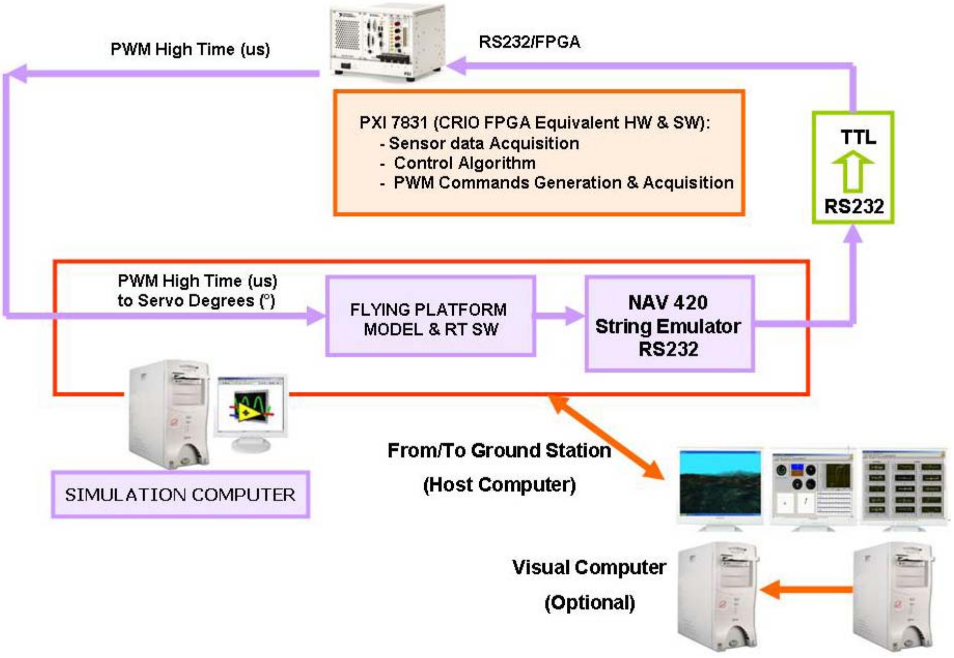

PWM High Time (us)
RS232/FPGA
PXI 7831 (CRIO FPGA Equivalent HW & SW):
- Sensor data Acquisition
- Control Algorithm
- PWM Commands Generation & Acquisition
TTL
RS232
PWM High Time (us) to Servo Degrees (°)
FLYING PLATFORM MODEL & RT SW
NAV 420 String Emulator RS232
SIMULATION COMPUTER
From/To Ground Station (Host Computer)
Visual Computer (Optional)

Figure 73: UNIBO HIL simulator

10.5 In-Flight Autopilot Tests

After HIL tests, the onboard control software was tested in flight. The complete flight campaign was done following five major subsequent steps:

- First only the attitude (φ and θ) PI controllers were tested. During these tests collective and tail commands were left to the R/C pilot for safety reasons. As shown in Table 9, the final proportional PI gains find by simulation results were almost correct while the integral gains were increased of an order of magnitude. This may be due to the fact that attitude controller are of course very sensible to external unknown disturbances which cannot hardly be simulated.
- Once the attitude controllers were somehow calibrated, the nested PI Velocity – Attitude controllers were tested (see Figure 74). During these tests, collective and tail commands were still left to the R/C pilot for safety reasons. As shown in Table 10, the final gains were much closer to the one found by simulations.
- The third step was to test the heading control together with the nested PI velocity controller. During these flight tests only collective was left to the R/C pilot for safety reasons. The value to be calibrated during these flights was the yaw rate to be sent to the gyro AVCS system. For this kind of helicopter we have found adequate a yaw rate of 10 deg/s. This value was kept intentionally low for safety reason but can be increased if necessary.
- In the fourth step the full PI controller was tested including the vertical velocity control. During these tests no commands was left to the pilot and the helicopter was flying completely autonomously. As shown in Tables 9 and 10, the final calibrated PI gains were higher with respect to the one calculated by simulations. This was due to the fact that, during simulations, the gains were kept intentionally low for the helicopter to have a very slow response. Vertical velocity flight tests can be very dangerous since small helicopters are very responsive to collective inputs and hence the helicopter can crash to the ground without any hope to recover it. Therefore, the helicopter team decided to keep the gains small at the beginning and increase them once it was sure that the helicopter was flying safely. The first test performed with the simulated gains showed that the helicopter was able to maintain hover conditions. However, the rate of climb/descent was quite very low and the PI gains were, therefore, increased.
- Finally, after each controller was fine tuned, the full control system was tested over a squared flight pattern. The distance tracked by the helicopter was kept within the R/C transmitter range and pilot good line of sight in order to recover the helicopter if needed. As shown in Figures 75 and 76, the helicopter was able to perform autonomously and successfully the preprogrammed pattern.

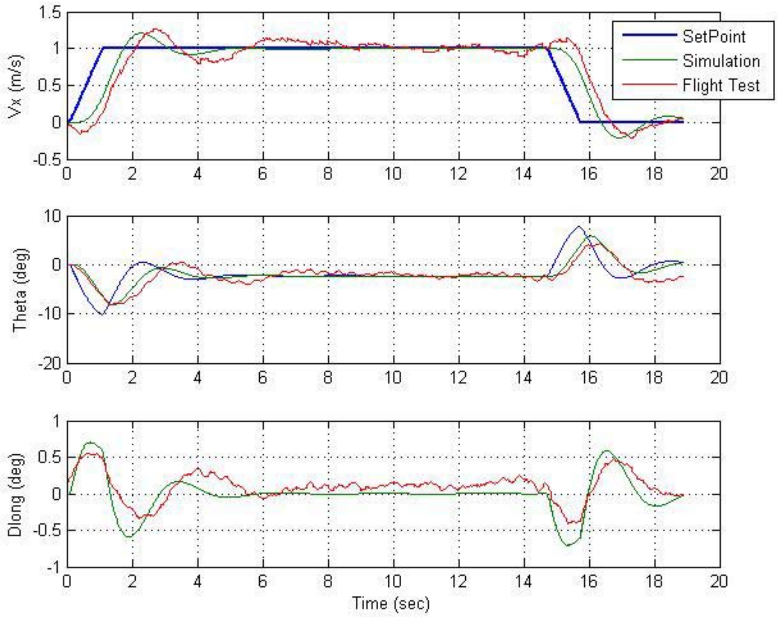


Figure 74: Simulate vs. Experimental longitudinal controller tracking performance

As an example Figure 74 shows good Vx and θ tracking performance and good agreement between simulation and real flight tests.

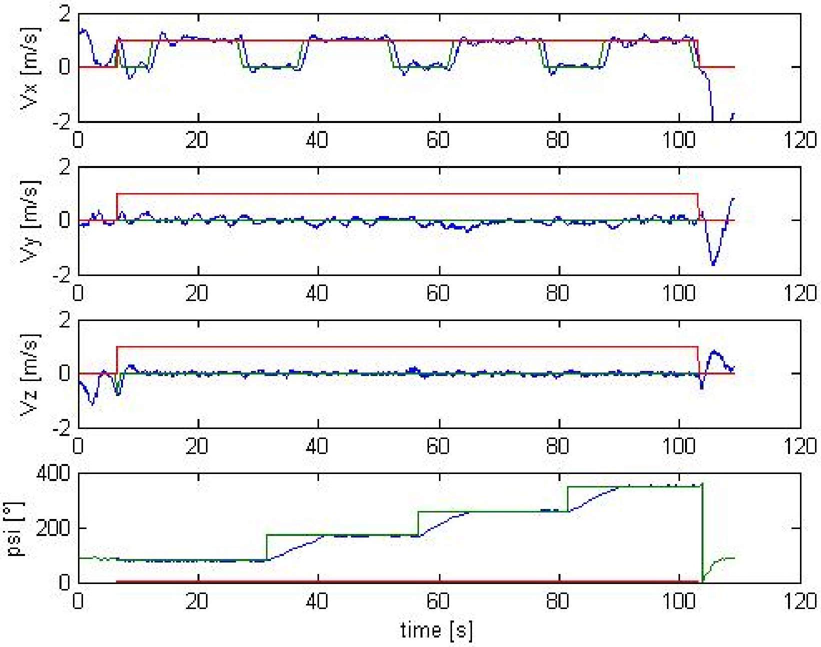


Figure 75: Recorded data during autonomous square pattern

In Figure 75:

- in Red→ autopilot ON(1) or OFF(0)
- in Blue→ flight data
- in Green→autopilot commanded values

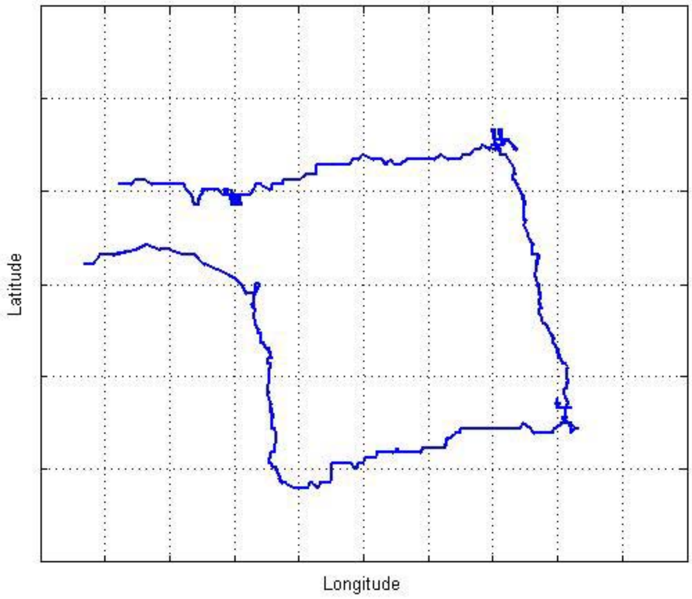


Figure 76: Autonomous square pattern RUAV ground track

Attitude PI Gains				
	$K_{p\theta}$ [deg/deg]	$K_{I\theta}$ [deg/(deg s)]	$K_{p\varphi}$ [deg/deg]	$K_{I\varphi}$ [deg/(deg s)]
Calculated	-0.77	-0.08	-1.04	-0.11
Experimental	-1	-1	-1	-1

Table 9: Calculated vs. Experimental attitude PI gains

Velocity PI Gains						
	K_{pVx} [deg s/m]	K_{IVx} [deg/m]	K_{pVy} [deg s/m]	K_{Ivy} [deg/m]	K_{pVz} [deg s/m]	K_{IVz} [deg/m]
Calculated	-13.2	-4.0	-11.5	-3.6	-3.6	-4.96
Experimental	-10	-1	-10	-1	-10	-10

Table 10: Calculated vs. Experimental velocity PI gains

11 UNIBO RUAV Project: Conclusion and Outlook

An avionics package was set up for the UNIBO RUAV using commercial and cost effective technology. HIL simulations and experimental flights were performed in order to test the feasibility to use the avionics hardware and software for helicopter identification model and control system development. A simple PID nested loop autopilot was designed based on identified transfer functions for the helicopter attitudes and velocities. Results demonstrated that the avionics system is able to provide accurate flight data measurements for dynamic model identification and for helicopter control capabilities.

As a result the RUAV flown autonomously and successfully along a preprogrammed square pattern. Through the rapid prototyping approach described in this work, UNIBO is the first Italian University that achieves complete autonomous helicopter flight.

In the near future, the project will continue with the development of take-off and landing autopilot modules and with the improvement of the actual controller.

The developed RUAV platform will then be used inside the University as flying test bed for researches in control and navigation laws, man-machine interfaces and system integration.

The feasibility to install the designed avionics package on an ultra-light helicopter, integrated with additional redundant systems, will be also investigated.

Acronyms

AV	Air Vehicle
AVCS	Angular Velocity Control System
CAPECON	Civil uav APplications & Economic effectivity of potential CONfiguration solutions
COTS	Commercial Off-The-Shelf
CRIO	CompactRIO
DD	Data Distribution
DL	Data Link
DRAM	Dynamic Random Access Memory
EM	Electro-Magnetic
EO	Electro-Optics
EU	European Union
FCS	Flight Control System
FPGA	Field Programmable Gate Array
GPS	Global Positioning System
GCS	Ground Control Station
GS	Ground Segment
GV	Ground Vehicle
HIL	Hardware In the Loop
ILOS	In Line Of Sight
I/O	Input/Output
ISA	International Standard Atmosphere
LQR	Linear Quadratic Regulator
LMS	Logistics & Maintenance Segment
MGCS	Mobile Ground Control Station
MGS	Mobile Ground Segment
MMP	Mission Modular Payload
NED	North East Down coordinate frame
NGCS	Navigation Guidance & Control System
NI	National Instruments
OLOS	Out of Line Of Sight
PID	Proportional Integral Derivative
PI	Proportional Integral
PWM	Pulse Width Modulation
R/C	Radio Controlled
RF	Radio Frequency
RUAV	Rotary Wing Unmanned Aerial Vehicle
SAS	Stability Augmentation System
TFT	Thin Film Transistor
TPP	Tip-Path Plane
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UNIBO	University of Bologna
USB	Universal Serial Bus
Wp	Waypoint

References

[1] U.S. Air Force Scientific Advisory Board, “Unmanned Aerial Vehicles in perspective: Effects, Capabilities and Technologies”, SAB-TR-03-01, July 2003.

[2] CAPECON Consortium, "Annex 1 Description of Work", CAPECON Project No. GRD1-2001-40162, Starting Date May 2002.

[3] S. Tsach, D. Penn and A. Levy Israel Aircraft Industries (IAI), “Advanced Technologies and Approaches for Next Generation UAVs”, ICAS 2002.

[4] R. Pretolani, G. Saggiani, B. Teodorani, “Mini/Small Rotary Wing UAV Technologies”, CAPECON Technical Report, Report ID 10.3, November 2005.

[5] R. Pretolani, G. Saggiani, B. Teodorani, “Development of a mission simulation environment for Rotary Wing UAV”, CAPECON Technical Report, Report ID 6.4, September 2004.

[6] R. Pretolani, G. Saggiani, B. Teodorani, ” RUAV Ground Support: Basic Requirements & First Layout Internal Report”, CAPECON Report UNIBO ID 6.1/1, 31 July 2003.

[7] V. Gavrilets, B. Metller, E. Feron, “Dynamic Model for a Miniature Aerobatic Helicopter”, M.I.T.

[8] G.D. Padfield, “Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling”, AIAA Education Series, Reston, VA, 1996.

[9] B. Mettler, M.B. Tischler, and T. Kanade, “System identification modeling of a small-scale unmanned rotorcraft for control design”, Journal of the American Helicopter Society, 47(1):50–63, January 2002.

[10] B. Mettler, “Identification, Modeling and Characteristics of Miniature Rotorcraft”, Kluver Academic Publishers,Boston, MA, 2002.

[11] M. LaCivita, W. Messner, and T. Kanade, “Modeling of small-scale helicopters with integrated first-principlesand integrated system identification techniques”, Monteal, Canada, June 2002. Presented at 58th Forum of American Helicopter Society.

[12] W. von Grunhagen, G. Bouwer, H.-J. Pausder, F. Henchel, and J. Kaletka, “A high bandwidth control system for the helicopter in-flight simulator atthes - modelling, performance and applications”, M. Tischler, editor, Advances in Aircraft Flight Control. Taylor and Francis, Cornwall, England, 1996.

[13] T. D. Talbot, B. E. Tingling, W. A. Decker, and R.T. Chen, “A mathematical model of a single main rotor helicopter for piloted simulation”, Technical Memorandum 84281, NASA, 1982.

[14] B. Mettler, M. Tischler, T. Kanade, and W. Messner, “Attitude control optimization for a small-scale unmanned helicopter”, Denver, CO, August 2000. AIAA Guidance, Navigation and Control Conference.

[15] B. Mettler, V. Gavrilets, E. Feron, and T. Kanade, “Dynamic compensation for high-bandwidth control of small-scale helicopter”, San Francisco, CA, January 2002. American Helicopter Society Specialist Meeting.

[16] A.R.S. Bramwell, “Bramwell’s Helicopter Dynamics”, AIAA, Reston VA, 2001.

[17] M. McConley, “Draper small autonomous aerial vehicle dynamic model”, Technical Report E41-98-091, Draper Laboratory, August 1998.

[18] J. G. Leishman, “Principles of helicopter aerodynamics”, Cambridge University Press, New York, 2000.

[19] Preatoni Marzio, Agusta, “Rotary Wing UAV Configurations –Final Technical Report”, CAPECON Technical Report, Report ID 6.4, May 2004.

[20] Ulrik B. Hald, Mikkel V. Hesselbaek, Jacob T. Holmgaard, Christian S. Jensen, Stefan L. Jakobsen, Martin Siegumfeldt, “Autonomous Helicopter Modelling and Control”, Aalborg University, May 2005.

[21] http://www.mathworks.com/products/matlab

[22] M. Ribaldi, “Progetto di sistemi di controllo”, Pitagora Editrice, Bologna 1995.

[23] V. Gavrilets, I. Martinos, B. Mettler and E. Feron, “Control Logic for Automated Aerobatic Flight of a Miniature Helicopter”, Massachusetts Institute of Technology, Cambridge, MA 02139.

[24] J.F. Boer, F. Fresta, H. Haverdings, M.L. Preatoni, R.Pretolani, G.M. Saggiani, B.Teodorani, AGUSTA, NLR, DIEM-University of Bologna, “ Small/Mini Rotary Wing UAV Configuration”, CAPECON Report ID 10.5, November 2005.

[25] M.E.Penati, G.Bertoni, “Automazione e Sistemi di Controllo”, Vol. II, Società Editrice Esculapio, Bologna, 1997.

[26] M. Niculescu, “Lateral Track Control Law for Aerosonde UAV”, AIAA 2001-0016, University of Washington, Seattle, January 2001

[27] B. Teodorani, “Progetto e realizzazione del sistema di gestione autonoma del volo e controllo in remoto per un velivolo UAV ad ala rotante”, II School of Engineering, University of Bologna, PhD Thesis, 2007

[28] B. Mettler, “Identification modeling and characteristics of miniature rotorcraft “, Kluwer Academic Publisher, Boston, M.A., USA, 2003.

[29] Gregg Buskey, Jonathan Roberts, Peter Corke, Gordon Wyeth, "Helicopter Automation Using a Low-Cost Sensing System", School of Information Technology and Electrical Engineering, University of Queensland, ST LUCIA, Queensland, Australia.